

**SOLUCIÓN AL PROBLEMA DEL *FLOW SHOP* HÍBRIDO (*HFS*) CON  
MÁQUINAS PARALELAS NO RELACIONADAS Y *BUFFERS* DE TAMAÑO  
LIMITADO MEDIANTE UNA METAHEURÍSTICA BASADA EN EL ALGORITMO  
COMPETITIVO IMPERIALISTA (*ICA*)**

**CAMILO ANDRÉS PABÓN SERRANO**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS  
ESCUELA DE ESTUDIOS INDUSTRIALES Y EMPRESARIALES  
BUCARAMANGA  
2017**

**SOLUCIÓN AL PROBLEMA DEL *FLOW SHOP* HÍBRIDO (*HFS*) CON  
MÁQUINAS PARALELAS NO RELACIONADAS Y *BUFFERS* DE TAMAÑO  
LIMITADO MEDIANTE UNA METAHEURÍSTICA BASADA EN EL ALGORITMO  
COMPETITIVO IMPERIALISTA (*ICA*)**

**CAMILO ANDRÉS PABÓN SERRANO**

**Trabajo de grado para optar al título de Ingeniero Industrial**

**Director**

**EDWIN ALBERTO GARAVITO HERNÁNDEZ**

M.s.c. Ingeniería Industrial

**Codirector**

**ELIANA MARCELA PEÑA TIBADUIZA**

M.s.c. Ingeniería Industrial

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS  
ESCUELA DE ESTUDIOS INDUSTRIALES Y EMPRESARIALES  
BUCARAMANGA**

**2017**

## AGRADECIMIENTOS

Primero que todo a nuestro señor Dios por brindarme sabiduría, calma, compañía y siempre guiarme y acompañarme por este camino. Este triunfo es de ÉL a través de mí.

A mis padres Soraya y Fermin por su esfuerzo, entrega y cariño. Son los mejores padres que la vida me pudo dar y mi mayor razón para querer salir cada día adelante. Quisiera devolverles todo lo que se merecen por su amor y confianza hacia mi, pero ni con el universo bastaría para eso.

A mi tía Omaira por ser más que una tía, una madre y brindarme sus cuidados desde el momento que nací. A mis hermanos Ana y Santi que además de ser mis hermanos, consejeros y soporte en muchos momentos difíciles, son mis mejores amigos. A mi amada Elenita por brindarme motivación y felicidad con solo una sonrisa o una palabra en aquellas tardes de trabajo y esfuerzo en Diciembre.

También quisiera agradecer el honor de haber pertenecido a la Universidad Industrial de Santander, por acogerme durante más de cinco años y brindarme momentos de conocimiento, aprendizaje y mucha satisfacción. De igual manera a la Escuela de Estudios Industriales y Empresariales, todos los profesores y funcionarios que compartieron algo de sus experiencia conmigo y que me han hecho no sólo una persona con más conocimientos sino un mejor ser humano. Hoy más que nunca me siento orgulloso de haber elegido el camino de ser un Ingeniero Industrial UIS.

En especial agradecer a los profesores Edwin Garavito y Eliana Peña por sus consejos y paciencia durante este camino y a mis compañeros Luis, Eslendis y Oliberth por su interés en ayudarme siempre y por compartirme sus ideas, sin ellos todo habría sido imposible.

Finalmente quisiera agradecer a Camila por acompañarme en todo momento, día a día, mes a mes y año a año y estar siempre ahí para mí, para cuidarme, darme aliento y ser mi soporte más grande. Este éxito es por ti también y sabes que siempre voy a estar para ti.

## CONTENIDO

	<b>Pág.</b>
INTRODUCCIÓN .....	23
1. GENERALIDADES DEL PROYECTO.....	26
1.1. PLANTEAMIENTO DEL PROBLEMA .....	26
1.2. JUSTIFICACIÓN DEL PROYECTO .....	27
1.3. OBJETIVOS .....	31
1.3.1. Objetivo general .....	31
1.3.2. Objetivos específicos.....	31
2. MARCO TEÓRICO.....	32
2.1. OPTIMIZACIÓN .....	32
2.1.1. Optimización combinatoria .....	33
2.2. COMPLEJIDAD COMPUTACIONAL.....	34
2.2.1. Tipos de complejidad computacional .....	35
2.3. MÉTODOS DE SOLUCIÓN DE PROBLEMAS DE OPTIMIZACIÓN COMBINATORIA.....	36
2.3.1. Algoritmos exactos .....	37
2.3.2. Algoritmos heurísticos .....	38
2.3.3. Algoritmos metaheurísticos .....	40
2.3.4. Metaheurísticas utilizadas para solucionar problemas de optimización combinatoria.....	41
2.4. PROGRAMACIÓN DE OPERACIONES .....	52
2.4.1. Clases de Schedule.....	52
2.4.2. Tipos de problema de scheduling.....	53
2.5. FLOW SHOP HÍBRIDO .....	55
2.5.1. Notación del problema.....	56
2.5.2. Buffers limitados.....	57
3. REVISIÓN DE LITERATURA.....	60
3.1. ANÁLISIS BIBLIOMÉTRICO .....	60
3.2. ANÁLISIS DE LA LITERATURA.....	64
4. DESARROLLO DEL MODELO MATEMÁTICO .....	75

4.1. DESCRIPCIÓN DEL SISTEMA PRODUCTIVO A ESTUDIARSE .....	75
4.2. ESQUEMA MATEMÁTICO DEL MODELO .....	76
4.2.1. Modelo de programación lineal entera mixta del problema .....	78
4.3. ANÁLISIS DE LOS RESULTADOS OBTENIDOS MEDIANTE EL MODELO DE PROGRAMACIÓN LINEAL ENTERA MIXTA DEL PROBLEMA .....	84
5. ESQUEMA DE LA METAHEURÍSTICA .....	90
5.1. PRESENTACIÓN DETALLADA DE LA METAHEURÍSTICA Y DEL ALGORITMO.....	91
5.1.1. Generación de los países iniciales. ....	91
5.1.2. Movimiento de los países colonia hacia su país imperialista.....	100
5.1.3. Información compartida entre las colonias .....	102
5.1.4. Revolución de los países colonia .....	103
5.1.5. Guerra entre los imperios y supresión del imperio más débil .....	104
5.1.6. Condición de parada. ....	106
6. PROCESO DE VALIDACIÓN Y RESULTADOS DE LA METAHEURÍSTICA .	108
6.1. DETALLE Y ANÁLISIS DE LOS RESULTADOS.....	110
7. CONCLUSIONES.....	143
8. RECOMENDACIONES .....	145
BIBLIOGRAFÍA .....	147
ANEXOS.....	155

## LISTA DE TABLAS

	<b>Pág.</b>
Tabla 1. <i>Gap</i> relativo para instancias con siete trabajos y nueve trabajos.....	85
Tabla 2. Tiempos (u.t) de procesamiento para i5j2k3-1 .....	96
Tabla 3. Tiempos (u.t) de alistamiento para i5j2k3-1.....	96
Tabla 4. Cálculo del costo total del país i5j2k3-1. ....	98
Tabla 5. Factores y niveles del <i>ICA</i> .....	109
Tabla 6. Resultados obtenidos para la instancia i7j2k3-1. ....	111
Tabla 7. Resultados obtenidos para la instancia i7j2k3-2. ....	113
Tabla 8. Resultados obtenidos para la instancia i7j3k3-1. ....	114
Tabla 9. Resultados obtenidos para la instancia i7j3k3-2. ....	116
Tabla 10. Resultados obtenidos para la instancia i7j3k3-3. ....	117
Tabla 11. Resultados obtenidos para la instancia i7j2k5-1. ....	118
Tabla 12. Resultados obtenidos para la instancia i7j2k5-2. ....	120
Tabla 13. Resultados obtenidos para la instancia i7j3k5-1. ....	121
Tabla 14. Resultados obtenidos para la instancia i7j3k5-2. ....	122
Tabla 15. Resultados obtenidos para la instancia i7j3k5-3. ....	124
Tabla 16. Resultados obtenidos para la instancia i9j2k3-1. ....	125
Tabla 17. Resultados obtenidos para la instancia i9j2k3-2. ....	127
Tabla 18. Resultados obtenidos para la instancia i9j3k3-1. ....	128
Tabla 19. Resultados obtenidos para la instancia i9j3k3-2. ....	129
Tabla 20. Resultados obtenidos para la instancia i9j3k3-3. ....	131
Tabla 21. Resultados obtenidos para la instancia i9j2k5-1. ....	132
Tabla 22. Resultados obtenidos para la instancia i9j2k5-2. ....	134
Tabla 23. Resultados obtenidos para la instancia i9j3k5-1. ....	135
Tabla 24. Resultados obtenidos para la instancia i9j3k5-2. ....	136
Tabla 25. Resultados obtenidos para la instancia i9j3k5-3. ....	138
Tabla 26. Resumen de los resultados obtenidos por el <i>ICA</i> ordenados por tamaño de <i>buffer</i> . ....	139

Tabla 27. Tiempos de procesamiento y alistamiento de tareas para la instancia i7j2k3-1.....	160
Tabla 28. Tiempos de procesamiento y alistamiento de tareas para la instancia i7j2k3-2.....	161
Tabla 29. Tiempos de procesamiento y alistamiento de tareas para la instancia i7j3k3-1.....	162
Tabla 30. Tiempos de procesamiento y alistamiento de tareas para la instancia i7j3k3-2.....	163
Tabla 31. Tiempos de procesamiento y alistamiento de tareas para la instancia i7j3k3-3.....	164
Tabla 32. Tiempos de procesamiento y alistamiento de tareas para la instancia i7j2k5-1.....	165
Tabla 33. Tiempos de procesamiento y alistamiento de tareas para la instancia i7j2k5-2.....	166
Tabla 34. Tiempos de procesamiento y alistamiento de tareas para la instancia i7j3k5-1.....	167
Tabla 35. Tiempos de procesamiento y alistamiento de tareas para la instancia i7j3k5-2.....	168
Tabla 36. Tiempos de procesamiento y alistamiento de tareas para la instancia i7j3k5-3.....	169
Tabla 37. Tiempos de procesamiento y alistamiento de tareas para la instancia i9j2k3-1.....	170
Tabla 38. Tiempos de procesamiento y alistamiento de tareas para la instancia i9j2k3-2.....	171
Tabla 39. Tiempos de procesamiento y alistamiento de tareas para la instancia i9j3k3-1.....	172
Tabla 40. Tiempos de procesamiento y alistamiento de tareas para la instancia i9j3k3-2.....	173
Tabla 41. Tiempos de procesamiento y alistamiento de tareas para la instancia i9j3k3-3.....	174

Tabla 42. Tiempos de procesamiento y alistamiento de tareas para la instancia i9j2k5-1.....	175
Tabla 43. Tiempos de procesamiento y alistamiento de tareas para la instancia i9j2k5-2.....	176
Tabla 44. Tiempos de procesamiento y alistamiento de tareas para la instancia i9j3k5-1.....	177
Tabla 45. Tiempos de procesamiento y alistamiento de tareas para la instancia i9j3k5-2.....	178
Tabla 46. Tiempos de procesamiento y alistamiento de tareas para la instancia i9j3k5-3.....	179
Tabla 47. Tiempos de procesamiento y alistamiento de tareas para la instancia i7j2k3 de Moratto y Pérez.....	180
Tabla 48. Tiempos de procesamiento y alistamiento de tareas para la instancia i7j3k3 de Moratto y Pérez.....	181
Tabla 49. Tiempos de procesamiento y alistamiento de tareas para la instancia i7j2k5 de Moratto y Pérez.....	182
Tabla 50. Tiempos de procesamiento y alistamiento de tareas para la instancia i7j3k5 de Moratto y Pérez.....	183
Tabla 51. Tiempos de procesamiento y alistamiento de tareas para la instancia i9j2k3 de Moratto y Pérez.....	184
Tabla 52. Tiempos de procesamiento y alistamiento de tareas para la instancia i9j3k3 de Moratto y Pérez.....	185
Tabla 53. Tiempos de procesamiento y alistamiento de tareas para la instancia i9j2k5 de Moratto y Pérez.....	186
Tabla 54. Tiempos de procesamiento y alistamiento de tareas para la instancia i9j3k5 de Moratto y Pérez.....	187
Tabla 55. Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i7j2k3-1.....	237

Tabla 56. Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i7j2k3-2.....	239
Tabla 57. Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i7j3k3-1.....	239
Tabla 58. Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i7j3k3-2.....	241
Tabla 59. Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i7j3k3-3.....	242
Tabla 60. Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i7j2k5-1.....	243
Tabla 61. Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i7j2k5-2.....	244
Tabla 62. Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i7j3k5-1.....	246
Tabla 63. Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i7j3k5-2.....	247
Tabla 64. Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i7j3k5-3.....	249
Tabla 65. Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i9j2k3-1.....	249

Tabla 66. Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i9j2k3-2.....	251
Tabla 67. Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i9j3k3-1.....	252
Tabla 68. Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i9j3k3-2.....	254
Tabla 69. Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i9j3k3-3.....	254
Tabla 70. Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i9j2k5-1.....	256
Tabla 71. Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i9j2k5-2.....	257
Tabla 72. Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i9j3k5-1.....	259
Tabla 73. Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i9j3k5-2.....	259
Tabla 74. Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i9j3k5-3.....	261
Tabla 75. Factores y niveles del <i>ICA</i> tenidos en cuenta para el diseño experimental.....	264
Tabla 76. Combinaciones de niveles de tratamiento del diseño factorial $2^{k-1}$ .....	264

Tabla 77. ANOVA para la instancia i7j2k3-1. ....	265
Tabla 78. ANOVA para la instancia i7j2k3-2. ....	267
Tabla 79. ANOVA para la instancia i7j3k3-1. ....	269
Tabla 80. ANOVA para la instancia i7j3k3-2. ....	271
Tabla 81. ANOVA para la instancia i7j3k3-3. ....	273
Tabla 82. ANOVA para la instancia i7j2k5-1. ....	275
Tabla 83. ANOVA para la instancia i7j2k5-2. ....	277
Tabla 84. ANOVA para la instancia i7j3k5-1. ....	279
Tabla 85. ANOVA para la instancia i7j3k5-2. ....	281
Tabla 86. ANOVA para la instancia i7j3k5-3. ....	283
Tabla 87. ANOVA para la instancia i9j2k3-1. ....	285
Tabla 88. ANOVA para la instancia i9j2k3-2. ....	287
Tabla 89. ANOVA para la instancia i9j3k3-1. ....	290
Tabla 90. ANOVA para la instancia i9j3k3-2. ....	291
Tabla 91. ANOVA para la instancia i9j3k3-3. ....	294
Tabla 92. ANOVA para la instancia i9j2k5-1. ....	296
Tabla 93. ANOVA para la instancia i9j2k5-2. ....	298
Tabla 94. ANOVA para la instancia i9j3k5-1. ....	301
Tabla 95. ANOVA para la instancia i9j3k5-2. ....	303
Tabla 96. ANOVA para la instancia i9j3k5-3. ....	305

## LISTA DE GRÁFICAS

	<b>Pág.</b>
Gráfica 1. Gráfico de artículos publicados por autor .....	61
Gráfica 2. Gráfico de artículos publicados ordenados por país .....	62
Gráfica 3. Gráfico de artículos publicados por año.....	63
Gráfica 4. Gráfico de artículos publicados ordenados por revista .....	64
Gráfica 5. Gráfica de porcentaje de <i>gap</i> relativo para las soluciones obtenidas por <i>GAMS</i> en instancias con tamaño de <i>buffer</i> 1. ....	88
Gráfica 6. Gráfica de porcentaje de <i>gap</i> relativo para las soluciones obtenidas por <i>GAMS</i> en instancias con tamaño de <i>buffer</i> 2. ....	88
Gráfica 7. Gráfica de porcentaje de <i>gap</i> relativo para las soluciones obtenidas por <i>GAMS</i> en instancias con tamaño de <i>buffer</i> 3. ....	89
Gráfica 8. Diagrama de <i>gantt</i> para el costo total o <i>makespan</i> del país i5j2k3-1....	98
Gráfica 9. Gráfica de resumen de resultados obtenidos por el <i>ICA</i> para la mejor solución observada en instancias con tamaño de <i>buffer</i> 1.....	140
Gráfica 10. Gráfica de resumen de resultados obtenidos por el <i>ICA</i> para la mejor solución observada en instancias con tamaño de <i>buffer</i> 2.....	140
Gráfica 11. Gráfica de resumen de resultados obtenidos por el <i>ICA</i> para la mejor solución observada en instancias con tamaño de <i>buffer</i> 3.....	141
Gráfica 12. Gráfica de resumen de porcentaje de diferencia en instancias con tamaño de <i>buffer</i> 1. ....	141
Gráfica 13. Gráfica de resumen de porcentaje de diferencia en instancias con tamaño de <i>buffer</i> 2. ....	142
Gráfica 14. Gráfica de resumen de porcentaje de diferencia en instancias con tamaño de <i>buffer</i> 3. ....	142
Gráfica 15. Diagrama de Pareto y gráfica normal de efector estandarizados para el <i>makespan</i> de la instancia i7j2k3-1. ....	265
Gráfica 16. Diagrama de residuos para el <i>makespan</i> de la instancia i7j2k3-1....	266

Gráfica 17. Diagrama de Pareto y gráfica normal de efector estandarizados para el <i>makespan</i> de la instancia i7j2k3-2. ....	267
Gráfica 18. Diagrama de residuos para el <i>makespan</i> de la instancia i7j2k3-2. ...	268
Gráfica 19. Diagrama de Pareto y gráfica normal de efector estandarizados para el <i>makespan</i> de la instancia i7j3k3-1. ....	269
Gráfica 20. Diagrama de residuos para el <i>makespan</i> de la instancia i7j3k3-1. ...	270
Gráfica 21. Diagrama de Pareto y gráfica normal de efector estandarizados para el <i>makespan</i> de la instancia i7j3k3-2. ....	271
Gráfica 22. Diagrama de residuos para el <i>makespan</i> de la instancia i7j3k3-2. ...	272
Gráfica 23. Diagrama de Pareto y gráfica normal de efector estandarizados para el <i>makespan</i> de la instancia i7j3k3-3. ....	274
Gráfica 24. Diagrama de residuos para el <i>makespan</i> de la instancia i7j3k3-3. ...	274
Gráfica 25. Diagrama de Pareto y gráfica normal de efector estandarizados para el <i>makespan</i> de la instancia i7j2k5-1. ....	275
Gráfica 26. Diagrama de residuos para el <i>makespan</i> de la instancia i7j2k5-1. ...	276
Gráfica 27. Diagrama de Pareto y gráfica normal de efector estandarizados para el <i>makespan</i> de la instancia i7j2k5-2. ....	277
Gráfica 28. Diagrama de residuos para el <i>makespan</i> de la instancia i7j2k5-2. ...	278
Gráfica 29. Diagrama de residuos para el <i>makespan</i> de la instancia i7j3k5-1. ...	280
Gráfica 30. Diagrama de Pareto y gráfica normal de efector estandarizados para el <i>makespan</i> de la instancia i7j3k5-1. ....	280
Gráfica 31. Diagrama de residuos para el <i>makespan</i> de la instancia i7j3k5-2. ...	282
Gráfica 32. Diagrama de Pareto y gráfica normal de efector estandarizados para el <i>makespan</i> de la instancia i7j3k5-2. ....	282
Gráfica 33. Diagrama de residuos para el <i>makespan</i> de la instancia i7j3k5-3. ...	284
Gráfica 34. Diagrama de Pareto y gráfica normal de efector estandarizados para el <i>makespan</i> de la instancia i7j3k5-3. ....	284
Gráfica 35. Diagrama de residuos para el <i>makespan</i> de la instancia i9j2k3-1. ...	285
Gráfica 36. Diagrama de Pareto y gráfica normal de efector estandarizados para el <i>makespan</i> de la instancia i9j2k3-1. ....	286

Gráfica 37. Diagrama de Pareto y gráfica normal de efector estandarizados para el <i>makespan</i> de la instancia i9j2k3-2. ....	287
Gráfica 38. Diagrama de residuos para el <i>makespan</i> de la instancia i9j2k3-2. ...	288
Gráfica 39. Diagrama de Pareto y gráfica normal de efector estandarizados para el <i>makespan</i> de la instancia i9j3k3-1. ....	290
Gráfica 40. Diagrama de residuos para el <i>makespan</i> de la instancia i9j3k3-1. ...	291
Gráfica 41. Diagrama de Pareto y gráfica normal de efector estandarizados para el <i>makespan</i> de la instancia i9j3k3-2. ....	292
Gráfica 42. Diagrama de residuos para el <i>makespan</i> de la instancia i9j3k3-2. ...	292
Gráfica 43. Diagrama de Pareto y gráfica normal de efector estandarizados para el <i>makespan</i> de la instancia i9j3k3-3. ....	295
Gráfica 44. Diagrama de residuos para el <i>makespan</i> de la instancia i9j3k3-3. ...	295
Gráfica 45. Diagrama de Pareto y gráfica normal de efector estandarizados para el <i>makespan</i> de la instancia i9j2k5-1. ....	297
Gráfica 46. Diagrama de residuos para el <i>makespan</i> de la instancia i9j2k5-1. ...	297
Gráfica 47. Diagrama de Pareto y gráfica normal de efector estandarizados para el <i>makespan</i> de la instancia i9j2k5-2. ....	299
Gráfica 48. Diagrama de residuos para el <i>makespan</i> de la instancia i9j2k5-2. ...	300
Gráfica 49. Diagrama de Pareto y gráfica normal de efector estandarizados para el <i>makespan</i> de la instancia i9j3k5-1. ....	301
Gráfica 50. Diagrama de residuos para el <i>makespan</i> de la instancia i9j3k5-1. ...	302
Gráfica 51. Diagrama de residuos para el <i>makespan</i> de la instancia i9j3k5-2. ...	304
Gráfica 52. Diagrama de Pareto y gráfica normal de efector estandarizados para el <i>makespan</i> de la instancia i7j3k5-2. ....	304
Gráfica 53. Diagrama de residuos para el <i>makespan</i> de la instancia i9j3k5-3. ...	306
Gráfica 54. Diagrama de Pareto y gráfica normal de efector estandarizados para el <i>makespan</i> de la instancia i9j3k5-3. ....	306

## LISTA DE FIGURAS

	Pág.
Figura 1. Diagrama de flujo del Algoritmo Competitivo Imperialista .....	48
Figura 2. Población inicial de países imperialistas junto a sus colonias.....	49
Figura 3. Competencia entre los imperialistas.....	51
Figura 4. Esquema básico del funcionamiento de una línea <i>flow shop</i> con <i>buffers</i> . .....	58
Figura 5. Figura del sistema productivo <i>HFS</i> con <i>buffers</i> a estudiarse .....	75
Figura 6. Determinación del tiempo de completamiento de la tarea $i$ , en la estación $k$ ( $C_{i,k}$ ) para el caso donde $E_{h,k} \geq C_{i,k} - 1$ . .....	83
Figura 7. Determinación del tiempo de completamiento de la tarea $i$ , en la estación $k$ ( $C_{i,k}$ ) para el caso donde $C_{i,k} - 1 \geq E_{h,k}$ . .....	83
Figura 8. Explicación de la dependencia del tiempo de completamiento de la tarea $i$ , en la estación $k$ ( $C_{i,k}$ ) de $C_{i,k} - 1$ y no de $E_{i,k} - 1$ . .....	84
Figura 9. Muestra de creación de un país para la instancia $i5j2k3-1$ .....	92
Figura 10. Asimilación del imperialista por parte de las colonias. ....	101
Figura 11. Procedimiento de información compartida entre las colonias. ....	102
Figura 12. Procedimiento de revolución de los países colonia.....	103
Figura 13. Diagrama de flujo de la metaheurística propuesta. ....	107
Figura 14. Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia $i7j2k3-1$ .....	238
Figura 15. Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia $i7j2k3-2$ .....	240
Figura 16. Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia $i7j3k3-1$ .....	241

Figura 17. Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i7j3k3-2.....	242
Figura 18. Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i7j3k3-3.....	245
Figura 19. Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i7j2k5-1.....	245
Figura 20. Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i7j2k5-2.....	246
Figura 21. Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i7j3k5-1.....	247
Figura 22. Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i7j3k5-2.....	248
Figura 23. Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i7j3k5-3.....	250
Figura 24. Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i9j2k3-1.....	251
Figura 25. Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i9j2k3-2.....	252
Figura 26. Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i9j3k3-1.....	253

Figura 27. Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i9j3k3-2.....	255
Figura 28. Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i9j3k3-3.....	256
Figura 29. Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i9j2k5-1.....	257
Figura 30. Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i9j2k5-2.....	258
Figura 31. Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i9j3k5-1.....	260
Figura 32. Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i9j3k5-2.....	261
Figura 33. Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con <i>buffers</i> limitados y <i>buffers</i> ilimitados para la instancia i9j3k5-3.....	262

## LISTA DE ANEXOS

	<b>Pág.</b>
Anexo A. Detalle del modelo matemático utilizado para el problema del <i>HFS</i> con <i>buffers</i> limitados escrito en lenguaje <i>GAMS</i> .....	155
Anexo B. Tablas con los tiempos de procesamiento y alistamiento de las tareas para cada una de las instancias. ....	159
Anexo C. Pseudocódigo de programación en lenguaje <i>Matlab</i> de la metaheurística basada en el algoritmo competitivo imperialista. ....	188
Anexo E. Análisis comparativo entre los resultados obtenidos por el <i>ICA</i> para el <i>HFS</i> con <i>buffers</i> limitados y el <i>ICA</i> para el <i>HFS</i> con <i>buffers</i> ilimitados. ....	235
Anexo F. Diseño experimental para determinar los factores más influyentes en el desempeño de la metaheurística basada en el <i>ICA</i> para el <i>HFS</i> con <i>buffers</i> limitados. ....	263

Los anexos citados en la siguiente lista se encuentran en la carpeta adjunta:

Anexo D. Resultados tenidos en cuenta para la validación de la metaheurística basada en el *ICA* para el problema del *HFS* con *buffers* limitados.

Anexo G. Datos tenidos en cuenta para el diseño experimental fraccionado  $2^{k-1}$  para determinar los factores más influyentes en la variable respuesta *makespan*.

## RESUMEN

**TÍTULO:** “SOLUCIÓN AL PROBLEMA DEL *FLOW SHOP* HÍBRIDO (*HFS*) CON MÁQUINAS PARALELAS NO RELACIONADAS Y *BUFFERS* DE TAMAÑO LIMITADO MEDIANTE UNA METAHEURÍSTICA BASADA EN EL ALGORITMO COMPETITIVO IMPERIALISTA (*ICA*)”<sup>1</sup>

**AUTOR:** PABÓN SERRANO, Camilo Andrés<sup>2</sup>

**PALABRAS CLAVE:** *Flow shop* híbrido, máquinas paralelas no relacionadas, *buffers* de tamaño limitado, algoritmo competitivo imperialista (*ICA*), metaheurística.

### DESCRIPCIÓN

Es usual el problema de asignar  $n$  tareas a  $m$  etapas de procesamiento en distintos ambientes de manufactura, donde cada etapa tiene múltiples máquinas en paralelo, las cuales pueden ser idénticas, uniformes o no relacionadas de acuerdo con la configuración del sistema. Para esta clase de problemas, el objetivo es encontrar una programación óptima que permita atender una función objetivo, ya sea la disminución de tiempos de procesamiento totales, el tiempo de flujo medio, el número de tareas tardías, entre otros.

El problema descrito anteriormente, se encuentra en la literatura como “*flow shop*” híbrido o “*flow shop*” flexible y hace parte del área de investigación de la optimización combinatoria clasificado como “*NP-hard*”, a causa de que es poco factible que exista un algoritmo de tiempo polinómico con la capacidad de encontrar la solución óptima.

La presente investigación desarrolla una metaheurística basada en el algoritmo competitivo imperialista combinado con operadores de algoritmos genéticos, con el objetivo de minimizar la variable “*makespan*” del problema del “*flow shop*” híbrido con máquinas paralelas no relacionadas, tiempos de alistamiento dependientes de la secuencia y “*buffers*” de tamaño limitado, una restricción la cual dicta que las zonas de almacenamiento de trabajos entre etapas tienen una capacidad limitada en cantidad para albergar los mismos.

Para comprobar la eficiencia del algoritmo se desarrolla una comparación de los resultados obtenidos para veinte instancias generadas por el autor entre las soluciones dadas por el modelo de programación lineal entera del problema programado en GAMS® y las respuestas arrojadas por la metaheurística programada en MATLAB®. Se encuentra que la metaheurística basada en el *ICA* da mejores soluciones que el método exacto para cuatro instancias y con porcentaje de diferencia muy bajo para otras ocho instancias con tiempos mucho menores a los utilizados por el método exacto.

---

<sup>1</sup> Proyecto de Grado.

<sup>2</sup> Facultad de Ingenierías Fisicomecánicas. Escuela de Estudios Industriales y Empresariales. Director: M.s.c. Edwin A. Garavito Hernández. Codirectora: M.s.c. Eliana M. Peña Tibaduiza.

## ABSTRACT

**TITLE:** “SOLUTION TO HYBRID FLOW SHOP (HFS) PROBLEM WITH UNRELATED PARALLEL MACHINES AND LIMITED SIZE BUFFERS THROUGH A METAHEURISTIC BASED ON THE IMPERIALIST COMPETITIVE ALGORITHM (ICA)”.<sup>3</sup>

**AUTHOR:** PABÓN-SERRANO, Camilo Andrés<sup>4</sup>

**KEYWORDS:** Hybrid flow shop, unrelated parallel machines, limited size buffers, imperialist competitive algorithm (ICA), metaheuristic.

### DESCRIPTION

The problem of assigning  $n$  tasks to  $m$  processing stages in different manufacturing environments is common, where each stage has multiple machines in parallel, which may be identical, uniform or unrelated according to the system configuration. For this kind of problems, the objective is to find out an optimal programming that allows to attend an objective function, either the reduction of total processing times, the average flow time, the number of late tasks, among others.

The problem described above is found in the literature as *hybrid flow shop* or *flexible flow shop* and is part of the research area of combinatorial optimization classified as *NP-hard*, because it is unfeasible to have a polynomial time algorithm with the ability to find the optimal solution.

The present research develops a metaheuristic based on the imperialist competitive algorithm combined with operators of genetic algorithms, with the objective of minimizing the *makespan* variable of the *hybrid flow shop* problem with unrelated parallel machines, sequence dependent set-up times and *buffers* of limited size, a restriction which dictates that inter-stage job storage areas have limited capacity in quantity to accommodate them.

To verify the efficiency of the algorithm, a comparison for the results obtained for twenty instances generated by the author between the solutions given by the linear programming model of the problem programmed in GAMS® and the answers given by the metaheuristics programmed in MATLAB® is developed. It is found that the metaheuristic based on the *ICA* gives better solutions than the exact method for four instances and with very low percentage difference for other eight instances with times much smaller than those used by the exact method.

---

<sup>3</sup> Proyecto de Grado.

<sup>4</sup> Facultad de Ingenierías Fisicomecánicas. Escuela de Estudios Industriales y Empresariales. Director: M.s.c. Edwin A. Garavito Hernández. Codirectora: M.s.c. Eliana M. Peña Tibaduiza

## INTRODUCCIÓN

“La administración de operaciones y suministros (AOS) ha sido un elemento medular para el incremento de la productividad que han registrado empresas de todo el mundo. Para crear una ventaja competitiva con las operaciones es preciso comprender cómo la función de operaciones y suministro contribuye a incrementar la productividad.”<sup>5</sup> Con base en lo inmediatamente citado, se vislumbra el papel preponderante de la dirección de operaciones a través de diferentes procesos, como el proceso de programación de trabajos, para lograr ventajas competitivas con el fin de mejorar el servicio al cliente.

El proceso de programación de la producción debe ser aplicado a los entornos de manufactura del siglo XXI, se deben analizar los sistemas con sus características reales, pues el lograr la mejor programación de la producción trae beneficios marcados para las empresas como reducción de costos, entregas más rápidas, mayor flexibilidad y fiabilidad en la entrega de pedidos.<sup>6</sup>

Además de lo descrito anteriormente, en algunos sistemas de producción modernos como los *JIT (Just In Time)* o líneas de flujo flexibles un correcto manejo de la programación de operaciones lleva a una reducción del inventario en proceso y a evitar bloqueos en máquinas de producción por falta de espacio en zonas de almacenamiento intermedias (también conocidas como *buffers*).<sup>7</sup>

El sistema de producción *flow shop* se presenta en la industria cuando un grupo de trabajos debe ser procesado siguiendo un flujo unidireccional a través de diferentes etapas. El *flow shop* presenta como característica que cada etapa de

---

<sup>5</sup> CHASE, R.B.; JACOBS, F. R.; AQUILANO, N.J., *Administración de Operaciones, Producción y cadena de suministros*, Editorial: McGraw Hill, 2009, 12va. Edición, p. prefacio.

<sup>6</sup> HEIZER, J.; RENDER, B., *Dirección de la producción y de operaciones, Decisiones tácticas*, Editorial: Pearson Educación, 2008, 8va. Edición, p. 203.

<sup>7</sup> SOLTANI, S. Abolfazli; KARIMI, Behrooz. Cyclic hybrid flow shop scheduling problem with limited buffers and machine eligibility constraints. En: *The International Journal of Advanced Manufacturing Technology*, 2014, p. 1-17.

procesamiento consta de solo una máquina, pero debido a cambios en la demanda se puede presentar la necesidad de incrementar la capacidad productiva mediante la compra de nuevos procesadores para algunas estaciones.<sup>8</sup> Aunque los nuevos procesadores pueden presentar características mejoradas con respecto a velocidad o tiempos de alistamiento frente a los antiguos procesadores, ambos tipos de máquina pueden coexistir en una misma línea de producción.

Presente en industrias como la textil o automotriz, esta nueva configuración del *flow shop* busca asignar y secuenciar de la mejor manera un grupo de  $n$  tareas a través de  $s$  etapas de procesamiento, algunas etapas con varias máquinas de características distintas dentro de una misma estación, en orden de optimizar una función objetivo, a esta variante del problema se le suele denominar como *flow shop* híbrido.

Debido a su alta complejidad y aplicabilidad, el *flow shop* híbrido ha despertado mucho interés en investigadores del área de la optimización combinatoria y se han desarrollado esquemas de solución heurísticos y metaheurísticos con el fin de encontrar soluciones factibles al problema junto a sus diferentes restricciones.

No siendo ajeno al interés despertado por el problema, el trabajo de investigación aquí presentado plantea a través de una metaheurística basada en el algoritmo competitivo imperialista dar solución al problema del *flow shop* híbrido con máquinas paralelas no relacionadas, tiempos de alistamiento dependientes de la secuencia y *buffers* de tamaño limitado.

---

<sup>8</sup> RIBAS, Imma; LEISTEN, Rainer; FRAMIÑAN, Jose M. Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective. En: Computers & Operations Research, 2010, vol. 37, no 8, p. 1439-1454.

<b>Cumplimiento de objetivos</b>	
<b>Objetivos específicos</b>	<b>Numerales relacionados</b>
Revisión de la literatura sobre el problema del <i>flow shop</i> híbrido con máquinas paralelas no relacionadas y <i>buffers</i> de tamaño limitado.	3
Formular el modelo de programación matemática que se ajuste el problema.	4.2
Ajustar la estructura del Algoritmo Competitivo Imperialista (ICA) de acuerdo a las necesidades del <i>flow shop</i> híbrido con máquinas paralelas no relacionadas y <i>buffers</i> de tamaño limitado.	5
Implementar el algoritmo adaptado en <i>MATLAB</i> y validarlo a través de diferentes instancias.	6, Anexo C
Comparar la solución encontrada mediante el modelo de programación matemática desarrollado en <i>GAMS</i> con los resultados obtenidos mediante el Algoritmo Competitivo Imperialista (ICA).	6.1

## 1. GENERALIDADES DEL PROYECTO

### 1.1 PLANTEAMIENTO DEL PROBLEMA

El *flow shop* híbrido (*Hybrid Flow Shop, HFS*) es un problema en el cual se tiene un conjunto de trabajos a ser procesados en una serie de  $n$  etapas con  $m$  máquinas de procesamiento en cada una de ellas; el objetivo del problema es establecer una adecuada asignación y secuenciación de los trabajos con el fin de optimizar una función objetivo siempre relacionada con el tiempo de terminación de los trabajos, la más común es el *makespan*. El problema también considera más de una etapa de procesamiento, por lo menos una etapa de procesamiento con más de una máquina y que todos los trabajos deben ser procesados en un mismo orden.

El *HFS* contempla diferentes variantes para su planteamiento, cambiando desde el número de etapas o la cantidad de las máquinas por *stage* hasta agregar restricciones para hacer el problema teórico más cercano a la realidad de industrias como la textil o automovilística donde el mismo se presenta. A pesar de esto, muy pocas veces se ha trabajado el problema en sus formas más realistas pues casi siempre se considera su forma más básica, por lo tanto, son pocas las investigaciones que incluyen más restricciones y que buscan aproximarse al entorno real de las industrias de fabricación.

El presente proyecto busca estudiar el problema del *flow shop* híbrido con máquinas paralelas no relacionadas, que se caracteriza por ser un problema para el cual no existen algoritmos que puedan encontrar una solución óptima en un tiempo polinómico, es decir, un problema *NP-Hard*.<sup>9</sup> Además de la restricción de máquinas paralelas no relacionadas también se incluyen las restricciones de tiempos de alistamiento dependientes de la secuencia y zonas de almacenamiento

---

<sup>9</sup> BOZORGIRAD, Mir Abbas; LOGENDRAN, Rasaratnam. Sequence-dependent group scheduling problem on unrelated-parallel machines. En: Expert Systems with Applications, 2012, vol. 39, p. 9021-9030.

o *buffers* de tamaño limitado, acercándolo a la realidad de la industria moderna y aportando mayor complejidad al *HFS*.

Con máquinas paralelas no relacionadas se habla de que los tiempos de procesamiento para los trabajos dependen en específico de cada procesador dentro de cada *stage*, en otras palabras, los tiempos de procesamiento cambian entre una y otra máquina de la misma etapa.

Con respecto a los tiempos de alistamiento dependientes de la secuencia se refiere a que el tiempo de *set-up* entre el trabajo  $h$  y el trabajo  $h+1$  va a depender de qué par de trabajos sean asignados a las posiciones  $h$  y  $h+1$  dentro de una misma etapa y para una misma máquina.

Al hablar de *buffers* de tamaño limitado, se asume la existencia de una zona intermedia de almacenamiento de trabajos entre la estación  $k$  y la estación  $k+1$  por donde pueden pasar los trabajos y almacenarse durante un tiempo en caso de que la siguiente etapa de procesamiento no se encuentre disponible, además, dicha zona tiene una capacidad de acumulación de trabajos limitada.

Como consecuencia de ser un problema *NP-Hard* se han desarrollado por parte de los investigadores varias metaheurísticas basadas en diferentes algoritmos y sus combinaciones para buscar dar solución al problema del *flow shop* híbrido. En este proyecto la metaheurística será basada en el Algoritmo Competitivo Imperialista (*Imperialist Competitive Algorithm, ICA*) que se ha venido aplicando en los últimos años como una nueva forma de solución al problema del *HFS* y se busca comparar su efectividad para esta nueva variante del problema frente a la implementación de un método exacto.

## **1.2 JUSTIFICACIÓN DEL PROYECTO**

Desde los primeros estudios de programación de operaciones de Johnson en 1954, el tema ha tomado una gran transcendencia debido a su grado de

complejidad<sup>10</sup> y por estar estrechamente conectado con el rendimiento en términos de velocidad, confiabilidad, calidad y costos de las empresas.<sup>11</sup>

La programación de operaciones se define de acuerdo a Soltani y Karimi como un problema de toma de decisiones donde se asigna una serie de tareas a un grupo de recursos durante un horizonte de tiempo predeterminado en orden de optimizar una función objetivo.<sup>12</sup> En el entorno competitivo actual, la secuenciación de operaciones juega un rol importante en las industrias, tanto así que una efectiva programación se ha vuelto necesaria para incluso sobrevivir en el mercado.<sup>13</sup>

En muchas industrias, es necesario incrementar el número de máquinas en paralelo para mejorar o balancear la capacidad en las estaciones, o también para reducir o eliminar el impacto del cuello de botella en la capacidad de producción.<sup>14</sup> De modificaciones como éstas nace el problema del *flow shop* híbrido, una extensión del *flow shop* original que consiste, de acuerdo a Jungwattanaki et al. en un proceso de producción multi-estaciones donde los trabajos deben visitar los *stages* en el mismo orden y que asume al menos una estación con múltiples máquinas.<sup>15</sup>

---

<sup>10</sup> ZANDIEH, M.; MOZAFFARI, E.; GHOLAMI, Mansour. A robust genetic algorithm for scheduling realistic hybrid flexible flow line problems. En: Journal of Intelligent Manufacturing, 2010, vol. 21, no 6, p. 731-743.

<sup>11</sup> DE FATIMA MORAIS, Marcia, et al. Multicriteria hybrid flow shop scheduling problem: literature review, analysis, and future research. En Independent Journal of Management & Production, 2014, vol. 5, no 4, p. 1004-1031.

<sup>12</sup> SOLTANI, S. Abolfazl; KARIMI, Behrooz. Op. Cit. p.1-17.

<sup>13</sup> ATTAR, S. F.; MOHAMMADI, M.; TAVAKKOLI-MOGHADDAM, R. Hybrid flexible flowshop scheduling problem with unrelated parallel machines and limited waiting times. En: The International Journal of Advanced Manufacturing Technology, 2013, vol. 68, no 5-8, p. 1583-1599.

<sup>14</sup> JUNGWATTANAKIT, Jitti, REODECHA, Manop, CHAOVALITWONGSE, Paveena, WERNER, Frank. A comparison of scheduling algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. En: Computers & operations research 36. 2009. Páginas: 358-378.

<sup>15</sup> JUNGWATTANAKIT, Jitti, REODECHA, Manop, CHAOVALITWONGSE, Paveena, WERNER, Frank. Ibid. p. 358-378.

Los *buffers* dentro de un *flow shop* híbrido son espacios de almacenamiento intermedio donde los trabajos son transferidos en caso de que la siguiente máquina de la secuencia esté ocupada,<sup>16</sup> generalmente se omiten o se asumen de tamaño infinito, pero como sucede en la realidad en diversas industrias o en sistemas de producción *JIT*,<sup>17</sup> dichos *buffers* tienen una capacidad limitada debido a diferentes particularidades tanto de la línea como también de las factorías.

El estudio de los *buffers* se hace importante debido a características relacionadas a ellos que pueden afectar el buen rendimiento de la programación de operaciones como es el bloqueo de las máquinas. Teniendo en cuenta que los *buffers* de tamaño ilimitado se alejan de la realidad de la industria su buena administración debe llevar a que se eviten trabajos saliendo de estaciones a buscar máquinas o zonas de almacenamiento no disponibles, en dado caso sucederá el mencionado bloqueo, definido por Rashidi, Jhandar y Zandich como un trabajo que permanece en una máquina y la bloquea para otros trabajos de la secuencia, hasta que exista disponibilidad del siguiente *buffer* o estación.<sup>18</sup>

Las máquinas paralelas no relacionadas es otra característica muy realista dentro de la industria moderna que considera que los tiempos de procesamiento de un trabajo dentro de una misma estación dependen de cada máquina en sí.<sup>19</sup> Esta restricción se puede presentar debido a diferencias entre las máquinas mismas (por ejemplo: entre máquinas nuevas y viejas), al hecho de que una máquina sea mejor que otra para un trabajo en particular o porque los trabajos tienen ciertas

---

<sup>16</sup> RASHIDI, E., JAHANDAR, M., ZANDICH, M., An improved hybrid multi-objective parallel genetic algorithm for hybrid flow shop scheduling with unrelated parallel machines. En: Int J Adv Manuf Technol 49. 2010. Páginas: 1129-1139.

<sup>17</sup> Ibid. p. 1129-1139

<sup>18</sup> RASHIDI, E., JAHANDAR, M., ZANDICH, M. Op. Cit. p. 1129-1139

<sup>19</sup> RIBAS, Imma, LEISTEN, Rainer, FRAMIÑAN, Jose. Op. Cit. p. 1439-1454.

características y solo pueden ser asignados a un tipo especial de máquina.<sup>20</sup>

Debido a que el problema del *flow shop* híbrido es *NP-Hard*,<sup>21</sup> se hace necesario en orden de reducir el consumo de recursos informáticos y de tiempo que conlleva la solución matemática clásica, el aproximarse al problema mediante métodos metaheurísticos que a pesar de no dar respuestas óptimas sí pueden dar soluciones de buena calidad al problema, en este caso, se abordará el problema a través del Algoritmo Competitivo Imperialista (*ICA* por sus siglas en inglés), que se define según Mendoza, Mago y Requena como una técnica heurística que basa su idea en el imperialismo y el proceso de competencia imperialista; este algoritmo considera el imperialismo como un nivel de evolución social humana, compuesto por modelos matemáticos y por los complicados procesos históricos y políticos, como herramienta para la optimización de la evolución.<sup>22</sup>

Finalmente se busca realizar un acercamiento a la industria a través de una naciente línea de investigación dentro de la Escuela de Estudios Industriales y Empresariales, ajustando el problema del *flow shop* híbrido con máquinas paralelas no relaciones y tiempos de alistamiento dependientes de la secuencia para incluir la restricción de *buffers* intermedios de tamaño limitado, que se presenta por autores como Zandieh et al.,<sup>23</sup> Attar et al.,<sup>24</sup> Rashidi et al.,<sup>25</sup> Wei<sup>26</sup> y

---

<sup>20</sup> Ibid. p. 1439-1454.

<sup>21</sup> RASHIDI, E., JAHANDAR, M., ZANDICH, M. Op. Cit. p. 1129-1139.

<sup>22</sup> MENDOZA, Franklin, MAGO, Enrique, REQUENA, Durylin. Herramientas computacionales "CI-API-1": Una alternativa para la solución de problemas en sistemas de distribución de energía eléctrica. En: Memorias de las IX Jornadas de Investigación 2011 – UNEXPO Puerto Ordaz. 2011. Páginas: 10-22.

<sup>23</sup> ZANDIEH, M., MOZAFFARI, E., GHOLAMI, M. Op. Cit. p. 731-743.

<sup>24</sup> ATTAR, S. F.; MOHAMMADI, M.; TAVAKKOLI-MOGHADDAM, R. Op. Cit. p.1583-1599.

<sup>25</sup> RASHIDI, E., JAHANDAR, M., ZANDICH, M. Op. Cit. p. 1129-1139.

<sup>26</sup> WEI, Qi. An Optimal Algorithm for a Special Flow Shop Problem with Infinite Buffer Capacity. En: Applied Mechanics and Materials Vols. 101-102. 2012. Páginas 379-382.

Fattahi et al.,<sup>27</sup> como una interesante variante a tener en cuenta dentro del problema.

### **1.3 OBJETIVOS**

#### **1.3.1 Objetivo general**

Desarrollar una metaheurística basada en el Algoritmo Competitivo Imperialista (ICA) con el fin de dar solución al problema del *flow shop* híbrido con máquinas paralelas no relacionadas y *buffers* de tamaño limitado.

#### **1.3.2 Objetivos específicos**

- Revisión de la literatura sobre el problema del *flow shop* híbrido con máquinas paralelas no relacionadas y *buffers* de tamaño limitado.
- Formular el modelo de programación matemática que se ajuste el problema.
- Ajustar la estructura del Algoritmo Competitivo Imperialista (ICA) de acuerdo a las necesidades del *flow shop* híbrido con máquinas paralelas no relacionadas y *buffers* de tamaño limitado.
- Implementar el algoritmo adaptado en *MATLAB* y validarlo a través de diferentes instancias.
- Comparar la solución encontrada mediante el modelo de programación matemática desarrollado en *GAMS* con los resultados obtenidos mediante el Algoritmo Competitivo Imperialista (ICA).

---

<sup>27</sup> FATTAHI, P., HOSSEINI, S. M. H., JOLAI, F., TAVAKKOLI-MOGHADDAN, R. A branch and bound algorithm for hybrid flow shop scheduling problema with setup time and assembly operations. En: Applied Mathematical Modelling 38, 2014, Páginas: 119-134.

## 2. MARCO TEÓRICO

### 2.1 OPTIMIZACIÓN

La optimización se puede definir como “el proceso de intentar encontrar la mejor solución posible a un problema, generalmente en un tiempo limitado”,<sup>28</sup> es una disciplina primordial en algunas ciencias como la investigación operativa, la inteligencia artificial y la informática.

Los problemas de optimización pueden ser vistos como aquellos en los que hay muchas posibles soluciones y, además, una clara forma de comparación entre ellas, en consecuencia, un problema de optimización existe sí y sólo sí, hay un conjunto de diferentes soluciones candidatas que se pueden comparar, y el problema, además, puede ser formulado matemáticamente de la siguiente manera:

$$O = \begin{cases} \text{opt } g(x) \\ \text{s. a:} \\ x \in SF \subset SS \end{cases} \quad (1)$$

Donde  $g(x)$  es la función que se quiere optimizar,  $SF$  es el conjunto de soluciones factibles y  $SS$  es el espacio de todas las soluciones.

Existen dos categorías para los problemas de optimización: la primera categoría es aquella para la cual la solución está definida a través de valores reales y la segunda categoría es en la cual las soluciones están agrupadas por valores enteros. Dentro de la segunda categoría existen una clase de problemas particulares denominados problemas de optimización combinatoria que busca encontrar el máximo (o el mínimo) de una función establecida sobre un conjunto finito de soluciones.

---

<sup>28</sup> DUARTE MUÑOZ, Abraham; PANTRIGO FERNÁNDEZ, J. J.; GALLEGO CARRILLO, M. Metaheurísticas. *Madrid: Dykinson, 2007.* p 1.

**2.1.1 Optimización combinatoria.** Se refiere a una rama de la optimización encargada del análisis de problemas que presentan la característica de tener una cantidad finita de soluciones posibles y, trabajar, además, con variables discretas.

La optimización combinatoria se deriva de la optimización matemática discreta,<sup>29</sup> en especial de la parte de la combinatoria de donde es muy común encontrar los conceptos de enumeración, permutación y el de combinación.

La enumeración se define partiendo de establecer un conjunto  $C$  de tamaño  $|C| = m$ , y de dicho conjunto listar de manera exacta todos los elementos que lo componen.

En segundo lugar, una combinación es una recopilación desordenada de distintos elementos generalmente de tamaño  $k$  tomados del grupo  $C$ . Por último, una permutación es una serie o secuencia que contiene elementos de  $k$  distintos entre sí.

En los problemas de optimización combinatoria se tiene como objetivo el encontrar una solución óptima, un valor máximo o mínimo, dependiendo de la función objetivo, dentro de un área de soluciones factibles.

Dentro de los problemas de optimización combinatoria existen dos tipos, el primero, denominado problemas de agrupación, son aquellos que se relacionan con la combinación óptima de elementos. El segundo tipo de problemas de optimización, se conocen como problemas de programación y se relacionan con la permutación óptima de los elementos, dentro de este segundo tipo de problemas se encuentra el problema de programación *flow shop* que mediante el ajuste de la secuencia de puestos de trabajo busca encontrar el tiempo mínimo de procesamiento total.

---

<sup>29</sup> YU, Xinjie; GEN, Mitsuo. Introduction to evolutionary algorithms. Springer Science & Business Media, 2010. p. 263.

Los problemas de optimización combinatoria presentan como particularidad que siempre existe un algoritmo exacto que permite obtener la solución óptima,<sup>30</sup> pero existe un problema, generalmente, dicho algoritmo suele ser poco eficiente pues en la mayoría de los casos el tiempo que tomaría en encontrar una solución crece de manera exponencial con respecto al tamaño del problema. Basándose en lo anterior se puede clasificar según la complejidad computacional a los problemas de optimización combinatoria.

## 2.2 COMPLEJIDAD COMPUTACIONAL

La complejidad computacional representa la dificultad que tiene para un computador u ordenador para solucionar un problema de optimización, esta dificultad influye de manera directa en la eficacia y eficiencia del algoritmo utilizado. De manera más formal se puede definir a la complejidad computacional como “la cantidad de recursos necesarios para efectuar un cálculo”,<sup>31</sup> que además puede ser medido en función de la memoria ocupada y el tiempo de ejecución y cuyo objetivo es “clasificar los problemas en función de la resolubilidad algorítmica práctica de los mismo”.<sup>32</sup>

- El concepto de *memoria ocupada* se entiende como total de memoria que ocupan todas las variables manejadas por el algoritmo, en otras palabras, es la suma del almacenamiento necesario para que se ejecute el mismo y está compuesto tanto por la memoria dinámica como por la memoria estática.
- El *tiempo de ejecución* tiene que ver con el número de datos de entrada del problema y con la velocidad del equipo de computación; se mide mediante la

---

<sup>30</sup> DUARTE, PANTRIGO Y GALLEGO. Op. Cit. p 2.

<sup>31</sup> CHÁVEZ, Marco Antonio Cruz; BERNAL, Pedro Moreno; ABARCA, Jesús del Carmen Peralta. Aplicación de la teoría de la complejidad en optimización combinatoria. *Inventio, la génesis de la cultura universitaria en Morelos*, 2014, no 20, p. 35.

<sup>32</sup> JIMÉNEZ, Mario de J. Pérez; CAPARRINI, Fernando Sancho. *Máquinas moleculares basadas en ADN*. Universidad de Sevilla, 2003. p. 23.

generación de una denominada “función temporal” la cual define el número de comandos ejecutados por el programa cuando resuelve un problema.

Cuando el costo es mínimo con respecto al costo espacial y costo temporal se dice que el algoritmo es eficiente.

**2.2.1 Tipos de complejidad computacional.** Visto desde la teoría de la complejidad, la complejidad de un problema de optimización se puede definir a través de los siguientes elementos:<sup>33</sup>

Un *modo de computación* que de precisión sobre la noción de aceptación de un dato de entrada y que a su vez también fije el significado de resolución de un problema.

Un *modelo de computación* que suministre los dispositivos en los cuales se resolverán los problemas, generalmente para un modelo de computación se consideran las máquinas de Turing que puede ser determinísticas o no determinísticas.

Una *medida de complejidad* que ayude a cuantificar los recursos utilizados por los dispositivos de computación durante la resolución de los problemas.

Por último, una *función total* que sea computable entre números naturales y que sirva de cota superior para los recursos utilizados.

Es posible catalogar, mediante el modelo computacional de Turing, los problemas por su grado de dificultad para resolverlos, a través de dicho modelo se han descrito problemas intratables conocidos como *NP (nondeterministic polynomial time)*, que cuentan con la característica de ser imposibles de resolver en un tiempo razonable cuando el número de variables que los componen son extremadamente

---

<sup>33</sup> JIMÉNEZ, CAPARRINI, Op.cit., p. 24.

grandes en cantidad,<sup>34</sup> dentro de los problemas de la clase *NP* están los que se listan a continuación:

**2.2.1.1 Problemas P.** Son aquellos problemas que pueden ser resueltos mediante una máquina de Turing determinística en un tiempo polinómico, entonces, existe un algoritmo en el que se saben los parámetros de entrada con los cuales se puede dar respuesta al problema.

**2.2.1.2 Problemas NP-Completos.** Son problemas los cuales la solución no ha sido encontrada en su forma exacta en un tiempo polinomial a través de algoritmos determinísticos, por lo tanto, se intentan resolver mediante algoritmos no deterministas, cuya solución deseada sea de complejidad polinómica.

Un problema se considera *NP-Completo* si existe la posibilidad de convertir el problema en otro de tal forma que la solución al segundo problema se pueda utilizar para resolver el primero.<sup>35</sup>

**2.2.1.3 Problemas NP-Duros.**<sup>36</sup> Tienen el mismo o superior nivel de dificultad en su resolución que los problemas *NP-Completos* ya que no existe algoritmo polinomial que permita comprobar una solución.

## **2.3 MÉTODOS DE SOLUCIÓN DE PROBLEMAS DE OPTIMIZACIÓN COMBINATORIA**

Los diferentes algoritmos que se utilizan para solucionar problemas de optimización combinatoria se pueden clasificar en las categorías enumeradas a continuación:

---

<sup>34</sup> JIMÉNEZ, CAPARRINI, Op.cit., p. 24.

<sup>35</sup> IGLESIAS MARTÍNEZ, José; y otros. Complejidad Computacional. Grado Ingeniería informática. Teoría de Autómatas y Lenguas formales.

<sup>36</sup> DUARTE. Op. Cit. p 10.

**2.3.1 Algoritmos exactos.** Se denominan algoritmos exactos a aquellos que garantizan resultados óptimos, pero no tienen tiempo de complejidad polinomial para encontrar la solución a problemas tipo *NP-Completos* o *NP-Duros*. Dentro de esta categoría se destacan los siguientes algoritmos:

**2.3.1.1 Método simplex.** Es un método desarrollado en el año de 1947 por G.B. Dantzig para solucionar problemas de logística como el despliegue de tropas o la entrega de suministros dentro de la fuerza aérea.<sup>37</sup>

El *método simplex* está basado en el método gráfico, el cual establece que la solución óptima a un problema está asociada siempre con un punto en una esquina del espacio de posibles soluciones. El objetivo del *método simplex* es mover la solución hacia una esquina; el método está en capacidad de mejorar el valor de la función tras cada iteración y se detiene al momento de encontrar la solución óptima.

**2.3.1.2 Algoritmo de ramificación y acotamiento.** Desarrollado en el año de 1960 por G. Doig es un esquema metódico que enumera de forma implícita las finitas posibles soluciones factibles a un problema de programación lineal. A pesar de que, teóricamente, el tamaño del árbol de enumeración tendería a aumentar de manera exponencial en los parámetros del problema, en la gran mayoría de los casos el algoritmo elimina la gran mayoría de soluciones factibles.

**2.3.2 Algoritmos heurísticos.** Son todos los métodos los cuales pueden suministrar una buena solución, pero dicha solución no es necesariamente la óptima, esto se debe a que el proceso iterativo puede detenerse al encontrar un óptimo local y por consiguiente no continuar la búsqueda del óptimo global.<sup>38</sup> Entre

---

<sup>37</sup> FISHBACK, Paul E. *Linear and nonlinear programming with Maple: an interactive, applications-based approach*. CRC Press, 2009.

<sup>38</sup> MARTÍ, Rafael; REINELT, Gerhard. *The linear ordering problem: exact and heuristic methods in combinatorial optimization*. Springer Science & Business Media, 2011.

las diversas razones para utilizar métodos heurísticos destacan las siguientes:

- No existe un método exacto para encontrar solución al problema en un tiempo moderado debido a la complejidad del problema mismo.
- La implementación computacional de un método exacto para resolver el problema (si es que existe) es muy costosa.
- Debido a que la flexibilidad del método heurístico es mayor que la del método exacto, existe la posibilidad de incluir condiciones o restricciones de complicada modelación.

**2.3.2.1 Clasificación de los algoritmos heurísticos.** No existe una clasificación exacta para los diferentes algoritmos heurísticos, esto debido a que son de naturaleza muy diferente entre sí, con algunos creados inclusive para problemas en específico, sin embargo, se les puede agrupar en algunas categorías globales como las siguientes:<sup>39</sup>

a. **Métodos constructivos.** son técnicas deterministas que dependiendo de la estrategia utilizada pueden construir una solución a un problema en particular, algunos métodos constructivos son:

- *Métodos de reducción.* Buscan características o propiedades que tienen en común las soluciones buenas para transformarlas en restricciones del problema, esto se hace con el objetivo de limitar el área de soluciones y así simplificar el problema.
- *Métodos de manipulación del modelo (métodos inductivos).* Examinan la manera de simplificar el problema original con el fin de obtener una solución para el problema simplificado, luego, tomando como base la solución encontrada se calcula la respuesta al problema original. Entre algunos ejemplos de éste método destacan: la agrupación de variables, la introducción de nuevas restricciones, la linealización, entre otras.

---

<sup>39</sup> DUARTE, PANTRIGO Y GALLEGU. Op. Cit. p. 27.

- *Estrategia de descomposición.* Esta estrategia toma el problema principal y lo divide en una serie de sub-problemas más sencillos de resolver, dicha división se realiza de manera sucesiva hasta tener un sub-problema con solución trivial.
  - *Estrategia voraz (algoritmo voraz, algoritmo ávido, algoritmo miope).* A partir de una semilla, paso a paso construye una respuesta factible para el problema.
- b. **Método de búsqueda:** Es un método que se caracteriza por empezar desde una solución factible e intentar mejorarla. Algunos ejemplos destacados de este método son:
- *Estrategia de búsqueda local 1.* A través de un análisis de la vecindad mejora gradualmente la solución factible determinando cuál es el primer movimiento que mejoraría la solución actual.
  - *Estrategia de búsqueda local 2.* Examinando la vecindad de la solución factible inicial se busca mejorarla progresivamente evaluando todos los movimientos posibles y sobre esa evaluación se seleccionan los movimientos que produzcan una mejora mayor en la función objetivo.
  - *Estrategia aleatorizada.* La estrategia aleatorizada inicia con una solución factible dada y además de una vecindad inscrita a esa solución, a partir de esto, se seleccionan de manera aleatoria soluciones vecinas dentro de la vecindad.

**2.3.3 Algoritmos metaheurísticos.** El término metaheurística fue acuñado en 1986 por Fred Glover. Con base en Osman y Kelly citados por Rafael Marti,<sup>40</sup> los algoritmos metaheurísticos son “procesos de generación iterativos que guían una heurística subordinada combinando inteligentemente diferentes conceptos para explorar y explotar los espacios de búsqueda usando las estrategias de aprendizaje para estructurar la información con el fin de encontrar soluciones casi óptimas diseñados para resolver problemas difíciles de optimización combinatoria,

---

<sup>40</sup> MARTI, Rafael. Procedimientos metaheurísticos en optimización combinatoria. *Matemáticas*, 2003, vol. 1, no 1, p. 3-62.

en los que los heurísticos clásicos no son efectivos”.

**2.3.3.1 Clasificación de los algoritmos metaheurísticos.** Al igual que los algoritmos heurísticos, no existe una clasificación formal de las metaheurísticas debido a que no hay características en común para todas las mismas, sin embargo, se pueden agrupar desde la implementación real y el enfoque conceptual o estado puro del algoritmo, pero así se hace más complicada la clasificación debido a que considera la falta de homogeneidad existente entre las diferentes metaheurísticas.<sup>41</sup>

Ahora se enumerarán algunas características utilizadas desde la taxonomía clásica para clasificar algoritmos metaheurísticos. Es importante hacer la salvedad de que las diferentes opciones de clasificación descritas no permiten necesariamente establecer una disgregación clara entre toda las metaheurísticas, por lo cual algunos autores toman esto y lo utilizan como referencia para efectuar una clasificación más robusta.

- *Inspiración natural.* Se refiere a aquellos algoritmos que se basan en un comportamiento biológico, social, cultural, entre otros.  
*Sin inspiración.* Algoritmos que se obtiene de manera directa de sus propiedades matemáticas.
- *Número de soluciones.*  
*Poblacionales.* Partiendo de un conjunto de soluciones se realiza una búsqueda de la solución óptima al problema.  
*Trayectoriales.* Para encontrar el óptimo se realiza una mejora iterativa de la respuesta, pero utilizando una sola solución.
- *Función objetivo.*  
*Estáticas.* Aquellas que no hacen ningún cambio o modificación a la función objetivo a través del tiempo de la búsqueda.  
*Dinámicas.* Durante la búsqueda, realizan modificaciones a la función objetivo.

---

<sup>41</sup> DUARTE, PANTRIGO Y GALLEGU. Op. cit. p. 35.

- *Vecindad.*

*Una vecindad.* Se utiliza solamente una estructura de vecindad para buscar la respuesta óptima al problema.

*Varias vecindades.* A través del proceso de búsqueda de la solución óptima se modifican las estructuras de vecindades.

- *Uso de memoria.*

*Sin memoria.* Se basan solamente en el estado anterior.

*Con memoria.* Tienen un esquema de memoria para recordar varios estados pasados.

**2.3.4 Metaheurísticas utilizadas para solucionar problemas de optimización combinatoria.** Se describirán a continuación algunas de las metaheurísticas utilizadas de forma más común.

**2.3.4.1 Búsqueda prohibida (*Tabu Search*).** Desarrollada en 1986 por Glover, se trata de un método de mejora donde se toma en cuenta un salto a la mejor solución de una vecindad, inclusive cuando dicha nueva solución tenga un costo peor que la respuesta actual, esto se hace con el fin de tener opción de escapar de óptimos locales favoreciendo otros mejores.

El algoritmo genera una nueva vecindad llamada “vecindad legal”, la cual se genera por una relación de vecindad de una solución junto a una lista de movimientos prohibidos que son la memoria de últimos movimientos efectuados para sortear repetir soluciones anteriores. La generación de la “vecindad legal” permite aumentar o variar el espacio de búsqueda de soluciones al problema, esto ayuda al algoritmo a evitar quedar estancado en óptimos locales.

**2.3.4.2 Recocido simulado o Temple simulado (*Simulated Annealing*).** Fue introducida por primera vez en los años 1980 por Kirkpatrick et al. El recocido simulado es propuesto como un equivalente al fenómeno físico del temple de un

metal en una fragua.

Se trata de un algoritmo aleatorizado de búsqueda local por vecindades; el método acepta de forma inmediata los movimientos que mejoran la función objetivo y los movimientos que desmejoran la función objetivo tienen una probabilidad de ser aceptados que dependen de un parámetro controlable llamado temperatura, que disminuye a medida que avanzan las iteraciones. Al procedimiento que controla la temperatura se le llama esquema de enfriamiento (*cooling Schedule*) el cual tiene poder en la calidad de la respuesta final.

#### **2.3.4.3 Búsqueda de vecindad variable (*Variable Neighborhood Search*).**

Fue concebida por Hansen y Mladenovic (1999 y 2001). Se trata de un algoritmo que se establece sobre un cambio sistemático de las estructuras de las vecindades dentro de un esquema de búsqueda local para encontrar una solución óptima.

La búsqueda de vecindad variable se fundamenta en tres elementos citados a continuación:

- Un óptimo local dentro de una vecindad no tiene que serlo necesariamente con respecto a otra vecindad.
- Un óptimo global es un óptimo local con respecto a todas las posibles estructuras de vecindad.
- En muchos problemas, los óptimos locales con respecto a una o varias estructuras de vecindad se encuentran relativamente cercanos.

El algoritmo VNS presenta varias categorías que se establecen dependiendo de la forma en la que se combinan los tres elementos citados anteriormente. Las categorías pueden ser:

- a. *Búsqueda del entorno variable básico (Basic Variable Neighborhood Search)*.  
En esta categoría se mezclan elementos de manera estocástica y determinista;

la estrategia comienza seleccionando un punto que pertenezca a una vecindad y se realiza la mejora mediante una estrategia de búsqueda local.

- b. *Búsqueda del entorno variable decente (Variable Neighborhood Descent Search)*. En esta segunda categoría el cambio de vecindad se hace de manera determinística. Cuando el proceso queda atrapado en un óptimo local se debe realizar un cambio en la vecindad para escapar de dicho óptimo.
- c. *Búsqueda del entorno variable reducido (Restricted Variable Neighborhood Search)*. La tercera categoría se caracteriza porque mezcla los tres elementos de forma estocástica. En éste método se escoge de manera aleatoria un punto perteneciente a una vecindad dada, pero sin realizar ninguna clase de optimización local.

**2.3.4.4 Algoritmos evolutivos (Evolutionary Algorithms).** Son algoritmos que tienen la ventaja de que pueden obtener una solución casi óptima, esto se debe a que su rendimiento con respecto al tiempo no se relaciona con el tamaño del problema y de ahí que el tiempo de respuesta se hace relativamente corto.

Los algoritmos evolutivos están compuestos por los siguientes elementos:

- *Población.* Son el conjunto de soluciones candidatas para un problema, también se les llama individuos.
- *Selección.* Es el esquema para la elección de individuos que funciona de manera que se toman los que sean mejores y estos transmitan su código genético a su descendencia.
- *Alteración.* Se trata del mecanismo que asiste la creación de nuevos individuos mediante la modificación de ciertas características de manera estocástica; la mutación de los antiguos individuos se puede realizar de dos maneras: mediante la mutación (o unaria) o mediante la de orden superior (o cruce).

Existe una clasificación de los algoritmos evolutivos en cuatro clases, la clasificación se basa en sus características de implementación. Las categorías son las siguientes:

- a. *Programación evolutiva*. Tiene como principal característica que no realiza cruce de individuos, esto tiene como consecuencia que el mecanismo de mutación se torne mucho más complejo.
- b. *Estrategias de evolución*. El segundo tipo de algoritmos evolutivos inicializa y luego evalúa una población para escoger a los padres de manera uniforme y aleatoria. La combinación de dos padres produce una cantidad de hijos mayor que los padres, dichos hijos se modifican mediante la implementación de una etapa de mutación.
- c. *Programación genética*. Este tipo busca edificar programas informáticos a través del uso de programas de generación aleatoria y de los principios de los algoritmos evolutivos.
- d. **Algoritmos genéticos (Genetic Algorithm)**. Fueron introducidos en el año 1975 por Holland. Tienen su fundamento en la teoría de Darwin de la evolución de las especies y se tratan de procedimientos iterativos sobre una población de individuos, cada individuo recibe el nombre de cromosoma y representa una posible respuesta al problema, a su vez, los cromosomas están compuestos por unidades discretas llamadas genes.

Los algoritmos genéticos cuentan con ciertos elementos que los caracterizan y que, a su vez, los diferencian de otros tipos de algoritmos genéticos, dichos elementos son:

- *Población inicial*. Se denomina población inicial a una generación aleatoria de respuestas al problema planteado.
- *Representación*. Se trata de la relación entre la codificación de las variables (genotipo) y las soluciones factibles (fenotipo), la relación se puede desarrollar a través de cadenas binarias, de palabras o de números.
- *Función de evaluación*. También denominada *fitness*, la función de evaluación existe con el objetivo de medir la calidad de los individuos (cromosomas) de una población.

- *Operadores genéticos*. Son los encargados de generar descendencia o nuevos individuos con cierta probabilidad, existen dos métodos principales:
  - *Mutación*. Se llama mutación al cambio de forma aleatoria de un individuo o cromosoma dentro de la población, la mutación se utiliza para preservar la diversidad de las respuestas y también con el fin de ampliar la exploración del espacio de posibles soluciones.
  - *Cruce*. El cruce se realiza seleccionando dos individuos dentro de una población inicial, dichos individuos reciben el nombre de padres e intercambian parte de sus genes para generar descendencia.
- *Selección*. Se trata del esquema que permite seleccionar con una mayor probabilidad a los individuos que posean el mejor valor de la función de evaluación. Los métodos más comunes de selección son:
  - *Torneo*. El método del torneo se trata de tomar  $p$  individuos de una población tamaño  $n$ , luego de tomarlos, se selecciona a aquellos que tengan el mejor valor de la función de evaluación.
  - *Método de la ruleta*. Es una analogía al juego de la ruleta de los casinos, a cada individuo de la población se le asigna una porción de la ruleta, el tamaño de dicha porción es proporcional al valor de la función de evaluación del individuo y también al promedio de las funciones de evaluación de los demás individuos de la población.

**2.3.4.5 Algoritmo competitivo imperialista (*Imperialist Competitive Algorithm, ICA*)**. El *ICA* es uno de los algoritmos evolutivos planteados de manera más reciente; fue propuesto por Athaspaz-Gargar y Lucas en 2007 y por Rajanioun et al. en 2008.

Su mecanismo se basa en la evolución social y política de la raza humana, haciendo especial énfasis para plantear estrategias de optimización en el

fenómeno del imperialismo, los métodos de conquista y la competencia imperialista.

El imperialismo puede ser entendido como la política de expandir las características socio-culturales y el poder de un país; este proceso se desarrolla a través del dominio y la conquista de otros países a los cuales se les llama colonias, dichas colonias, tienen a adoptar las principales características del país conquistador.

La idea del imperialismo es hacer cada vez más fuerte al país imperialista, pues así tiene mayores posibilidades de conquistar más países, en cambio, si el imperialista es débil, corre mucho más riesgo de perder sus colonias.

El concepto de imperialismo deriva en la idea de imperio, que es la unión del país imperialista junto con sus colonias; a su vez, el imperio cuenta con un poder que es la suma del poderío del país imperialista con el poderío de sus colonias, dicho poder es el que ayuda a los imperios más poderosos a conquistar los imperios con menor poder.

El *ICA*,<sup>42</sup> emprende su funcionamiento con una población inicial de individuos también llamados países, luego, los países con mejor valor de aptitud son elegidos para ser imperialistas, mientras el resto de países serán las colonias; las mencionadas colonias son repartidas entre los países imperialistas dependiendo del poderío de cada uno de ellos.

Posteriormente, tras dividir las colonias entre los imperialistas, éstas se mueven hacia su respectivo país imperialista, luego, se da inicio a la competencia entre imperios, se eliminará aquel imperio que no tenga la capacidad de incrementar su poder o al menos, mantener su tamaño. Tras el desarrollo de la competencia se

---

<sup>42</sup> ATASHPAZ-GARGARI, Esmail; LUCAS, Caro. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. En *Evolutionary computation, 2007. CEC 2007. IEEE Congress on*. IEEE, 2007. p. 4661-4667.

obtendrá como resultado un aumento gradual del poder de los imperios más poderosos y la caída de los más débiles.

Para observar de manera más detallada las etapas del algoritmo competitivo imperialista se presenta la figura 1.

1. *Generación de imperios iniciales.* Dentro de un problema de optimización con una dimensión de  $N$  variables, un país, o en términos de algoritmos genéticos: un cromosoma, es un conjunto de  $1 \times N_{var}$  y se puede entender de la siguiente manera:

$$Pais = [p_1, p_2, p_3 \dots \dots p_N] \quad (2)$$

Si se evalúan las variables  $p_i$  en la función de costo  $f$ , se obtendrá el costo de un país como muestra la ecuación (3):

$$Costo = f(p_1, p_2, p_3 \dots \dots p_N) \quad (3)$$

En la siguiente página, se detallarán cada uno de los diferentes pasos del ICA mediante la figura 1.

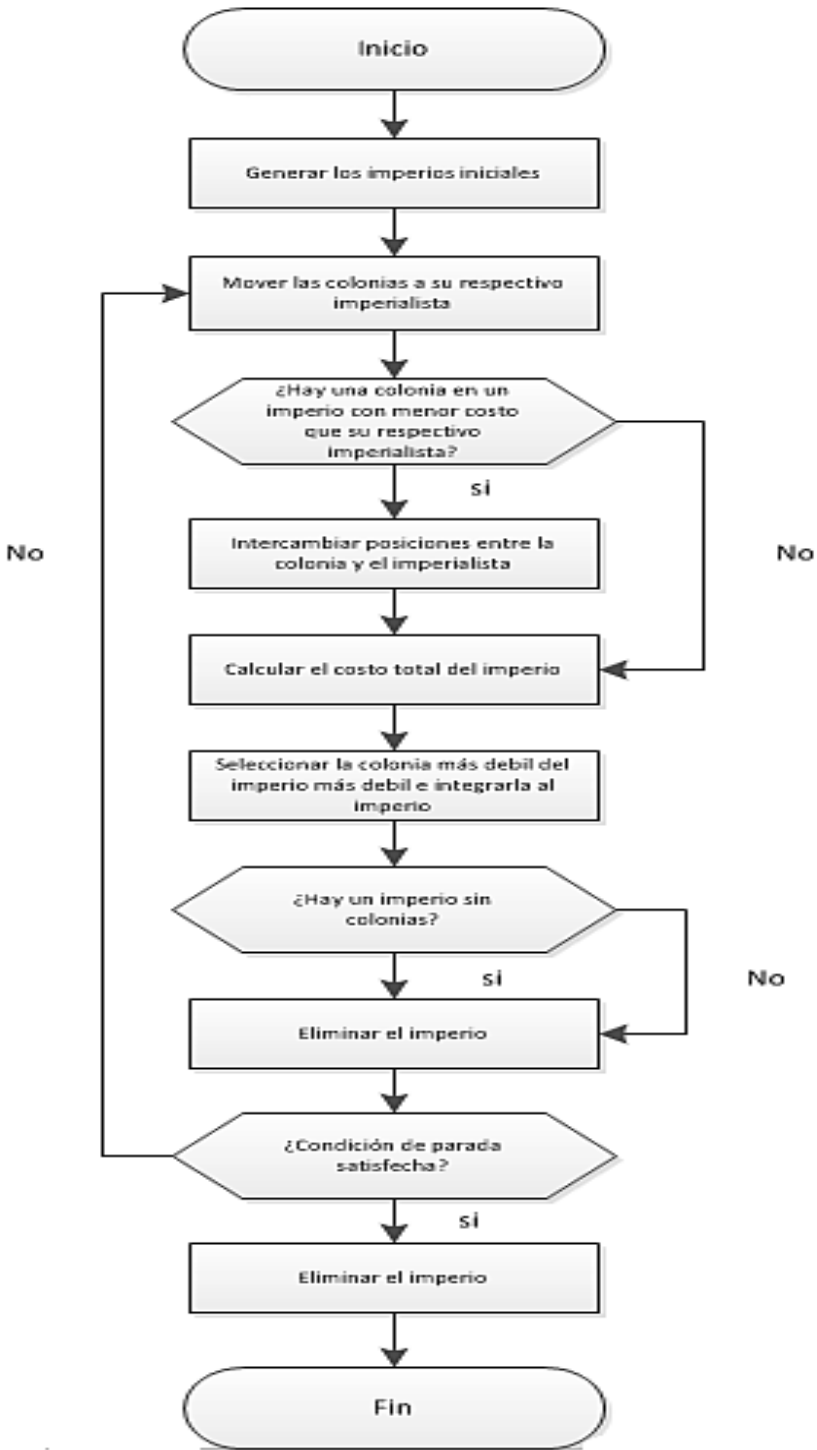
Para iniciar, se genera de forma aleatoria una población con tamaño  $N_{pop}$ , de dicha población se seleccionan  $N_{imp}$  países imperialistas y  $N_{col}$  colonias. Para asignar las colonias a los países imperialistas se toma como referencia el costo normalizado y el poder normalizado, ambos se pueden calcular siguiendo las ecuaciones (4) y (5) que se presentan a continuación:

$$C_n = c_n - \max\{c_i\} \quad (4)$$

$$P_n = \left| \frac{C_n}{\sum_{i=1}^{N_{imp}} C_i} \right| \quad (5)$$

Donde  $c_i$  es el costo del  $i$ -ésimo imperialista y  $C_n$  es el costo normalizado.

Figura 1. Diagrama de flujo del Algoritmo Competitivo Imperialista



Fuente: ATASHPAZ, Esmail; LUCAS, Caro. Op. cit. p. 4462

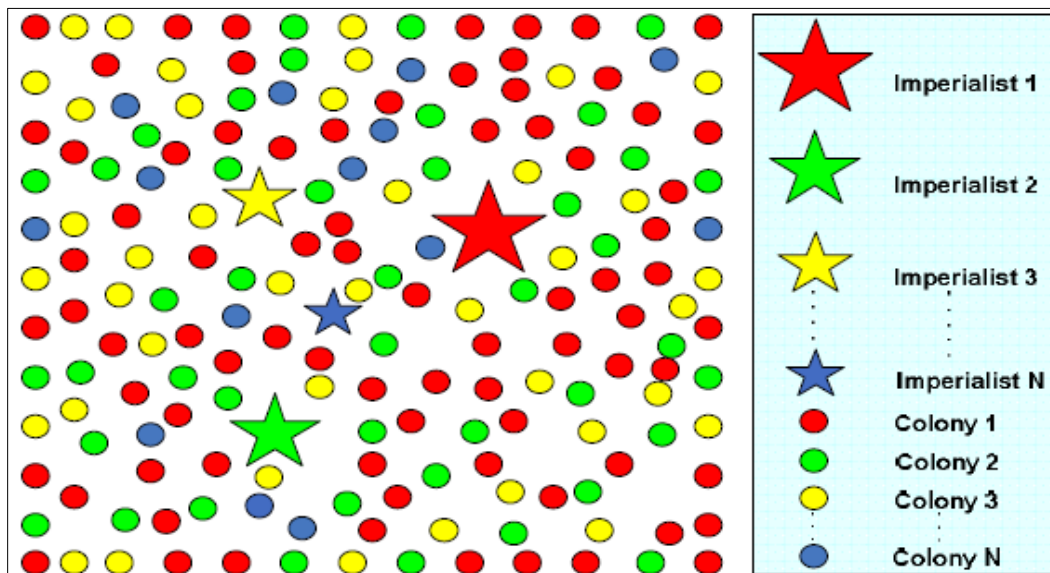
Basándose en las ecuaciones (4) y (5), para calcular el número inicial de colonias para un imperio se utiliza la siguiente ecuación matemática:

$$NC_n = \text{round}\{P_n * N_{col}\} \quad (6)$$

Ahora bien, para finalmente asignar las colonias a cada imperialista, se toman de forma aleatoria  $NC_n$  colonias de toda la población; dichas colonias junto con su imperialista formarán la población inicial de un imperio.

Como se puede observar en la figura 2, el país imperialista que tiene el mayor poder será el que mayor número de colonias tendrá.

**Figura 2.** Población inicial de países imperialistas junto a sus colonias.



**Fuente:** ATASHPAZ, Esmail; LUCAS, Caro. Op. cit. p. 4463

2. *Movimiento de las colonias hacia el imperialista.* El segundo paso del ICA se refiere al mejoramiento de las colonias que empieza el imperialista.

Una colonia se mueve  $x$  unidades hacia su imperialista, un vector desde la colonia hacia el imperialista es el que detalla la dirección del movimiento. La

variable  $x$  que se mencionó anteriormente es una variable aleatoria que sigue una distribución uniforme como la siguiente:

$$x \sim U(0, \beta * d) \quad (7)$$

Se define a  $d$  como la distancia entre el imperialista y la colonia y a  $\beta$  como un número mayor que 1. El sentido de que  $\beta > 1$  es asegurar que las colonias se aproximen al imperialista desde ambos lados. Para expandir la búsqueda de diferentes soluciones alrededor del imperialista se suma una cantidad aleatoria de desviación a la dirección del movimiento llamada  $\theta$ . Dicha cantidad aleatoria sigue una distribución uniforme entre  $-\gamma \leq \theta \leq \gamma$ , que corresponde a un parámetro que modifica la desviación desde la dirección original.

3. *Intercambio de posiciones entre el imperialista y una colonia.* Puede suceder el caso que, durante el movimiento de una colonia hacia su respectivo imperialista, la misma halle una posición con menor costo que el del imperialista, en tal caso de que esto suceda, el imperialista se mueve hacia la posición de la colonia y viceversa.
4. *Cálculo del poder total de un imperio.* El poder total de un imperio está conformado tanto por el poder del país imperialista como por el poder de sus colonias, todas las colonias tienen influencia sobre el poder total sin importar que tan insignificante sean las mismas. La siguiente ecuación define el costo total del imperio:

$$TC_n = \text{costo}(\text{imperialista}_n) + \xi [\text{Costo promedio}(\text{colonias del imperio}_n)] \quad (8)$$

Donde  $TC_n$  es el costo total del  $n$ -ésimo imperio y  $\xi$  es un número positivo pero menor que 1. El efecto de  $\xi$  se ve en que si el valor de este parámetro es pequeño entonces el poder total de imperio se verá poco influenciado por las colonias y dependerá casi en su totalidad del país imperialista, mientras si  $\xi$  es

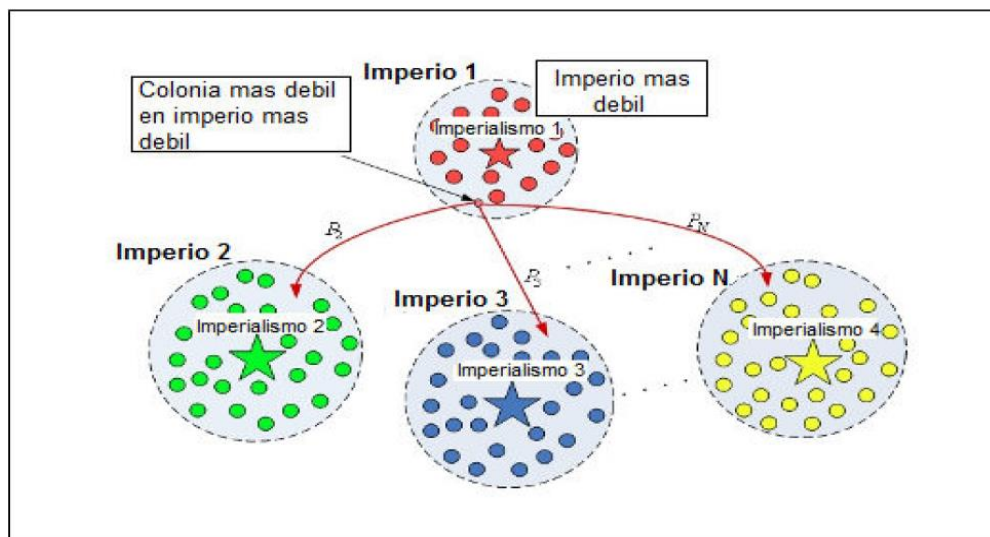
cercano a 1 entonces las colonias tendrán una influencia fuerte sobre el costo total del imperio.

5. *Competencia imperialista.* En el paso 5 se desarrolla la competencia entre imperios, esto sucede porque los imperialistas intentarán tomar posesión de las colonias de otros imperios y controlarlas, la competencia traerá como resultado la reducción del poder de los imperios más débiles y el aumento de poder de los imperios más fuertes.

La competencia imperialista comienza con la selección de la colonia más débil del imperio más débil, los imperios competirán por obtener dicha colonia, la secuela de la competencia dependerá de la probabilidad que tenga un imperio de arrebatarse la posesión de una colonia de otro imperio y dicha probabilidad es proporcional al poder del imperio. En la figura 3 se puede observar el proceso de competencia imperialista.

6. *Dstrucción del imperio más débil.* Como consecuencia del paso anterior, el imperio más débil al final de la competencia colapsará y sus colonias serán repartidas entre los imperios restantes.

**Figura 3.** Competencia entre los imperialistas.



**Fuente:** ATASHPAZ, Esmail; LUCAS, Caro. Op. cit. p. 4464

## 2.4 PROGRAMACIÓN DE OPERACIONES

Se entiende como programación de operaciones o *scheduling* al proceso de asignación de actividades y recursos con el fin de lograr optimizar una o varias medidas de desempeño.

Cuando se habla de *scheduling* es importante hacer una salvedad sobre su diferencia con el término secuencia. El *scheduling* se refiere a la asignación de puestos de trabajo en un entorno complejo con ajustes de máquina, esto quiere decir, por ejemplo, que una tarea tenga preferencia sobre otras asignadas previamente a una máquina. Por su parte la secuenciación corresponde a una permutación del orden de los trabajos a procesarse en una máquina ya definida.

De acuerdo en lo enunciado por Heizer y Render, el *scheduling* presenta una importancia de tipo estratégico por tres principales razones:

- Ayuda en el cumplimiento de compromisos y a realizar entregas fiables.
- Favorece el movimiento más veloz de bienes y servicios a través de las instalaciones; el efecto de esto, es poder aprovechar de mejor manera la capacidad instalada y así, lograr disminuir costos.
- Permite lograr capacidad adicional, un proceso de fabricación más rápido y tener una flexibilidad relativa del proceso, estos tres beneficios representan a la larga una mejoría en el nivel de servicio al cliente.

**2.4.1 Clases de Schedule.** El proceso de *scheduling* se puede clasificar en tres clases principales que se citan a continuación:

**2.4.1.1 Schedule sin retraso.** El *schedule* sin retraso es aquel en el que no existe máquina inactiva mientras una tarea está en espera para ser procesada.

**2.4.1.2 Schedule activo.** Se presenta *schedule* activo cuando no existe posibilidad de construir otro calendario de trabajo haciendo cambios en el orden de procesamiento de las máquinas con por lo menos una operación finalizando antes de tiempo y ninguna operación finalizando retasada.

**2.4.1.3 Schedule semi-activo.** Existe un *schedule* clasificado como semi-activo cuando una operación no puede ser terminada antes del tiempo programado sin que existan cambios en el orden de procesamiento en cualquiera de las máquinas.

**2.4.2 Tipos de problema de *scheduling*.** Así como con las clases de *scheduling*, los tipos de problemas también puede ser clasificados, la cual se hace tomando como fundamento en la configuración del sistema productivo. La configuración del sistema producto es la manera como los operarios, máquinas y centros de trabajo se organizan para procesar los trabajos. A continuación, se presentan los diferentes tipos de problemas de acuerdo a la configuración donde se presentan:<sup>43</sup>

**2.4.2.1 Project scheduling.** Se utiliza en talleres organizados por proyectos, en este caso, la programación de operaciones es la encargada de la secuencia, pero enfocada a los trabajos con restricciones de precedencia y asignación de recursos.

En la organización de las tareas se utilizan principalmente dos técnicas de programación: la primera, la técnica *CPM (Critical Path Method)*, que se utiliza en proyectos con actividades determinísticas y, en segundo lugar, la técnica *PERT (Programa de Evaluación y Revisión Técnica)* utilizada en proyectos con actividades estocásticas.

---

<sup>43</sup> YAGMAHAN, Betül; YENISEY, Mehmet Mutlu. Scheduling practice and recent developments in flow shop and job shop scheduling. En *Computational intelligence in flow shop and job shop scheduling*. Springer Berlin Heidelberg, 2009. p. 261-300.

**2.4.2.2 Single machine scheduling.** Es la secuenciación de varias tareas, pero en una sola máquina. En este caso, la programación se puede enfocar a algunas características de los trabajos como los tiempos de procesamiento.

**2.4.2.3 Parallel machine scheduling.** Es una ampliación del *single machine scheduling*, en esa variante del problema existen varias máquinas del mismo tipo (pero no necesariamente idénticas) en una estación. Además, cada tarea puede ser procesada por cualquiera de las mencionadas máquinas.

Las decisiones a tomar en esta clase de *scheduling* se relacionan con determinar a cuáles máquinas asignar los trabajos y también con establecer una secuencia de procesamiento a las tareas asignadas a cada máquina. Los procesadores dentro de la misma estación pueden ser de tres diferentes clases:

- *Máquinas idénticas:* Todas y cada una de las máquinas de la estación tienen idénticas especificaciones, y no hay diferencia en el procesamiento de trabajos entre las distintas máquinas.
- *Máquinas uniformes:* Las máquinas tienen diferentes velocidades de procesamiento ( $s_j$ ), y los trabajos tienen unos requerimientos especiales de procesamiento ( $p_i$ ), así, el tiempo de procesamiento de una máquina se define como  $p_i/s_j$  unidades de tiempo.
- *Máquinas no relacionadas:* Cada una de las tareas tiene diferentes tiempos de procesamiento en cada una de las distintas máquinas de la estación.

**2.4.2.4 Open shop scheduling.** Se presenta cuando la secuenciación se hace para una serie de  $m$  máquinas y cada una de las tareas debe pasar por cada una de las máquinas, sin embargo, el orden de procesamiento no tiene importancia.

**2.4.2.5 Flow shop scheduling.** Presenta como característica principal que todas tareas tienen un mismo orden de procesamiento, por lo tanto, todos los trabajos deben pasar por un grupo de máquinas que realizan diferentes etapas de procesamiento bajo un mismo orden.

Las máquinas en un *flow shop* están dispuestas de manera lineal, y debido a esto, el procesamiento de una tarea está compuesto por una lista ordenada de actividades donde la *i*-ésima actividad de cada etapa de trabajo está determinada por la misma máquina requerida y el tiempo de procesamiento en ella.<sup>44</sup>

Al plantearse un problema tipo *flow shop* se debe partir de los siguientes supuestos:

- El grupo de  $n$  tareas a procesar, se encuentra disponible desde el instante cero.
- Los detalles y características de las diferentes tareas son conocidos en avance.
- Los tiempos de alistamiento son dependientes de la secuencia.
- Distintas máquinas se encuentran continuamente disponibles.

## **2.5 FLOW SHOP HÍBRIDO**

Se trata de una variante del *flow shop* original. También se le conoce como *flow shop* con múltiples procesadores, *flow shop* compuesto o *flow shop* flexible pero realmente el término *flow shop* híbrido es el más adecuado, pues se trata de un híbrido de dos diferentes sistemas productivos, el primero, una estación con varias máquinas en paralelo y el segundo, el *flow shop* original.

El *HFS*, por sus siglas en inglés, inicia con un grupo de tareas a ser procesadas por una serie de  $m$  estaciones que deben cumplir las siguientes características:

---

<sup>44</sup> BŁAŻEWICZ, Jacek, et al. *Handbook on scheduling: from theory to applications*. Springer Science & Business Media, 2007.

- Todas las tareas deben pasar por las estaciones de procesamiento bajo un mismo orden.
- El número de estaciones de procesamiento  $m$ , debe ser al menos 2.
- Cada una de las etapas de procesamiento, tiene  $m(k)$  máquinas en paralelo y por lo menos una de las etapas debe tener  $m(k) > 1$  procesadores.
- Existe la posibilidad de que un trabajo pueda omitir cualquier número de estaciones de trabajo, siempre y cuando sea procesado en por lo menos una de las etapas globales.

**2.5.1 Notación del problema.** En la notación propuesta por Graham et al. y enunciada por Blazewicz,<sup>45</sup> el *HFS* puede ser clasificado basándose en tres parámetros,  $\alpha, \beta, \gamma$ , una descripción detalla de cada uno se hace ahora:

$\alpha$ : Describe el tamaño y tipo del taller. El parámetro  $\alpha$  está compuesto por cuatro elementos,  $\alpha_1$ , se refiere a la configuración general del taller,  $\alpha_2$ , detalla el número de estaciones del taller, mientras tanto  $\alpha_3$  y  $\alpha_4$ , describen las propiedades de los procesadores en las distintas estaciones.

Alfa cuenta con algunas clasificaciones entre las que destacan:

- $Fm$ : *Flow shop* simple con  $m$  máquinas paralelas
- $F(k_1, k_2, \dots, k_n)$ : *Flow shop* híbrido con  $k_i$  máquinas paralelas idénticas en la etapa  $i$ .

$\beta$ : El segundo parámetro  $\beta$ , es el encargado de enumerar las características del taller, dichas características provienen del problema del *flow shop* simple:

Entre las más utilizadas están:

- *Fechas de liberación* ( $r_j$ ): Se refiere al instante en el cual las tareas pueden empezar a ser procesadas.

---

<sup>45</sup> BŁAŻEWICZ. Op. Cit. p., 5

- *Precedencia (prec)*: Existe la característica que algunos trabajos deben ser completados antes de que otros puedan empezar a ser procesados.  
Si una tarea tiene al menos un predecesor y al menos un sucesor, entonces, a dicha tarea se le conoce como *chair*, en el caso que el trabajo tenga al menos un sucesor se le denomina *intree* y, por último, si la tarea tiene por lo menos un predecesor se le conoce como *outtree*.
- *Preemption (pmtn)*: Detalla si el procesamiento de una tarea en una máquina puede interrumpirse o inclusive ser terminado de forma posterior en otra máquina.
- *No espera (no-wait)*: Existe en caso de que los trabajos no puedan ser almacenados en zonas intermedias entre dos estaciones de trabajo sucesivas.

$\gamma$ : El tercer y último parámetro es en el encargado de describir el criterio a ser maximizado o minimizado en el problema, para el caso del *flow shop* las funciones objetivo más comunes son:

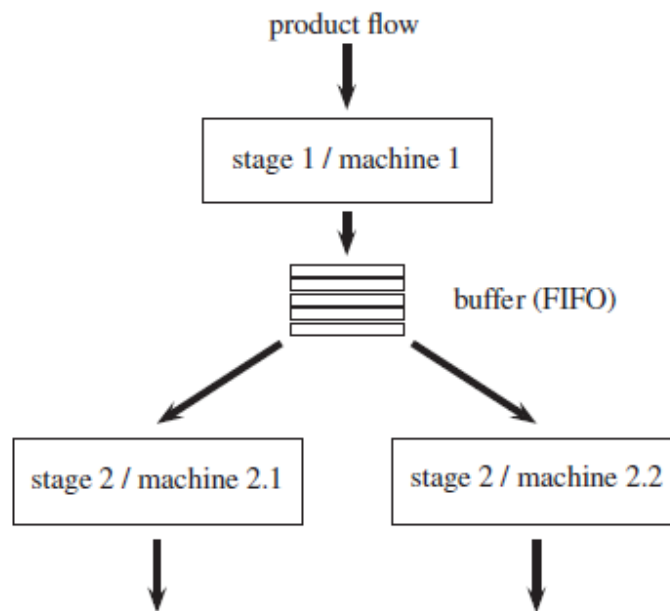
- *CMax*: También conocido como *makespan*, se trata del plazo máximo de ejecución de una tarea.
- *LMax*: Corresponde al *maximum lateness*, y se define como la diferencia entre el tiempo de finalización de un trabajo y su fecha de vencimiento.
- *Tardanza ponderada*: Conocida en inglés como *weighted tardiness*, se refiere al tiempo que una tarea excede a la fecha de entrega planeada.

**2.5.2 Buffers limitados.** Es una restricción de los problemas tipo *flow shop*, se conocen también como zonas de almacenamiento limitadas.

Los *buffers* pueden ser entendidos como zonas de almacenamiento transitorias entre dos etapas de procesamiento consecutivas en una línea de producción. Dicha zona de almacenamiento tiene como tarea albergar los trabajos en proceso durante un tiempo limitado para luego ser transportados a la siguiente estación de procesamiento. En la figura 4 se muestra el esquema de funcionamiento básico de los *buffers*.

Los *buffers* son útiles en dado caso que un trabajo salga de una estación y la siguiente etapa se encuentre ocupada o no pueda procesar el trabajo en cuestión aún. Para esa situación, la tarea debe ser almacenada en un *buffer* y apenas esté libre la siguiente etapa de trabajo, la tarea sale de la zona de almacenamiento hacia el siguiente procesador. La disciplina de salida de tareas dentro del *buffer* generalmente se asume como *FIFO*,<sup>46</sup> entonces, los primeros trabajos en entrar a ser almacenados son los que tienen prioridad de salir a la siguiente estación.

**Figura 4.** Esquema básico del funcionamiento de una línea *flow shop* con *buffers*.



**Fuente:** ALMEDER, Christian; HARTL, Richard F. Op. Cit.

Existen casos donde la capacidad de la zona de almacenamiento intermedia es limitada y puede darse la situación donde una tarea necesite ser almacenada momentáneamente en el *buffer* y dicho sitio se encuentre a su capacidad máxima. Cuando ocurre esto, se presenta el fenómeno del bloqueo,<sup>47</sup> donde básicamente

<sup>46</sup> ALMEDER, Christian; HARTL, Richard F. Op. Cit. p. 88-95.

<sup>47</sup> RASHIDI, E., JAHANDAR, M., ZANDICH, M. Op. Cit. p. 1129-1139.

la máquina de donde proviene el trabajo sufre un “bloqueo” de su capacidad de procesar nuevos trabajos hasta que el *buffer* o la siguiente etapa de procesamiento se desocupen y la tarea pueda salir del procesador.

El asumir los *buffers* como una etapa de procesamiento extra con máquinas especiales de tiempo de alistamiento cero y tiempo de procesamiento cero para cualquier trabajo en cualquier secuencia<sup>48</sup> es el método utilizado de forma más común para la inclusión de ésta restricción dentro del modelo de programación lineal del *flow shop* híbrido.

---

<sup>48</sup> MIRANDA LUGO, P.L. et al. Um modelo de programação inteira mista para a programação da produção em flowshop híbrido com buffers limitados. En: Simpósio Brasileiro de Pesquisa Operacional, 2013, p. 154-165.

### 3. REVISIÓN DE LITERATURA

#### 3.1 ANÁLISIS BIBLIOMÉTRICO

El presente análisis bibliométrico pretende mostrar una descripción de la evolución investigativa que ha tenido el tema del *flow shop* híbrido, se hace énfasis en aquellos artículos que incluyen la restricción de *buffers* intermedios de tamaño limitado y también las características de máquinas paralelas no relacionadas y tiempos de alistamiento dependientes de la secuencia.

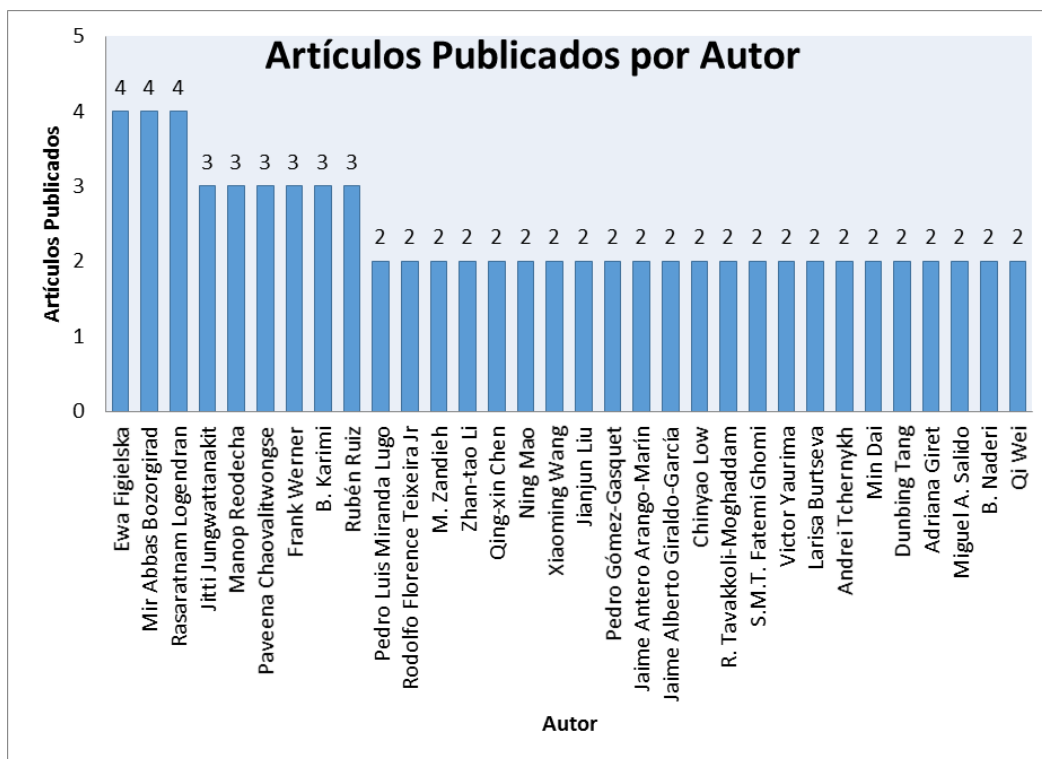
La búsqueda se ha realizado en las bases de datos: *Proquest, Elsevier, Scopus, Springer Link y Web of Science*. Durante la exploración de literatura, se ha dado preferencia a artículos publicados a partir del año 2014; se tuvieron en cuenta artículos investigativos, actas de conferencia y capítulos de libros; se encontraron investigaciones en inglés, español y portugués para obtener un total de 74 artículos a analizar.

La ecuación de búsqueda utilizada incluye los siguientes términos: ((*“hybrid flow shop” OR “flexible flow shop”*) AND *“unrelated parallel machine”* AND *“limited buffers”*). A continuación, se presentarán algunas estadísticas de los artículos estudiados.

Los 74 artículos analizados para la revisión fueron escritos por 150 autores, entre colaboraciones y artículos individuales. En la gráfica 1 se muestran los autores con mayor número de artículos publicados.

Los autores con más aportes investigativos son Figielska, Bozorgirad y Logendran con 4 publicaciones cada uno de ellos; con 3 investigaciones aparecen autores como Jungwattanakit, Reodecha, Chaovalitwongse, Werner, Karimi y Ruiz, además aparecen 23 autores con 2 publicaciones y 118 investigadores con un solo artículo publicado.

**Gráfica 1.** Gráfico de artículos publicados por autor

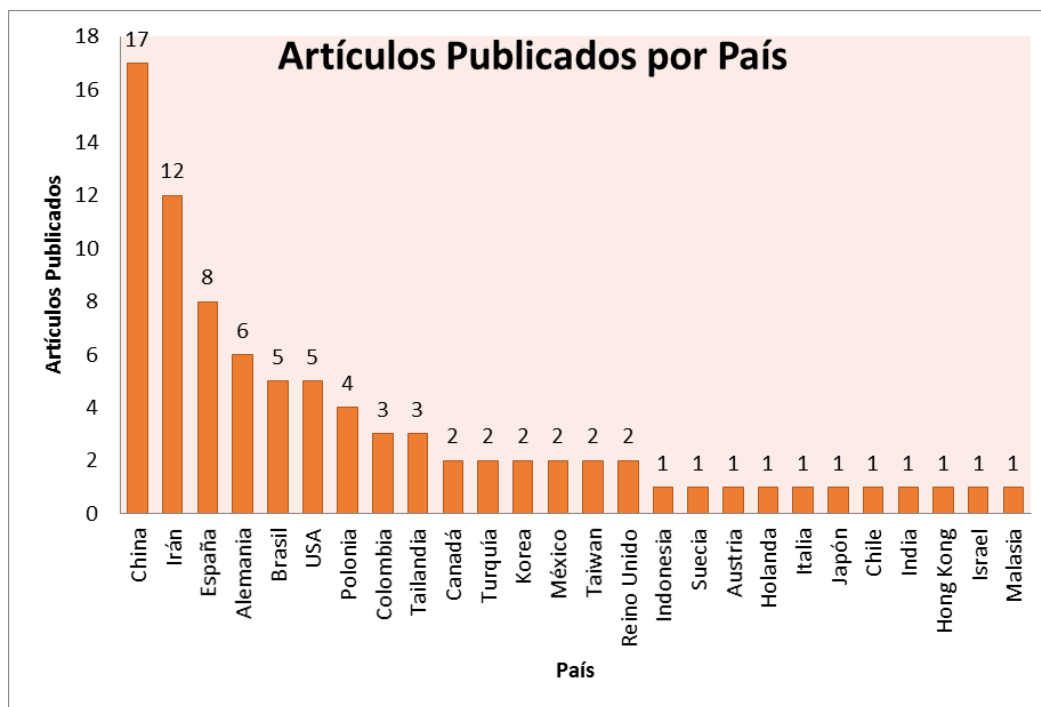


Como autores colombianos destacados se presentan Jaime Giraldo y Jaime Arango con 2 publicaciones, y con una publicación cada uno aparecen: Juan C. López, Omar Castrillón, Javier Parra y finalmente Gonzalo Mejía.

Para el análisis geográfico del origen de los artículos presentado a continuación, se tomó como referencia la nación donde fue realizada la investigación, es decir, el centro de investigación donde se generó el artículo, en la gran mayoría de los casos el centro de investigación estaba ubicado en el mismo país de origen de los autores y en el caso de investigadores de diverso origen para un mismo artículo, se tuvieron en cuenta los centros de trabajo de todos los autores.

En la gráfica 2 se presentan los países con artículos publicados de *Flow Shop* híbrido que se tuvieron en cuenta para la investigación y el número de investigaciones originadas en cada nación.

**Gráfica 2.** Gráfico de artículos publicados ordenados por país



Las investigaciones fueron originadas entre 26 naciones de los continentes asiático, europeo y americano. Los países con mayores aportes fueron China con 17 artículos publicados, Irán con 12 investigaciones, España con 8, Alemania con 6 y Brasil con 5 artículos.

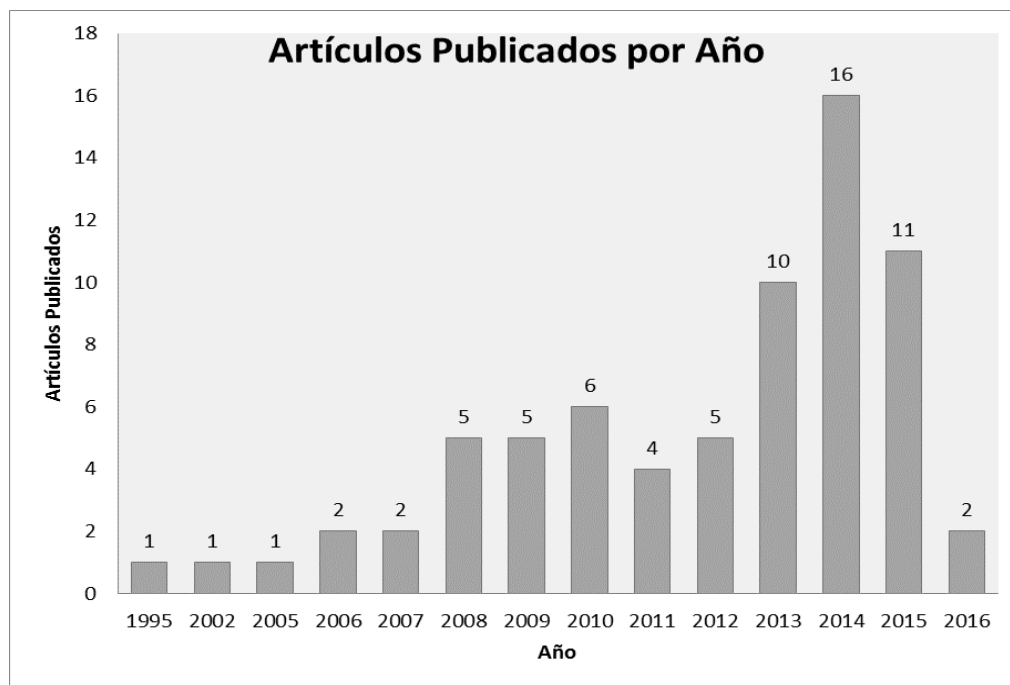
Colombia destaca en la investigación del *HFS* con 3 publicaciones, siendo el tercer país de América con más aportes, el segundo de América latina con más investigaciones publicadas y el octavo a nivel global dentro de los artículos encontrados.

De acuerdo a lo analizado, se observa un aumento consistente del número de publicaciones desde mediados de la década del 2000, también, en los últimos años dicho aumento se ha intensificado, pues, en 2014 se publicaron 16 artículos, en 2015 11 artículos y en 2013, 10 investigaciones. De lo corrido de 2016 se han encontrado 2 artículos.

La investigación de mayor antigüedad analizada es del año 1995, la siguiente es de 7 años después, del año 2002, luego pasaron 3 años hasta 2005 para encontrar la siguiente y a partir de dicho año se encuentran publicaciones todos los años hasta el presente.

En la gráfica 3 se observarán los artículos publicados por año, ordenados de manera cronológica.

**Gráfica 3.** Gráfico de artículos publicados por año



Finalmente, con respecto al medio por el que se publican los artículos utilizados para la investigación destacan dos revistas sobre el resto: la primera, *International Journal of Advanced Manufacturing Technology* con 10 investigaciones publicadas, y en segundo lugar *Computers & Operations Research* con 8 publicaciones.

Tras dichas revistas, aparecen con 4 artículos publicados cada una: *International Journal of Production Economics*, *Computers & Industrial Engineering* y *Expert Systems with Applications*.

En la gráfica 4 se muestran las revistas con mayor cantidad de publicaciones dentro de los artículos tenidos en cuenta para la revisión de literatura.

**Gráfica 4.** Gráfico de artículos publicados ordenados por revista



### 3.2 ANÁLISIS DE LA LITERATURA

El problema del *flow shop* híbrido se puede dividir en dos corrientes principales: la primera, considera máquinas paralelas idénticas y sus estudios iniciaron junto con el problema original en 1954 cuando Johnson<sup>49</sup> lo abordó considerando dos y tres etapas de procesamiento; la segunda corriente, considera el *flow shop* híbrido con máquinas paralelas no relacionadas, que se popularizó en 1995 con el trabajo de Uetake et al.<sup>50</sup> con dos etapas de producción, desde ese momento el problema con máquinas paralelas no relacionadas ha recibido mayor atención con

<sup>49</sup> JOHNSON, S.M. Optimal two-and three-stage production schedules with setup times included. En: Naval Research Logistics Quarterly. 1954, no 1. p. 61-68.

<sup>50</sup> UETAKE, Toshifumi; TSUBONE, Hitoshi; OHBA, Masaaki. A production scheduling system in a hybrid flow shop. En: International journal of production economics, 1995, vol. 41, no 1, p. 395-398.

revisiones de literatura destacas como las de Morais et al.<sup>51</sup> y Wang<sup>52</sup> que concluyen, respectivamente, el bajo porcentaje de investigación de la segunda corriente con respecto a la primera y cuáles son los métodos de solución comúnmente utilizados.

Con base en Linn y Zahng<sup>53</sup> el *flow shop* híbrido se puede clasificar en tres clases principales en términos de su complejidad. La primera es el *HFS* con dos etapas que presenta a su vez tres variantes: la primera variante con una sola máquina en la primera etapa y  $m$  máquinas paralelas no relacionadas en la segunda; esta primera variante es estudiada por Almeder y Hartl<sup>54</sup> en un caso de aplicación en la industria metalúrgica considerando zonas intermedias de almacenamiento de tamaño limitado y anteriormente por Uetake et al.<sup>55</sup> con la consideración de procesamiento por lotes o familias.

La segunda variante presenta  $m$  máquinas paralelas no relacionadas en el primer *stage* y una sola en el segundo, es trabajada por Low et al.<sup>56</sup> que consideran restricciones de precedencia y elegibilidad de máquinas y Figielska<sup>57,58</sup> en dos trabajos donde considera preferencia de tareas y recursos adicionales limitados.

---

<sup>51</sup> DE FATIMA MORAIS, Marcia, et al. Op. Cit. p. 1004-1031.

<sup>52</sup> WANG, Wenpeng. Review on Hybrid Flow Shop Scheduling. En: Information Technology, Computer Engineering and Management Sciences (ICM), 2011 International Conference on. IEEE, 2011. p. 7-10.

<sup>53</sup> LINN, Richard; ZHANG, Wei. Hybrid flow shop scheduling: a survey. En: Computers & industrial engineering, 1999, vol. 37, no 1, p. 57-61.

<sup>54</sup> ALMEDER, Christian; HARTL, Richard F. A metaheuristic optimization approach for a real-world stochastic flexible flow shop problem with limited buffer. En: International Journal of Production Economics, 2013, vol. 145, no 1, p. 88-95.

<sup>55</sup> UETAKE, Toshifumi; TSUBONE, Hitoshi; OHBA, Masaaki. Op. Cit. p. 395-398.

<sup>56</sup> LOW, Chinyao; HSU, Chou-Jung; SU, Chwen-Tzeng. A two-stage hybrid flowshop scheduling problem with a function constraint and unrelated alternative machines. En: Computers & Operations Research, 2008, vol. 35, no 3, p. 845-853.

<sup>57</sup> FIGIELSKA, Ewa. A new heuristic for scheduling the two-stage flowshop with additional resources. En: Computers & Industrial Engineering, 2008, vol. 54, no 4, p. 750-763.

Finalmente, la tercera variante con  $m$  máquinas paralelas no relacionadas en ambas etapas es estudiada por Rabiee et al.<sup>59</sup> que presentan las restricciones de *no-wait* y tiempos de alistamiento dependientes de la secuencia.

Continuando con las tres clases principales del *HFS* planteadas por Linn y Zhang,<sup>60</sup> aparece el *flow shop* híbrido de tres etapas de procesamiento en el que al menos un *stage* tiene  $m$  máquinas paralelas no relacionadas, esta clase del *Flow Shop* híbrido es estudiada por Li et al.<sup>61</sup> considerando restricciones de grupo en la segunda etapa y proponen 10 algoritmos para solucionar el problema. Anteriormente el problema fue trabajado por Yaurima et al.<sup>62</sup> en un caso en la industria de fabricación de televisores con circuito impreso a bordo con restricciones de tiempos de alistamiento dependientes de la secuencia, de disponibilidad y *buffer* de tamaño limitado.

El *flow shop* híbrido con múltiples etapas es la tercera clase principal del *HFS*; dicha clase ha sido trabajada por Nagano y Mocellin<sup>63</sup> quienes presentan el

---

<sup>58</sup> FIGIELSKA, Ewa. A genetic algorithm and a simulated annealing algorithm combined with column generation technique for solving the problem of scheduling in the hybrid flowshop with additional resources. En: Computers & Industrial Engineering, 2009, vol. 56, no 1, p. 142-151.

<sup>59</sup> RABIEE, M., et al. An intelligent hybrid meta-heuristic for solving a case of no-wait two-stage flexible flow shop scheduling problem with unrelated parallel machines. En: The International Journal of Advanced Manufacturing Technology, 2014, vol. 71, no 5-8, p. 1229-1245.

<sup>60</sup> LINN, Richard; ZHANG, Wei. Op. Cit. p.57-61.

<sup>61</sup> LI, Zhan-tao, et al. Approximation algorithms for the three-stage flexible flow shop problem with mid group constraint. En: Expert Systems with Applications, 2015, vol. 42, no 7, p. 3571-3584.

<sup>62</sup> YAURIMA, Victor; BURTSEVA, Larisa; TCHERNYKH, Andrei. Hybrid flowshop with unrelated machines, sequence dependent setup time and availability constraints: An enhanced crossover operator for a genetic algorithm. Parallel processing and applied mathematics. En: Springer Berlin Heidelberg, 2008. p. 608-617.

<sup>63</sup> NAGANO, Marcelo; MOCCELLIN, João. Flowshops paralelos com processadores não-relacionados. En: Processing, 2002, p. 1-8.

problema como dos *flow shop* tradicionales en paralelo y dos heurísticas para solucionar el problema.

Posteriormente es trabajado por Jungwattanakit et al.<sup>64,65,66</sup> en varios estudios donde consideran, en todos ellos, la restricción de tiempos de alistamiento no dependientes de la secuencia y proponen diferentes métodos de solución para cada uno los trabajos. Finalmente, otra investigación fue realizada por Low<sup>67</sup> proponiendo solucionar el problema mediante una heurística basada en el recocido simulado.

Al igual que el *HFS* con dos etapas, el *flow shop* híbrido con múltiples etapas puede dividirse en dos variantes: la primera se caracteriza porque los trabajos pueden omitir algunos *stages* de procesamiento y se le denomina *flow shop* híbrido flexible y ha sido trabajada por varios autores como Arango et al.<sup>68</sup>, Li et al.<sup>69</sup>, De Siqueira et al.,<sup>70</sup> Attar et al.,<sup>71</sup> Borzorgirad y Logendran,<sup>72</sup> Zandieh et al.<sup>73</sup> y Ruiz et al.<sup>74</sup>

---

<sup>64</sup> JUNGWATTANAKIT, Jitti, et al. Constructive and simulated annealing heuristics for hybrid flow shops with unrelated parallel machines. En: Proceedings of the 3rd OR-CRN Operations Research Conference, Bangkok, Thailand. 2006. p. 110-121.

<sup>65</sup> JUNGWATTANAKIT, Jitti, et al. Algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. En: The International Journal of Advanced Manufacturing Technology, 2008, vol. 37, no 3-4, p. 354-370.

<sup>66</sup> JUNGWATTANAKIT, Jitti, et al. A comparison of scheduling algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. En: Computers & Operations Research, 2009, vol. 36, no 2, p. 358-378.

<sup>67</sup> LOW, Chinyao. Simulated annealing heuristic for flow shop scheduling problems with unrelated parallel machines. En: Computers & Operations Research, 2005, vol. 32, no 8, p. 2013-2025.

<sup>68</sup> ARANGO-MARÍN, Jaime Antero; GIRALDO-GARCÍA, Jaime Alberto; CASTRILLÓN-GÓMEZ, Omar Danilo. Applying TOC Heuristics to Job Scheduling in a Hybrid Flexible Flow Shop. En: Dyna, 2014, vol. 81, no 186, p. 113-119.

<sup>69</sup> LI, Zhan-tao, et al. Scheduling rules for two-stage flexible flow shop scheduling problem subject to tail group constraint. En: International Journal of Production Economics, 2013, vol. 146, no 2, p. 667-678.

La segunda variante es denominada como *flow shop* híbrido cíclico o reentrante y es estudiada por autores como Soltani y Karimi<sup>75</sup> que proponen un algoritmo de recocido simulado y uno genético para la solución el problema con restricciones de elegibilidad de máquinas y *buffers* de tamaño limitado, y también por Kim y Lee<sup>76</sup> que evalúan el *makespan* y la tardanza total.

Varios investigadores abordan el problema con diversas restricciones, entre las más destacadas están los tiempos de alistamiento dependientes de la secuencia que ha sido trabajada por Defersha y Chen<sup>77</sup> que establecen que los tiempos de alistamiento dependientes de la secuencia pueden ser anticipatorios o no anticipatorios, Jenabi et al.<sup>78</sup> que la combinan con un problema de lote económico

---

<sup>70</sup> DE SIQUEIRA, Eduardo Camargo; DE SOUZA, Sergio Ricardo; SOUZA, Marcone Jamilson Freitas. Um algoritmo iterated greedy search aplicado à minimização do makespan no problema flowline híbrido e flexível, 2013, p.44-55.

<sup>71</sup> ATTAR, S. F.; MOHAMMADI, M.; TAVAKKOLI-MOGHADDAM, R. Op. Cit. P. 1583-1599.

<sup>72</sup> BOZORGIRAD, Mir Abbas; LOGENDRAN, Rasaratnam. Hybrid flowshop scheduling problem with a bi-criteria objective and group technology assumption. En: IIE Annual Conference. Proceedings. Institute of Industrial Engineers-Publisher, 2013. p. 3004.

<sup>73</sup> ZANDIEH, M., MOZAFFARI, E., GHOLAMI, M. Op. Cit. p. 731-743.

<sup>74</sup> RUIZ, Rubén; ŞERIFOĞLU, Funda Sivrikaya; URLINGS, Thijs. Modeling realistic hybrid flexible flowshop scheduling problems. En: Computers & Operations Research, 2008, vol. 35, no 4, p. 1151-1175.

<sup>75</sup> SOLTANI, S. Abolfazl; KARIMI, Behrooz. Op. Cit. p. 1-17.

<sup>76</sup> KIM, H. W.; LEE, D. H. Heuristic algorithms for re-entrant hybrid flow shop scheduling with unrelated parallel machines. Proceedings of the Institution of Mechanical Engineers, En: Part B: Journal of Engineering Manufacture, 2009, vol. 223, no 4, p. 433-442.

<sup>77</sup> DEFERSHA, Fantahun Melaku; CHEN, Mingyuan. Mathematical model and parallel genetic algorithm for hybrid flexible flowshop lot streaming problem. En: The International Journal of Advanced Manufacturing Technology, 2012, vol. 62, no 1-4, p. 249-265.

<sup>78</sup> JENABI, Masoud, et al. Two hybrid meta-heuristics for the finite horizon ELSP in flexible flow lines with unrelated parallel machines. En: Applied Mathematics and Computation, 2007, vol. 186, no 1, p. 230-245.

de programación, Parra y Mejía<sup>79</sup> que estudian la restricción en la industria ladrillera.

Otros autores como Ruiz y Maroto<sup>80</sup> estudian el problema con otras restricciones con un caso en la industria de fabricación de baldosas cerámicas, también, Nejati et al.<sup>81</sup> que trabajan la restricción junto con características de transferencia de lotes y sub-lotes, Bozorgirad y Logendran<sup>82</sup> que comparan dos metaheurísticas para solucionar el problema, Salazar y Sarzuri<sup>83</sup> que consideran los tiempos de alistamiento anticipatorios y proponen un algoritmo genético para mejorar la tardanza total; también Ebrahimi et al.<sup>84</sup> consideran los tiempos de alistamiento dependiente de la secuencia para familias de productos y proponen un algoritmo genético modificado para minimizar el *makespan* y la tardanza total y finalmente, Zabihzadeh y Rezaeian<sup>85</sup> que comparan dos metaheurística basadas en la colonia de hormigas y el algoritmo genético para minimizar el *makespan*.

---

<sup>79</sup> PARRA, Javier; MEJÍA, Gonzalo. Desarrollo de un modelo para la secuenciación de trabajos en la mediana industria ladrillera de la localidad XIX. 2006.

<sup>80</sup> RUIZ, Ruben; MAROTO, Concepción. A genetic algorithm for hybrid flow shops with sequence dependent setup times and machine eligibility. En: European Journal of Operational Research, 2006, vol. 169, no 3, p. 781-800.

<sup>81</sup> NEJATI, M. et al. Multi-job lot streaming to minimize the weighted completion time in a hybrid flow shop scheduling problem with work shift constraint. En: The International Journal of Advanced Manufacturing Technology, 2014, vol. 70, issue 1-4, p. 501-514.

<sup>82</sup> BOZORGIRAD, Mir Abbas; LOGENDRAN, Rasaratnam. A comparison of local search algorithms with population-based algorithms in hybrid flow shop scheduling problems with realistic characteristics. En: The International Journal of Advanced Manufacturing Technology, 2016, vol. 83, issue 5, p. 1135-1151.

<sup>83</sup> SALAZAR, Eduardo; SARZURI, René. Algoritmo genético mejorado para la minimización de la tardanza total en un flowshop flexible con tiempos de preparación dependientes de la secuencia. En: Ingeniare. Revista chilena de ingeniería, 2015, vol. 23, p. 118-127.

<sup>84</sup> EBRAHIMI, M. et al.. Hybrid flow shop scheduling with sequence dependent family setup time and uncertain due dates. En: Applied Mathematical Modelling, 2014, vol. 38, p. 2490-2504.

<sup>85</sup> ZABIHZADEH, S.S; REZAEIAN, J. Two meta-heuristic algorithms for flexible flow shop scheduling problem with robotic transportation and release time. En: Applied Soft Computing, 2016, vol. 40, p. 319-330.

Otra restricción destacada es la existencia de *buffers* o zonas intermedias de almacenamiento que pueden tener o no capacidad limitada. Dicha restricción es abordada en dos trabajos por Figielska,<sup>86, 87</sup> en el primero considera los *buffers* de capacidad limitada y los combina con reglas de preferencia en un *HFS* de dos etapas; en el segundo trabajo, también con *HFS* de dos etapas, considera de nuevo los *buffers* de capacidad limitada junto con otra restricción de requerimiento de recursos para el procesamiento de tareas. Yaurima<sup>88</sup> en 2009 trabaja también los *buffers* considerándolos de capacidad limitada. Xu et al.<sup>89</sup> consideran las zonas intermedias de almacenamiento con capacidad ilimitada y proponen como solución una modificación del algoritmo memético planteado por Eusiff y Lansey.

Otros autores como Miranda Lugo et al.<sup>90</sup> combinan la restricción en su variante de capacidad limitada con tiempos de alistamiento dependientes de la secuencia y elegibilidad de máquina. Li y Pan<sup>91</sup> consideran los *buffers* de tamaño limitado y proponen como solución un algoritmo híbrido que combina la colonia artificial de

---

<sup>86</sup> FIGIELSKA, Ewa. A heuristic for scheduling in a two-stage hybrid flowshop with renewable resources shared among the stages. En: European Journal of Operational Research, 2014, vol. 236, no 2, p. 433-444.

<sup>87</sup> FIGIELSKA, Ewa. Heuristic algorithms for preemptive scheduling in a two-stage hybrid flowshop with additional renewable resources at each stage. En: Computers & Industrial Engineering, 2010, vol. 59, no 4, p. 509-51.

<sup>88</sup> YAURIMA, Victor; BURTSEVA, Larisa; TCHERNYKH, Andrei. Hybrid flowshop with unrelated machines, sequence-dependent setup time, availability constraints and limited buffers. En: Computers & Industrial Engineering, 2009, vol. 56, no 4, p. 1452-1463.

<sup>89</sup> XU, Ye, et al. An effective shuffled frog leaping algorithm for solving hybrid flow-shop scheduling problem. En: Advanced Intelligent Computing. Springer Berlin Heidelberg, 2012. p. 560-567.

<sup>90</sup> MIRANDA LUGO, P.L. et al. Um modelo de programação inteira mista para a programação da produção em flowshop híbrido com buffers limitados. En: Simpósio Brasileiro de Pesquisa Operacional, 2013, p. 154-165.

<sup>91</sup> LI, Jun-qing; PAN, Quan-ke. Solving the large-scale hybrid flow shop scheduling problem with limited buffers by a hybrid artificial bee colony algorithm. En: Information Sciences, 2014, vol. 316, p. 487-502.

abejas con la búsqueda tabú. Más recientemente Wei et al.<sup>92</sup> en un *HFS* de dos etapas, consideran *buffers* de tamaño máximo dos, pues concluyen que un tamaño mayor de la zona de almacenamiento traería peores resultados para minimizar el *makespan*. Posteriormente Tran y Ng<sup>93</sup> trabajan ambas variantes, tamaño infinito o finito de los *buffers*, en un mismo estudio y proponen un algoritmo *water-flow* inspirado tanto en el ciclo hidrológico en meteorología como en el fenómeno de la erosión en la naturaleza con el objetivo de minimizar principalmente el tiempo de completamiento de los trabajos. Finalmente, Yu y Li<sup>94</sup> consideran junto a zonas de almacenamiento intermedias de capacidad limitada, fechas de vencimiento de trabajos en un *HFS* de dos *stages* y proponen un algoritmo híbrido de dos etapas para minimizar la tardanza.

El objetivo del problema del *HFS* es minimizar alguna medida de desempeño, que con base en lo expuesto por Castrillón et al.<sup>95</sup> siempre es una función del tiempo de terminación de los trabajos. La medida mayormente utilizada es la de minimizar el *makespan* presente en estudios de, Yaurima et al.,<sup>96</sup> Gómez-Gasquet et al.,<sup>97</sup>

---

<sup>92</sup> WEI, Q. et al. A FPTAS for a two-stage hybrid flow shop problem and optimal algorithms for identical Jobs. En: Theoretical Computer Science, 2014, vol. 524, p. 78-89.

<sup>93</sup> TRAN, T.H; NG, K.M. A water-flow algorithm for flexible flow shop scheduling with intermediate buffers. En: J Sched, 2011, vol. 14, p. 483-500.

<sup>94</sup> YU, Y; LI, T. Hybrid Algorithms for Hybrid Flow Shop Scheduling considering Limited Buffers and Due Dates. En: Advances in information Sciences and Service Sciences (AISS), 2013, vol. 5, num. 10, p. 861-868.

<sup>95</sup> CASTRILLÓN O.; GIRALDO J.; SARACHE W. Técnicas inteligentes y estocásticas en scheduling. En: Un enfoque en la Prod. y las operaciones, 2010, Unibiblios.

<sup>96</sup> YAURIMA, Victor; BURTSEVA, Larisa; TCHERNYKH, Andrei. Hybrid flowshop with unrelated machines, sequence dependent setup time and availability constraints: An enhanced crossover operator for a genetic algorithm. Parallel processing and applied mathematics. En: Springer Berlin Heidelberg, 2008. p. 608-617.

<sup>97</sup> GÓMEZ-GASQUET, Pedro; ANDRÉS, Carlos; LARIO, Francisco-Cruz. An agent-based genetic algorithm for hybrid flowshops with sequence dependent setup times to minimize makespan. En: Expert Systems with Applications, 2012, vol. 39, no 9, p. 8095-8107.

Ruiz y Maroto,<sup>98</sup> López et al.,<sup>99</sup> Cui y Gu,<sup>100</sup> Fattahi et al.,<sup>101</sup> Tang et al.,<sup>102</sup> Wang y Choi<sup>103</sup> y Marichelvam et al.<sup>104</sup> Algunas otras medidas de desempeño utilizadas de acuerdo a Castrillón et al.,<sup>105</sup> son máxima tardanza, tardanza total, número de trabajos tardíos, entre otras.

Existen varios métodos de solución para el problema del *flow shop* híbrido, los más utilizados son los algoritmos genéticos que han sido trabajados por autores como Figielska<sup>106</sup> en sus ya mencionados trabajos para la segunda variante del *HFS* de dos etapas, Gómez-Gasquet et al.,<sup>107</sup> que utilizan un algoritmo genético

---

<sup>98</sup> RUIZ, Ruben; MAROTO, Concepción. Op. Cit. p.781-800.

<sup>99</sup> LÓPEZ J.C et al.. Reducción del Tiempo de Terminación en la Programación de la Producción de una Línea de Flujo Híbrida Flexible (HFS). En: Información Tecnológica, 2015, vol. 26(3), p. 157-172.

<sup>100</sup> CUI, Z; GU, X. An improved discrete artificial bee colony algorithm to minimize the makespan on hybrid flow shop problems. En: Neurocomputing, 2015, vol. 148, p. 248-259.

<sup>101</sup> FATTAHI, P., HOSSEINI, S. M. H., JOLAI, F., TAVAKKOLI-MOGHADDAN, R. Op. Cit. p.119-139.

<sup>102</sup> TANG, D. et al.. Energy-efficient dynamic scheduling for a flexible flow shop using an improved particle swarm optimization, En: Computers in Industry, 2015.

<sup>103</sup> WANG, K; CHOI, S.H. A holonic approach to flexible flow shop scheduling under stochastic processing times. En: Computers & Operations Research, 2014, vol. 43, p. 157-168.

<sup>104</sup> MARICHELVAM, M.K. et al.. Improved cuckoo search algorithm for hybrid flow shop scheduling problems to minimize makespan. En: Applied Soft Computing, 2014, vol. 19, p. 93-101.

<sup>105</sup> CASTRILLÓN O.; GIRALDO J.; SARACHE W. Op. Cit.

<sup>106</sup> FIGIELSKA, Ewa. A genetic algorithm and a simulated annealing algorithm combined with column generation technique for solving the problem of scheduling in the hybrid flowshop with additional resources. En: Computers & Industrial Engineering, 2009, vol. 56, no 1, p. 142-151.

<sup>107</sup> GÓMEZ-GASQUET, Pedro; ANDRÉS, Carlos; LARIO, Francisco-Cruz. An agent-based genetic algorithm for hybrid flowshops with sequence dependent setup times to minimize makespan. En: Expert Systems with Applications, 2012, vol. 39, no 9, p. 8095-8107.

de programación multi-agente *MAGSA*, Zandieh et al.,<sup>108</sup> que modifican los valores de los parámetros del algoritmo genético en busca de mejores respuestas.

También trabajando algoritmos genéticos están Jun y Park<sup>109</sup> que lo presentan como solución combinada con las reglas *NEH* y *EDD* para un caso en la industria de fabricación de transformadores, Lugo y Teixeira<sup>110</sup> que lo plantean modificando el operador de mutación por un operador de búsqueda local y Costa et al.<sup>111</sup> que proponen un algoritmo genético para la variante del *HFS* donde algunas máquinas pueden procesar grupos de trabajos de forma simultánea.

Finalmente, el Algoritmo Competitivo Imperialista (*ICA*) también ha sido utilizado últimamente para solucionar el problema del *HFS* por varios autores como Naderi y Yazdani<sup>112</sup> que lo proponen como un algoritmo efectivo para un modelo con restricciones de sub-lotes y tiempos de alistamiento dependientes de la secuencia, Rabiee et al.<sup>113</sup> combinan el *ICA* con recocido simulado para su ya mencionado

---

<sup>108</sup> ZANDIEH, M.; MOZAFFARI, E.; GHOLAMI, Mansour. Op. Cit. p. 731-743.

<sup>109</sup> JUN, Sungbum; PARK, Jinwoo. A hybrid genetic algorithm for the hybrid flow shop scheduling problem with nighttime work and simultaneous work constraints: A case study from the transformer industry. En: Expert Systems with Applications, 2015, vol. 42, no 15, p. 6196-6204.

<sup>110</sup> MIRANDA LUGO, Pedro, et al. Algoritmo genético com busca local para a programação da produção em sistemas flow-shop híbridos. En: Congreso Latino-Iberoamericano de Investigación Operativa (44:24-28, septiembre, Rio de Janeiro, Brasil). Publicaciones. Universidad Federal de Sao Carlos. 2012, p. 19-29.

<sup>111</sup> COSTA, Antonio; CAPPADONNA, Fulvio Antonio; FICHERA, Sergio. A novel genetic algorithm for the hybrid flow shop scheduling with parallel batching and eligibility constraints. En: The International Journal of Advanced Manufacturing Technology, 2014, vol. 75, no 5-8, p. 833-847.

<sup>112</sup> NADERI, B.; YAZDANI, M. A model and imperialist competitive algorithm for hybrid flow shops with sublots and setup times. En: Journal of Manufacturing Systems, 2014, vol. 33, p. 647-653.

<sup>113</sup> RABIEE, M., et al.. Op. Cit. p.1229-1245

estudio en la segunda variante del *HFS* de dos *stages* y Khalili y Naderi<sup>114</sup> que destacan su rendimiento sobre otros algoritmos para un *flow shop* híbrido con tiempos de alistamiento dependientes de la secuencia y restricción de *no-wait* con el objetivo de minimizar el *makespan* y la tardanza total.

---

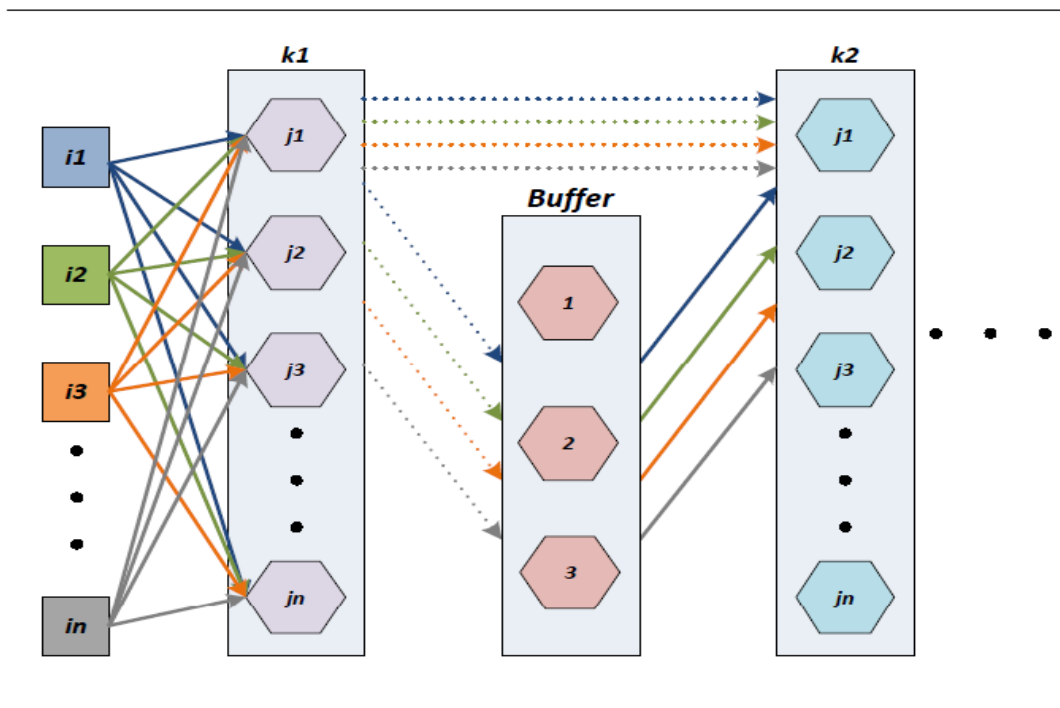
<sup>114</sup> KHALILI, M.; NADERI, B. A bi-objective imperialist competitive algorithm for no-wait flexible flow lines with sequence dependent setup times. En: Int J Adv Manuf Technol, 2015, vol. 76, p. 461-469.

## 4. DESARROLLO DEL MODELO MATEMÁTICO

### 4.1 DESCRIPCIÓN DEL SISTEMA PRODUCTIVO A ESTUDIARSE

Se estudiará el sistema productivo conocido como *flow shop* híbrido con máquinas paralelas no relacionadas y *buffers* de tamaño limitado, el cual es utilizado para la elaboración de bienes que deben seguir un flujo unidireccional a través del sistema, es decir, todos los trabajos siguen el mismo flujo por las diferentes estaciones de procesamiento y por lo tanto deben ser procesados en al menos una de las  $m$  máquinas dentro de cada etapa, para todas las etapas o centros de trabajo, para ilustrar el sistema descrito se presenta la figura 5 de la siguiente página.

**Figura 5.** Figura del sistema productivo *HFS* con *buffers* a estudiarse



El *HFS* con *buffers* limitados cuenta con la característica que en las diferentes estaciones de trabajo existen como mínimo dos máquinas en paralelo no relacionadas, además, las características de dichas máquinas son distintas y

aunque todas pueden procesar todos los trabajos, los tiempos de completamiento son diferentes entre sí. Como consecuencia, el tiempo que demora un trabajo siendo procesado en un centro de trabajo se considera dependiente de la máquina.

Otra característica importante es que los tiempos de alistamiento entre trabajos se consideran dependientes de la secuencia de los mismos, entonces, el tiempo de preparación de un trabajo va a depender de su predecesor y de la máquina a la cual sea asignado.

Con relación a los buffers de tamaño limitado, se consideran zonas de almacenamiento temporal de los trabajos en proceso entre las diferentes estaciones, como ya se dijo, dicha zona cuenta con una capacidad limitada de albergar trabajos con respecto a la cantidad y su uso se hace necesario cuando un trabajo acaba su procesamiento en un centro de trabajo y el siguiente centro se encuentra ocupado o no puede procesar el trabajo aún.

Finalmente, los tiempos de alistamiento y de procesamiento para los trabajos son conocidos. Los tiempos de alistamiento siguen una distribución uniforme entre 25 y 75 y los tiempos de procesamiento se acomodan también a una distribución uniforme entre 75 y 125 de acuerdo con lo expuesto por Miranda Lugo et al.<sup>115</sup>

## **4.2 ESQUEMA MATEMÁTICO DEL MODELO**

El modelo de programación lineal entera mixta está basado en el presentado por Miranda Lugo et al.<sup>116</sup> en 2013, el cual se modificó para el problema del *flow shop* híbrido con máquinas paralelas no relacionadas, tiempos de alistamiento

---

<sup>115</sup> MIRANDA LUGO, P.L. et al. Um modelo de programação inteira mista para a programação da produção em flowshop híbrido com buffers limitados. En: Simpósio Brasileiro de Pesquisa Operacional, 2013, p. 154-165.

<sup>116</sup> Ibid.

dependientes de la secuencia y *buffers* de tamaño limitado con el objetivo de minimizar el *makespan*.

La elección de este modelo de programación lineal se hace con base a que presenta las características básicas necesarias para modelar el *hybrid flow shop*, su comprensión no es complicada y además al estar basado en una modificación al *HFS* parecida a la planteada en esta investigación se pudo realizar una buena adaptación del modelo a las necesidades del problema como estancias en el *buffer* y bloqueo de máquinas.

El problema cuenta con los siguientes supuestos básicos para su planteamiento:

- Cada estación consta de, al menos, dos máquinas paralelas no relacionadas.
- En el instante  $t=0$ , todas las máquinas de todas las estaciones están disponibles.
- Todas las máquinas, de todas las estaciones, pueden procesar todos los trabajos.
- Cada trabajo debe ser asignado a solo una máquina en cada una de las estaciones de trabajo.
- Cada máquina debe procesar solamente un trabajo a la vez.
- No existen límites de asignación de trabajos para cada máquina.
- En cada una de las máquinas, se considera la existencia de un trabajo imaginario cero; este trabajo imaginario cero se toma como un calentamiento de la máquina al inicio de la secuencia y tendrá efecto en la asignación de tareas y los tiempos de alistamiento junto al primer trabajo asignado.
- Los tiempos de alistamiento son dependientes de la secuencia.
- Los *buffers* son considerados como máquinas “especiales” dentro de una estación de almacenamiento intermedio entre etapas, cuentan con tiempo de procesamiento cero y tiempo de alistamiento cero sea cual sea la secuencia.
- El manejo de los *buffers* en el modelo de programación lineal no se hace como un parámetro de entrada, debido a que por la naturaleza del modelo eso no

puede ser posible, por lo tanto, el tamaño de *buffer* se maneja mediante una penalización en la matriz de tiempos de alistamiento y de procesamiento. Como los *buffers* se consideran máquinas con características “especiales”, en las estaciones donde el tamaño de la zona de almacenamiento es menor que el número de máquinas, para los *buffers* que no se deben utilizar se les pone tiempos muy superiores al cero, para que así *GAMS* descarte esas posiciones ya que el objetivo del modelo es minimizar el *makespan*. Para más detalle, en la instancia presentada como ejemplo en el anexo A se puede ver este manejo.

- Finalmente, cuando un trabajo termine de ser procesado en una estación, debe ser asignado a alguna de las máquinas disponibles en la próxima estación; en dado caso que no existan procesadores disponibles en la próxima etapa debe pasar a una zona de almacenamiento intermedia o *buffer* cuyo tamaño es limitado, si tanto el *buffer* como las máquinas de la siguiente estación están ocupados el trabajo se quedará en el procesador de la estación actual bloqueándolo y evitando que pueda procesar otros trabajos.

#### **4.2.1 Modelo de programación lineal entera mixta del problema**

Se presentará a continuación el detalle de cada uno de los índices y conjuntos, parámetros y variables de decisión del modelo junto a su respectiva función objetivo y restricciones.

Índices y conjuntos:

- $h, i, g$  = Tareas.
- $j$  = Máquinas.
- $k$  = Estaciones.

Parámetros:

- $B$  = Número escalar muy grande, mayor que la suma de todos los tiempos de procesamiento en todos los trabajos para todas las máquinas.

- $S_{hijk}$  = Tiempo de alistamiento de la tarea  $h$  siendo precedida por la tarea  $i$ , en la máquina  $j$ , de la estación  $k$ .
- $P_{ijk}$  = Tiempo de procesamiento de la tarea  $i$  en la máquina  $j$ , de la estación  $k$ .

Variables de decisión:

- $Y_{hijk} = 1$ , si la tarea  $h$  precede a la tarea  $i$  en la máquina  $j$  de la estación  $k$ .  
0 en el caso contrario.
- $C_{ik}$  = Tiempo de finalización de la tarea  $i$  en la estación  $k$ .
- $E_{ik}$  = Tiempo de salida de la tarea  $i$  en la estación  $k$ .
- $C_{m\acute{a}x}$  = *Makespan*. Momento en que termina el procesamiento del último trabajo en la última estación.

Función objetivo:

$$\text{Min. } C_{m\acute{a}x} \quad (9)$$

Restricciones:

$$\sum_j \sum_h Y_{hijk} = 1 \quad \forall i, k; h \neq i; i \neq 0; \quad (10)$$

$$\sum_j \sum_h Y_{ihjk} \leq 1 \quad \forall i, k; h \neq i; h \neq 0; i \neq 0; \quad (11)$$

$$\sum_g Y_{ghjk} \geq Y_{hijk} \quad \forall h, i, j, k; h \neq i; g \neq h; g \neq i; h \neq 0; i \neq 0; \quad (12)$$

$$\sum_j (Y_{hijk} + Y_{ihjk}) \leq 1 \quad \forall h, i, k; h \neq i; i \neq 0; \quad (13)$$

$$\sum_i Y_{hijk} \leq 1 \quad \forall h, j, k; h \neq i; i \neq 0; h = 0; \quad (14)$$

$$C_{ik} = 0 \quad \forall k; i = 0; \quad (15)$$

$$C_{ik} + B(1 - Y_{hijk}) \geq E_{hk} + S_{hijk} + P_{ijk} \quad \forall h, i, j, k; \quad h \neq i; \quad i \neq 0; \quad (16)$$

$$C_{ik} + B(1 - Y_{hijk}) \geq C_{i,k-1} + S_{hijk} + P_{ijk} \quad \forall h, i, j, k; \quad h \neq i; \quad i \neq 0; \quad k \neq 1; \quad (17)$$

$$E_{i,k-1} = C_{ik} - \sum_j \sum_h Y_{hijk} \cdot S_{hijk} - \sum_j \sum_h Y_{hijk} \cdot P_{ijk} \quad \forall i, k; \quad h \neq i; \quad i \neq 0; \quad k \neq 1; \quad (18)$$

$$E_{ik} \geq C_{ik} \quad \forall i, k; \quad k \neq \text{última estación} \quad (19)$$

$$E_{ik} = C_{ik} \quad \forall i, k; \quad k = \text{última estación} \quad (20)$$

$$C_{m\acute{a}x} \geq C_{ik} \quad \forall i, k; \quad k = \text{última estación} \quad (21)$$

$$Y_{hijk} \in \{0,1\} \quad \forall h, i, j, k; \quad h \neq i; \quad i \neq 0; \quad (22)$$

$$C_{ik}, E_{ik} \geq 0 \quad \forall i, k \quad (23)$$

La ecuación (9), es la encargada de definir la función objetivo del problema, para este caso corresponde a la minimización del *makespan*. La ecuación (10) asegura que cada tarea debe ser precedida por únicamente otra tarea, en solamente una máquina de cada estación, además, garantiza la no división de trabajos en lotes de procesamiento al cerciorarse que las tareas sean asignadas a una sola máquina en cualquier estación de trabajo. Por su parte, la ecuación (11), indica que cada trabajo, en caso de ser asignado a cualquier máquina de cualquier estación, debe tener a lo sumo, un sucesor. En cuanto a la ecuación (12), su función es cerciorarse que una tarea procesada en una máquina asignada debe tener como máximo un predecesor dentro de la misma máquina. La ecuación (13) tiene por trabajo evitar la existencia de precedencias cruzadas, en otras palabras, se busca que si al trabajo *i*, lo precede el trabajo *h*, entonces el trabajo *h* no preceda al *i*. La ecuación (14) asegura que el trabajo ficticio inicial cero debe tener como máximo un solo sucesor en cualquier máquina de cualquier estación.

En cuanto a la ecuación (15) se encarga de asegurar que el tiempo de completamiento del trabajo ficticio inicial sea cero. Por su parte, las ecuaciones

(16) y (17) son las encargadas de calcular el tiempo de finalización de cada trabajo. Para calcular el tiempo de finalización de una tarea dentro de una estación asignada pueden existir dos escenarios, el primero donde el tiempo de completamiento de la tarea  $i$  dependa del tiempo de salida de la tarea inmediatamente anterior en la secuencia dentro de la misma estación  $k$  y el segundo donde el tiempo de completamiento de  $i$  depende del tiempo de completamiento de la misma tarea  $i$ , pero en la estación inmediatamente anterior, o sea, de la estación  $k-1$ .

En el primer escenario actúa la ecuación número (16), la cual calcula la terminación de la tarea para todas las estaciones basándose en el tiempo de salida de la tarea anterior de la secuencia en la misma estación, y asegura que el tiempo de completamiento de la tarea  $i$  en la estación  $k$  deba ser mayor o igual al tiempo de salida de la tarea  $h$  (predecesora de  $i$ ) de la estación  $k$  sumado al tiempo de alistamiento y de procesamiento de la tarea  $i$  en la máquina  $j$ , de la estación  $k$ , así mismo, verifica que ninguna tarea sea procesada en una máquina asignada sin que la tarea anterior haya salido de la misma.

Para el segundo escenario actúa la ecuación número (17), la cual también calcula el tiempo de finalización para todas las estaciones, exceptuando la primera, y toma como base para el cálculo el tiempo de completamiento de la misma tarea en la estación inmediatamente anterior, es por dicha razón que no actúa en la primera etapa, pues al ser la primera, no existen estaciones anteriores. La ecuación asegura que el tiempo de completamiento de la tarea  $i$  en la estación  $k$  deba ser mayor o igual al tiempo de completamiento de la misma tarea  $i$  en la estación inmediatamente anterior (estación  $k-1$ ) sumado al tiempo de alistamiento y de procesamiento de la tarea  $i$  en la máquina  $j$ , de la estación  $k$ , además, tiene el encargo de asegurar que cada trabajo sea procesado de manera subsecuente en todas las etapas.

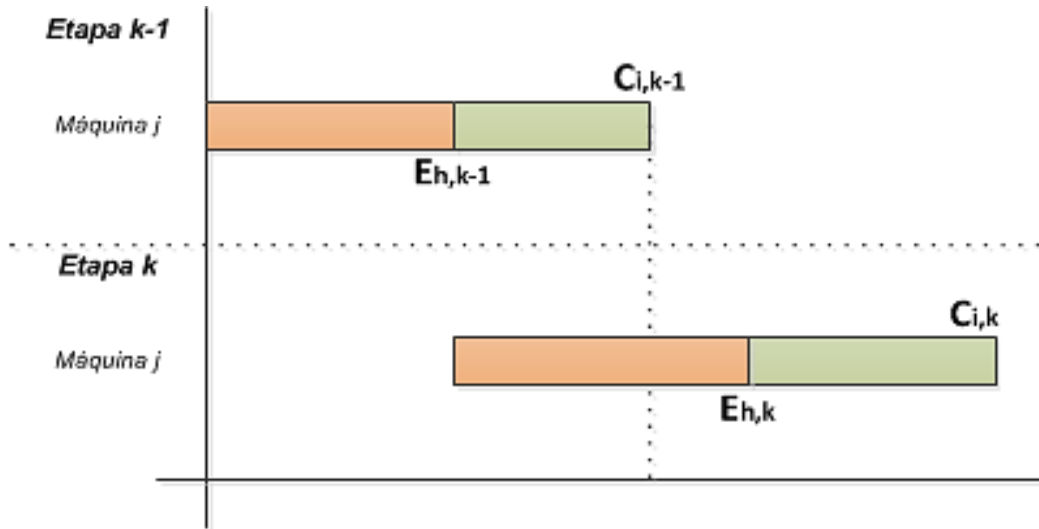
Se puede presentar la duda del porqué para el segundo escenario, el tiempo de finalización de la tarea  $i$  en la estación  $k$  depende del tiempo del completamiento de la misma tarea  $i$  en la estación anterior  $k-1$  y no de su mismo tiempo de salida en la estación  $k-1$ . El único caso donde el tiempo de completamiento de una tarea dada en una estación asignada es diferente a su tiempo de salida en la misma estación asignada es cuando existe un bloqueo o una estancia en el *buffer* y para que se presenten estas dos situaciones se debe dar el hecho que cuando la tarea  $i$  terminó su procesamiento en la etapa  $k-1$ , la tarea  $h$ , o la predecesora de  $i$  en la secuencia, no ha terminado su procesamiento en la siguiente estación  $k$ , y así, el tiempo de completamiento y a su vez de salida de la tarea  $h$  en la etapa  $k$  ( $E_{h,k}$ ) es mayor que el tiempo de completamiento de  $i$  en la estación anterior ( $C_{i,k-1}$ ) y para ese caso entraría a actuar la ecuación (16) y se haría redundante plantear la ecuación (17) con base al tiempo de salida pues terminarían actuando ambas ecuaciones para una misma situación. Con el fin de reflejar de mejor manera el comportamiento de las ecuaciones (16) y (17) y la explicación inmediatamente anterior se presentan las figuras 6, 7 y 8 donde se explican cada una respectivamente.

La ecuación (18) es la encargada de realizar el cálculo del tiempo de salida de cada tarea en cada etapa, el cálculo se hace con base en el tiempo de finalización de la misma tarea en la siguiente estación, y básicamente dice que el tiempo de salida de una tarea  $i$  de estación  $k-1$  (estación anterior), tiene que ser igual al tiempo de completamiento de la misma tarea  $i$ , pero de la estación  $k$  (la actual) menos tiempo de alistamiento y de procesamiento de la tarea  $i$  en la máquina  $j$ , de la estación actual  $k$ .

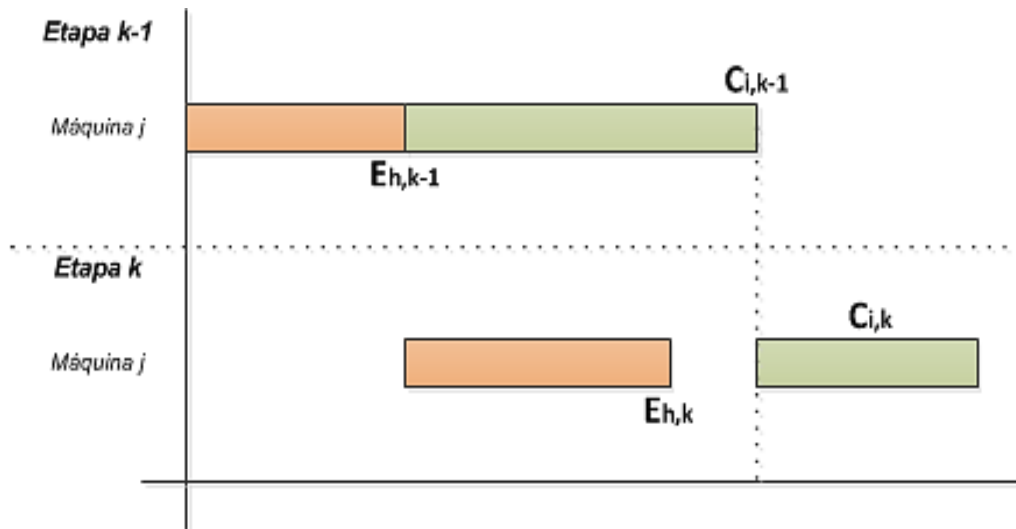
El trabajo de la ecuación (19) es garantizar que ningún trabajo deje una estación antes que su procesamiento en la misma esté finalizado. Por su parte, la ecuación (20) indica que cada trabajo sale del sistema cuando su procesamiento en la última estación esté finalizado. El *makespan* es definido mediante la ecuación

(21). Por último, las ecuaciones (22) y (23) sencillamente definen las variables de decisión del problema.

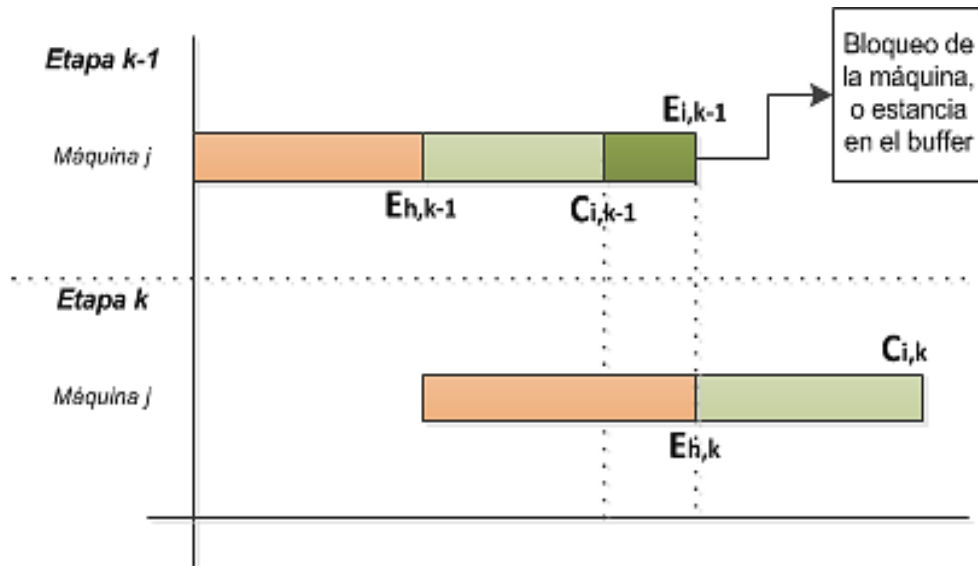
**Figura 6.** Determinación del tiempo de completamiento de la tarea  $i$ , en la estación  $k$  ( $C_{i,k}$ ) para el caso donde  $E_{h,k} \geq C_{i,k-1}$ .



**Figura 7.** Determinación del tiempo de completamiento de la tarea  $i$ , en la estación  $k$  ( $C_{i,k}$ ) para el caso donde  $C_{i,k-1} \geq E_{h,k}$ .



**Figura 8.** Explicación de la dependencia del tiempo de completamiento de la tarea  $i$ , en la estación  $k$  ( $C_{i,k}$ ) de  $C_{i,k-1}$  y no de  $E_{i,k-1}$ .



Para un detalle del modelo estudiado programado en lenguaje *GAMS* se presenta en el anexo A el pseudocódigo para la instancia de siete trabajos, tres etapas de procesamiento y dos estaciones con zona de almacenamiento de trabajos, dos máquinas paralelas no relacionadas en cada etapa y tamaño del *buffer* uno.

#### 4.3 ANÁLISIS DE LOS RESULTADOS OBTENIDOS MEDIANTE EL MODELO DE PROGRAMACIÓN LINEAL ENTERA MIXTA DEL PROBLEMA

Se prosiguió a plantear el modelo de programación lineal entera mixta del problema en lenguaje *GAMS*, consecuentemente se ejecutaron cada una de las 20 instancias planteadas para el problema utilizando el *solver* *CPLEX 12.4.0.1* del programa *GAMS 23.9.5* en una computadora con procesador Intel Core i5-4570, 3.2 GHz con una memoria *RAM* de 8 *Gigabytes* y un sistema operativo de 64 *bits*. En el anexo B se encuentra el detalle de cada una de las 20 instancias utilizadas.

Se estableció un tiempo límite de ejecución del modelo mediante el *software* *GAMS* de 10800 segundos, pues debido a la complejidad del problema ya en estudios anteriores se había planteado un tiempo límite de resolución; para definir

el tiempo utilizado en este trabajo se tomó como base el utilizado por Miranda Lugo et al.<sup>117</sup> los cuales fijaron el tiempo límite en 3600 segundos, se decidió ampliarlo para dar la oportunidad al *solver* de encontrar respuestas de mejor calidad y así tener una mejor comparación frente a las respuestas arrojadas por la metaheurística.

**Tabla 1.** Gap relativo para instancias con siete trabajos y nueve trabajos.

Trabajos ( <i>i</i> )			7				
Máquinas por estación ( <i>j</i> )			2		3		
Tamaño del buffer			1	2	1	2	3
Etapas ( <i>k</i> )	3 (5)	GAP (%)	37,3422%	9,9998%	33,3082%	12,4158%	10,5887%
	5 (9)		43,9846%	30,1702%	26,0406%	23,4133%	21,3772%
Trabajos ( <i>i</i> )			9				
Máquinas por estación ( <i>j</i> )			2		3		
Máquinas en el buffer			1	2	1	2	3
Etapas ( <i>k</i> )	3 (5)	GAP (%)	55,2210%	46,3289%	50,4056%	36,5287%	48,4871%
	5 (9)			46,6291%		34,4474%	37,0131%

En la tabla 1 se presentan los resultados obtenidos a través del modelo de programación lineal entera mixta del problema; para cada una de las instancias se muestra la calidad de la solución obtenida dentro del tiempo límite de ejecución mediante el concepto de % *gap* relativo, que se entiende como la diferencia en porcentaje entre la solución arrojada por el *solver* y la mejor solución posible del problema. Como se puede observar, de las instancias planteadas para el problema ninguna alcanzó la mejor solución posible dentro del tiempo límite.

Es importante destacar desde el comienzo, que en 2 instancias de las 20 utilizadas (un 10% del total y ambas con tamaño de *buffer* uno), el *solver* no logró converger en una solución acorde al contexto del problema dentro de los límites de tiempo establecidos. Esto se puede deber a una restricción del modelo de

<sup>117</sup> MIRANDA LUGO, P.L. et al. Um modelo de programação inteira mista para a programação da produção em flowshop híbrido com buffers limitados. *En: Simpósio Brasileiro de Pesquisa Operacional*, 2013, p. 154-165.

programación lineal entera mixta que exige mucha capacidad de computación y no permite encontrar una mejor respuesta; se presenta este argumento pues éstas instancias se corrieron durante un tiempo de resolución superior al de las demás (18000 segundos) y además, se les aumentó el tiempo de penalización para *buffers* que no deben ser utilizados y al momento de detenerse el *solver* se obtenían respuestas con *gaps* relativos superiores al 95%.

Para intentar solucionar el problema se modificaron las ecuaciones (16), (17) y (18) las cuales calculan tiempos de completamiento y salida incluyendo la variable  $Y_{hijk}$  en la parte derecha de las mismas, esto con el fin de evitar exceso de cálculos en secuencias de tareas que al final no se van a tener en cuenta para la solución final (como la variable binaria da pares ordenados de trabajos cuando se hace cero, los mismos, en ese orden no se van a tener en cuenta para la secuencia), pero para los casos probados, el modelo arrojaba error, también se probó disminuyendo la penalización a los *buffers* no utilizados, pero para este caso *GAMS* también arrojaba error.

Se mantuvo el presente modelo de programación lineal porque para el resto de instancias da soluciones para el caso del *HFS* con *buffers* limitados y porque para otras investigaciones como las de Moratto y Pérez<sup>118</sup> que utilizan como base el mismo modelo de programación lineal entera mixta y tiempos con la misma distribución de probabilidad, instancias como estas tomaban cerca de siete días en dar una solución acorde al contexto del problema, por lo tanto, no es atípico este tipo de situaciones para el modelo planteado en instancias grandes; a su vez, esto demuestra la complejidad computacional del problema y como la misma se ve potenciada cuando el tamaño del *buffer* es muy pequeño con relación al número de trabajos a asignarse.

---

<sup>118</sup> MORATTO CHIMENTY, ESELENDIS; PÉREZ FIGUEREDO, LUIS EDUARDO. Metaheurística basada en el algoritmo competitivo imperialista (*ICA*) aplicada a la solución del problema del *flow shop híbrido (HFS)* con máquinas paralelas no relacionadas. Trabajo de grado. Ingeniería Industrial. Universidad Industrial de Santander. 2016.

De las instancias planteadas, solo la respuesta perteneciente a la instancia i7j2k3(5)-2 (7 trabajos, 2 máquinas por estación, 5 etapas de las cuales 3 son etapas de procesamiento y tamaño del *buffer* 2) tuvo un *gap* relativo inferior al 10%. Del total de 20 instancias planteadas el 25%, equivalente a 4 instancias, presentan un *gap* entre el 10% y el 25%, todas éstas 4 instancias se encuentran dentro del grupo de siete trabajos y ninguna tiene tamaño de *buffer* 1; dos pertenecen al grupo de tamaño de *buffer* 2 y las dos restantes *buffer* tamaño 3.

Entre el 25% y el 50% de *gap* relativo están el 55% de las instancias, o específicamente, 11 variantes del problema, de las cuales 5 pertenecen al grupo con siete tareas (cuatro de las mencionadas instancias son de tamaño del *buffer* 1 y una de tamaño de *buffer* 2), 6 pertenecen al grupo de nueve tareas (cuatro con tamaño de *buffer* 2 y dos con tamaño de *buffer* 3).

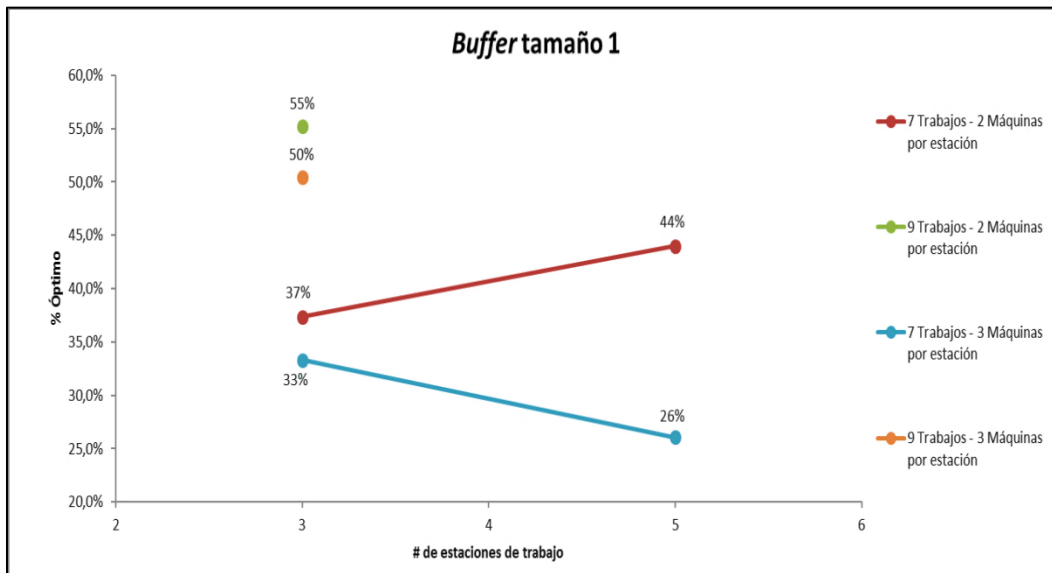
Para el grupo entre 50% y 75% de *gap* relativo están en total un 10% de las instancias, que representa a 2 variaciones del problema, las cuales pertenecen al grupo de nueve trabajos (ambas instancias con tamaño de *buffer* 1). Es importante destacar que ninguna instancia supera el 75% de *gap* relativo.

Como se puede observar en la tabla 1 y mediante el análisis presentado, a medida que aumenta el número de tareas a procesar y el número de centros de trabajos la complejidad del problema incrementa, ahora bien, el *gap* se ve especialmente afectado cuando el tamaño del *buffer* es pequeño (sobre todo cuando es tamaño uno) y, por lo tanto, la calidad de la solución obtenida para estos casos es de menor calidad aún. En consecuencia, a medida que aumenta la complejidad de los diferentes factores mencionados para el problema, se hace mucho más grande la dificultad para hallar una respuesta óptima en un tiempo razonable.

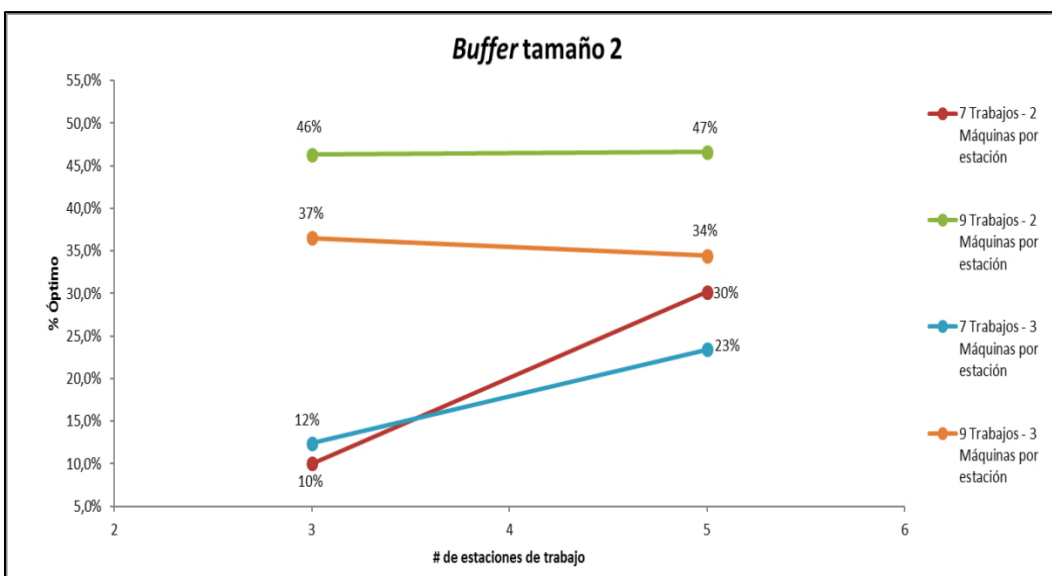
Mediante las gráficas 5, 6 y 7 se presentarán los porcentajes de *gap* relativo para cada una de las instancias planteadas ordenadas por tamaño de *buffer*, donde las respuestas con mayor porcentaje de *gap* relativo serán soluciones con menor calidad. En las diferentes gráficas se evidencia el impacto negativo en la calidad

de la respuesta al aumentar el número de tareas y a su vez, de un tamaño de *buffer* pequeño respecto a la cantidad de tareas a procesar. Se debe tener en cuenta para la gráfica 5 la ausencia de los dos datos para el caso de 9 trabajos a procesar con tamaño de *buffer* 1 por lo ya explicado al comienzo de éste análisis.

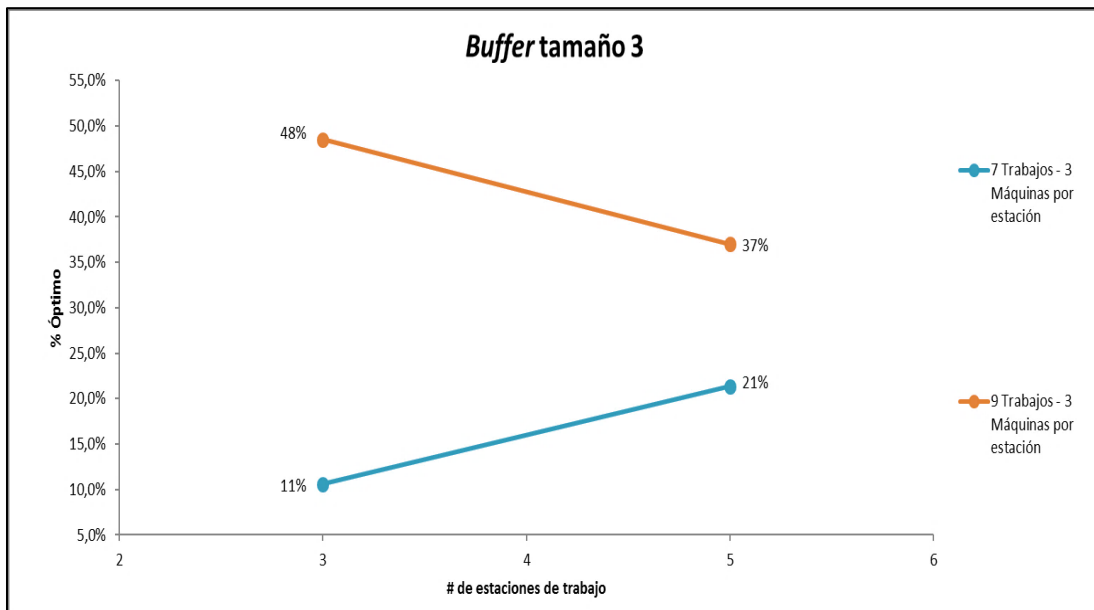
**Gráfica 5.** Gráfica de porcentaje de *gap* relativo para las soluciones obtenidas por GAMS en instancias con tamaño de *buffer* 1.



**Gráfica 6.** Gráfica de porcentaje de *gap* relativo para las soluciones obtenidas por GAMS en instancias con tamaño de *buffer* 2.



**Gráfica 7.** Gráfica de porcentaje de *gap* relativo para las soluciones obtenidas por GAMS en instancias con tamaño de *buffer* 3.



## 5. ESQUEMA DE LA METAHEURÍSTICA

Se selecciona para esta investigación una metaheurística cuyo principal componente es un algoritmo híbrido basado en el algoritmo competitivo imperialista (planteado originalmente por Atashpaz-Gargari y Lucas<sup>119</sup>) modificado por Rabiee et al.<sup>120</sup> los cuales incluyen características de algoritmos genéticos tales como selección, cruce y mutación. Además, se realizaron modificaciones en dos aspectos para adaptar la metaheurística y el algoritmo al *HFS* con *buffers* de tamaño limitado.

La primera modificación del algoritmo consiste en ya no incluir el tamaño de *buffers* mediante la matriz de tiempos y alistamientos como en los parámetros del modelo de programación lineal sino mediante un parámetro de entrada en la interfaz de la metaheurística, esto se hace con el fin de reducir el tamaño de las mencionadas matrices y no sobrecargar la memoria del computador de datos.

El segundo cambio del algoritmo se realiza en el algoritmo original sobre la generación de la población inicial, con el fin de mejorar la calidad de las respuestas, se modifica la generación al azar de los países iniciales. En la primera parte de la sección 5.1.1 se presentan los cambios de forma más específica.

Además, como estrategia para mejorar los resultados, se incluye una penalización en el *makespan* para aquellas secuencias a las cuales por bloqueos en exceso no se les pueda calcular el tiempo de completamiento de la última tarea. Se implementa esta estrategia con el fin de no eliminar ni modificar arbitrariamente respuestas de la metaheurística más allá de la población inicial, pues estas mismas respuestas durante el movimiento hacia el imperialista o durante los operadores genéticos que se detallarán más adelante pueden dar buenas

---

<sup>119</sup> ATASHPAZ, Esmaeil; LUCAS, Caro. Op. cit. p. 4661-4667.

<sup>120</sup> RABIEE, M., et al.. Op. Cit. p.1229-1245.

soluciones. De la misma manera se penalizan para que el algoritmo pueda converger de forma más rápida hacia una respuesta de buena calidad.

La elección del algoritmo híbrido basado en el *ICA* como base de la metaheurística que busca dar solución al problema se realiza por tres razones principalmente. La primera porque es un algoritmo relativamente nuevo que no ha sido utilizado extensamente de acuerdo a la revisión de la literatura para el problema del *HFS* con *buffers* de tamaño limitado, consecuentemente, se evidenció una buena oportunidad de probar el rendimiento del mismo para problemas de secuenciamiento de tareas con estas nuevas características.

La segunda razón, es continuar una línea de investigación del grupo OPALO que busca ajustar el *hybrid flow shop* con modelos más cercanos a la realidad y respuestas de mejor calidad mediante la aplicación de metaheurísticas principalmente basadas en el *ICA*.

Finalmente, como tercera razón se mantiene como base de la metaheurística al *ICA* para realizar una comparación con respuestas obtenidas de investigaciones pasadas del grupo OPALO que utilizaron dicho algoritmo para modelar un *HFS* con máquinas paralelas no relacionadas y tiempos de alistamiento dependientes de la secuencia, pero con *buffers* ilimitados.

## **5.1 PRESENTACIÓN DETALLADA DE LA METAHEURÍSTICA Y DEL ALGORITMO**

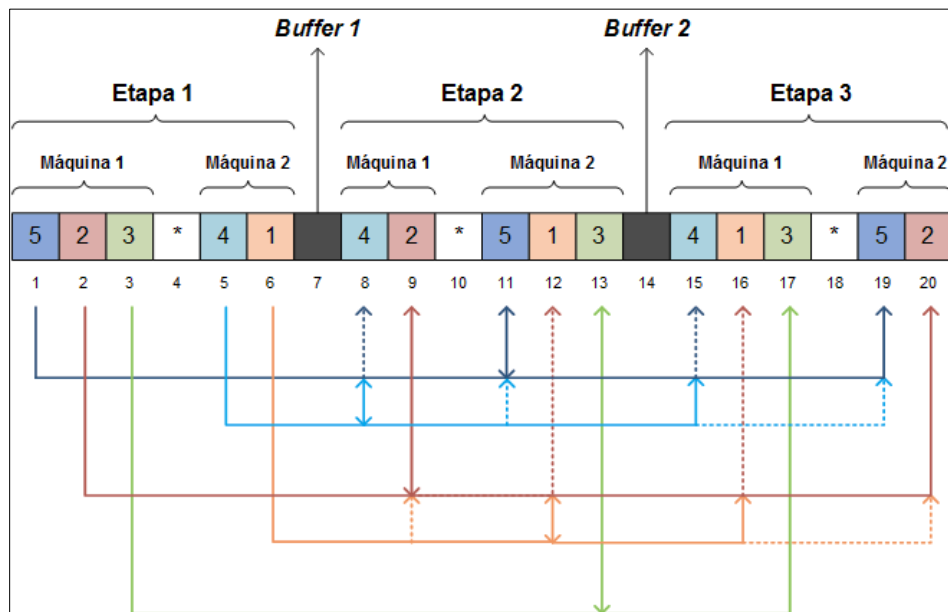
**5.1.1 Generación de los países iniciales.** Los países son la población inicial del algoritmo y se obtienen al generar una cantidad  $x$  de los mismos.

Un país es un vector de  $1 \times [(i + j) k - 1]$  el cual es la representación de la secuencia de procesamiento que llevarán los trabajos en las diferentes máquinas de todas las estaciones del *flow shop*. Dentro del vector,  $i$  simboliza el número de tareas a secuenciar,  $j$  es el número de máquinas en paralelo existentes en cada

una de las estaciones y finalmente, las etapas del proceso productivo vienen representadas por la  $k$ .

El vector funciona internamente calculando las posiciones relacionadas con las estaciones de procesamiento a través de la expresión  $(i + j)(k - 1)$  mientras la asignación de máquinas y tareas a cada una de las dichas estaciones se realiza mediante una distribución uniforme entre  $1$  y  $(i + j - 1)$ .

**Figura 9.** Muestra de creación de un país para la instancia i5j2k3-1.



Mediante la figura 9 se ilustra un ejemplo de generación de un país para un total de 5 tareas, 2 máquinas paralelas no relacionadas por estación, 3 estaciones de procesamiento y 1 espacio en el *buffer*.

$$i = 5, j = 2, k = 3.$$

$$\text{Vector generado: } 1 \times [(i + j) k - 1]$$

$$\text{Vector generado: } 1 \times [(5 + 2) 3 - 1] = 1 \times 20$$

$$\text{Posiciones de las estaciones dentro del vector generado: } (i + j)(k - 1)$$

Posición estación de trabajo 1:  $(5 + 2)(1 - 1) = 0$

Posición del *buffer 1* (cambio a la estación de trabajo 2):  $(5 + 2)(2 - 1) = 7$

Posición del *buffer 2* (cambio a la estación de trabajo 3):  $(5 + 2)(3 - 1) = 14$

La generación de los países iniciales comienza con componer completamente al azar la secuencia de los trabajos para la primera estación de procesamiento. Para el caso presentado en la figura 9 más adelante, se generan al azar los trabajos desde la posición 1 hasta la 6, donde el símbolo (\*) representa el cambio entre la máquina 1 y la máquina 2 dentro de la etapa 1.

El cambio de máquina para todas y cada una de las etapas se genera también al azar con una restricción de ubicación central, o que restringe generarse en las esquinas de la secuencia para la estación en cuestión (para el ejemplo de la figura 9, en la etapa 1, evitar generarse en las posiciones 1 o 6), esto con el fin de evitar que queden máquinas sin utilizarse y así sobrecargar de tareas las otras, lo cual a la larga lleva a bloqueos y peores resultados de *makespan* para la secuencia.

Para el caso de secuencias con tres máquinas por estación, a parte de la restricción de ubicación central también se incluye otra restricción que evita que los puntos de cambio de máquina queden en posiciones contiguas, con el mismo fin de evitar que queden máquinas sin trabajos por procesar. La decisión de incluir estas dos restricciones se basa en la observación de las características de las secuencias generadas por *GAMS* y que se supone, son las mejores respuestas de las instancias a evaluar, así, se busca que el algoritmo tenga más facilidades para generar secuencias parecidas o mejores a la presentadas en *GAMS* de forma más rápida y sencilla incluyendo estas dos restricciones.

Retomando el ejemplo de la figura 9 y habiendo terminado el secuenciamiento de la etapa 1, se inicia el de la estación 2. Para ordenar los trabajos de dicha etapa se utiliza una disciplina *FIFO* (*First In, First Out*) basada en la secuencia de la estación anterior; más específicamente, se supone que el primer trabajo asignado

a la primera máquina de la etapa 1 (el trabajo 5) siendo el primero en entrar a la estación es el primero en terminar su procesamiento, luego y en segundo lugar, termina el segundo trabajo en entrar a la estación, en otras palabras, el primer trabajo asignado a la segunda máquina de la etapa 1 (la tarea 4), posteriormente, el tercer trabajo en terminar su procesamiento es el segundo trabajo asignado a la primera máquina de la estación 1 (el trabajo 2) y así sucesivamente hasta el último trabajo de la secuencia en ser finalizado utilizando el método *FIFO* (en el ejemplo de la figura 9, la tarea 3).

Para el caso de secuencias con tres máquinas, el procedimiento es idéntico al presentado anteriormente, pero suponiendo que el primer trabajo de la tercera máquina para la etapa  $k$  sería el tercero en terminar su procesamiento dentro de la misma etapa y se desarrolla así sucesivamente para cada ronda de trabajos.

Continuando con el ejemplo de la figura 9, luego de tener el orden con el que los trabajos salen de la estación 1 se deben ordenar para la etapa 2, para eso se utiliza un procedimiento lógico donde el primer trabajo en salir de la estación 1 (la tarea 5) puede tomar solo dos posiciones, la primera posición de la máquina uno de la etapa 2 (posición 8) o la primera posición de la máquina dos de la etapa 2 (posición 11). La elección entre esas dos posiciones se hace completamente al azar entre la una o la otra porque de otra manera se debería comparar cuál posición de las posibles es mejor para un trabajo en cuestión de tiempo y eso podría llevar al algoritmo a arrojar soluciones en tiempos computacionales más grandes.

Posteriormente termina su procesamiento en la etapa 1 el trabajo 4, el cual podría tomar la primera posición de la máquina uno en la estación 2 (posición 8) o la primera posición de la máquina dos en la estación 2 (posición 11) pero al encontrarse ya la posición 11 ocupada entonces toma la opción restante (la posición 8) que es el primer espacio de la máquina uno en la etapa 2. Luego, termina en la estación 1 el trabajo 2 el cual a su vez puede tomar la segunda

posición de la máquina uno en la estación 2 (posición 9) o la segunda posición de la máquina dos en la etapa 2 (posición 12) y de nuevo la elección se hace completamente al azar entre estas dos por las razones ya mencionadas. A continuación, viene el trabajo 1, el cual podría tomar las mismas posiciones 9 o 12 del trabajo 2, pero de nuevo, al encontrarse ocupada la posición 9, debe tomar la posición 12 de la secuencia.

Finalmente, el trabajo 3 termina su procesamiento en la etapa 1 y por descarte se envía a la última y única posición disponible en la estación 2, la posición 13. El procedimiento se repite de la misma manera descrita para la etapa 3 pero ahora con base en el orden de salida *FIFO* por secuencia de los trabajos en la estación 2, en caso de existir más etapas también se realiza el mismo procedimiento hasta completar la secuencia entera del país.

Se elige este procedimiento lógico con el fin mantener un equilibrio en el número de trabajos para cada máquina, tratando de asignarlos en orden y en la misma cantidad para cada una de ellas y además, al ser los primeros trabajos en salir de la estación  $k$  los primeros en ser procesados en la etapa  $k+1$  se busca evitar estancias prolongadas dentro del *buffer* a la espera de que desocupen máquinas para cumplir con la secuenciación que podrían a su vez llevar a bloqueos de máquinas y a respuestas de mala calidad.

La decisión de tomar el método *FIFO* basado en la secuencia y no en la matriz de tiempos de completamiento para cada trabajo se justifica porque para hacerse de la segunda manera se tendría que ir calculando el *makespan* de cada trabajo para decidir cuál tarea sale primero de la estación  $k$  por tiempo real y de esa manera, algoritmo se volvería poco efectivo en cuestión de tiempo desde el mismo momento de la generación de los países por la cantidad de cálculos que debería hacer solo para generar los mismos. La cantidad de países a conformar inicialmente depende de un parámetro de entrada del algoritmo que se mostrará más adelante.

- **Cálculo del costo de los países iniciales.** Luego de generar la población de países iniciales, se debe calcular el costo inicial de cada uno de ellos; el costo es la variable que se busca optimizar mediante el algoritmo.

Para el caso de ésta investigación, dicha variable a optimizar es el *makespan* o tiempo máximo de completamiento representado por el momento en el que termina de procesarse la última tarea de la secuencia de trabajos en la última estación.

Retomando el ejemplo presentado en la figura 9 para la generación de países, se calcula el *makespan* del país representado, para ello se deben tener en cuenta tanto los tiempos de procesamiento del trabajo  $i5j2k3-1$  como los tiempos de alistamiento del mismo que se presentan en la tabla 2 y 3 respectivamente.

En la tabla 4 se muestra el procedimiento para el cálculo del *makespan* de la instancia  $i5j2k3-1$ , mientras en gráfica 8 se presenta el diagrama de *gantt* del mismo.

**Tabla 2.** Tiempos (u.t) de procesamiento para  $i5j2k3-1$ .

	<i>k1</i>		<i>k2</i>		<i>k3</i>	
	<i>j1</i>	<i>j2</i>	<i>j1</i>	<i>j2</i>	<i>j1</i>	<i>j2</i>
<i>i1</i>	104	125	98	123	90	86
<i>i2</i>	81	95	91	108	86	92
<i>i3</i>	108	103	115	111	103	80
<i>i4</i>	83	97	88	122	108	99
<i>i5</i>	91	105	116	82	118	78

**Tabla 3.** Tiempos (u.t) de alistamiento para  $i5j2k3-1$ .

<i>i</i>	<i>k1</i>									
	<i>j1</i>					<i>j2</i>				
	1	2	3	4	5	1	2	3	4	5
<i>0</i>	57	29	58	28	52	48	28	64	34	74
<i>1</i>		50	54	54	73		61	67	37	25
<i>2</i>	43		56	40	32	46		50	46	73
<i>3</i>	45	48		37	48	62	35		57	57
<i>4</i>	38	32	74		45	70	44	47		26
<i>5</i>	36	58	33	40		25	60	49	28	

	<i>k2</i>									
	<i>j1</i>					<i>j2</i>				
<i>i</i>	1	2	3	4	5	1	2	3	4	5
0	66	58	42	65	45	57	38	61	36	39
1		67	59	69	39		27	27	65	64
2	52		53	38	53	46		62	73	44
3	36	40		36	44	46	58		25	66
4	43	33	31		57	56	64	26		52
5	70	53	74	70		45	30	33	59	

	<i>k3</i>									
	<i>j1</i>					<i>j2</i>				
<i>i</i>	1	2	3	4	5	1	2	3	4	5
0	46	37	41	71	52	68	25	69	49	49
1		46	32	43	58		57	45	46	54
2	47		35	36	36	43		58	69	43
3	62	53		44	64	56	58		34	48
4	55	72	63		74	63	51	67		60
5	69	30	73	67		67	34	62	61	

**Tabla 4.** Cálculo del costo total del país i5j2k3-1.

	Etapa 1						Buffer 1	Etapa 2						Buffer 2	Etapa 3						
	Máquina 1			*	Máquina 2			Máquina 1			*	Máquina 2			Máquina 1			*	Máquina 2		
	5	2	3		4	1		4	2	5		1	3		4	1	3		5	2	
T.C.	91	81	108		97	125		88	91		82	123	111		108	90	103		78	92	
T.A.	52	58	56		34	70		65	33		39	45	27		71	55	32		49	34	
T.C. + T.A.	143	139	164		131	195		153	124		121	168	138		179	145	135		127	126	
<b>Makespan</b>	143	282	446		131	326		284	408		264	494	632		463	639	774		391	534	
<b>Bloqueos</b>																					

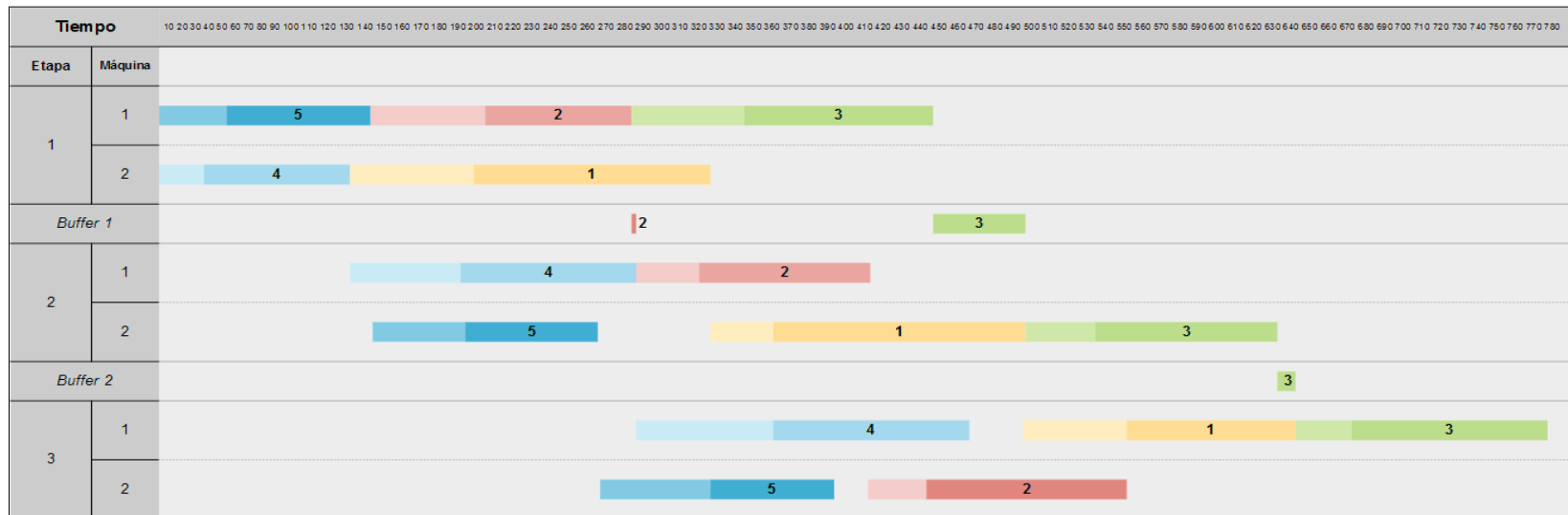
  

Trabajo	T.E.B.	T.S.B.
2	282	284
3	446	494

Trabajo	T.E.B.	T.S.B.
3	632	639

**Gráfica 8.** Diagrama de *gant* para el costo total o *makespan* del país i5j2k3-1.



- **Determinación de los países imperialistas.** Una vez finalizado el cálculo del costo de cada uno de los países iniciales se pasa a determinar qué países serán los imperialistas.

Los imperialistas serán aquellos países cuyo valor de *makespan* sea de los mínimos dentro del conjunto países de la población inicial, los países restantes, es decir, aquellos con un valor de *makespan* superior a los mínimos del conjunto serán las colonias dentro del algoritmo.

Posteriormente se realiza el cálculo del costo normalizado de los imperialistas a través de la ecuación (24) que es presentada a continuación:

$$C_n = \max c_i - c_n \quad (24)$$

Dónde:

$C_n$  es el costo normalizado del  $n$ -ésimo imperialista.

$c_i$  es el costo máximo dentro de todas las colonias.

$c_n$  es el costo del  $n$ -ésimo imperialista.

A continuación, con base en el costo normalizado encontrado anteriormente se calcula el poder normalizado.

El poder normalizado se utiliza para obtener el porcentaje de colonias que se le asignará a cada imperialista y con las cuales podrá compartir sus características, dicho poder normalizado es inversamente proporcional al costo normalizado total de los imperialistas y se calcula a través de la ecuación (25) presentada a continuación:

$$P_n = \left| \frac{C_n}{\sum_{i=1}^{N_{imp}} C_i} \right| \quad (25)$$

Dónde

$P_n$  es el poder normalizado del  $n$ -ésimo imperialista

$C_n$  es el costo normalizado del  $n$ -ésimo imperialista.

$N_{imp}$  es el número total de imperialistas.

$C_i$  es el costo normalizado del  $i$ -ésimo imperialista.

- **Asignación de colonias a los imperialistas constituyendo los imperios iniciales.** Mediante la ecuación (25) se ha conocido la proporción de colonias que se le asignará a cada imperialista, ahora, mediante la ecuación (26) se asignan la cantidad numérica exacta de colonias que pertenecerán a cada imperialista formando los imperios iniciales.

Las colonias son ordenadas mediante una distribución uniforme entre 1 y el número total colonias existentes dentro de la población inicial. El número concreto de colonias que debe tener cada imperialista se calculan mediante la ya mencionada ecuación (26) que se presenta a continuación:

$$NC_n = \text{round}\{P_n * N_{col}\} \quad (26)$$

Dónde:

$NC_n$  es el número de colonias a asignar al  $n$ -ésimo imperialista.

$P_n$  es el poder normalizado del  $n$ -ésimo imperialista.

$N_{col}$  es el número total de colonias dentro de la población inicial.

**5.1.2 Movimiento de los países colonia hacia su país imperialista.** Luego de tener el número exacto de colonias que pertenecen a cada uno de los imperialistas se procede a la formación de los imperios. Los imperios están conformados por el imperialista junto a sus respectivas colonias.

Teniendo el número exacto de colonias para cada imperialista se elige específicamente cuáles colonias van a pertenecer a cada imperio y esto se hace seleccionando de manera aleatoria dentro conjunto de países restantes (luego de elegir a los imperialistas) y teniendo en cuenta el número obtenido para las colonias de cada imperio en el paso anterior.

- **Cálculo del costo total de los imperios.** El costo total de un imperio viene dado por el poder del imperialista más un porcentaje del poder de sus respectivas colonias y se define mediante la siguiente ecuación:

$$TC_n = \text{costo}(\text{imperialista}_n) + \xi [\text{Costo promedio}(\text{colonias del imperio}_n)] \quad (27)$$

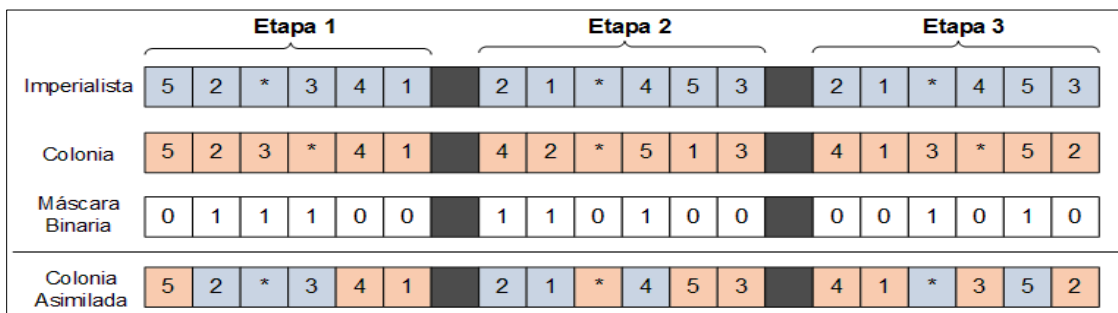
Donde  $TC_n$  es el costo total del  $n$ -ésimo imperio y  $\xi$  es una constante entre 0 y 1.

- **Asimilación del imperialista por parte de las colonias.** Basándose en la realidad del fenómeno del imperialismo, las mejores soluciones encontradas hasta ahora, los imperialistas, impondrán parte de sus características a las colonias dentro de su mismo imperio compartiendo información entre el mismo imperialista y sus colonias. En consecuencia, parte de la estructura de las colonias será igual al de su correspondiente imperialista.

El paso asimilación se le aplica a todas y cada una de las colonias de los diferentes imperios y funciona mediante la generación de un nuevo vector del mismo tamaño que las colonias, pero con valores de 0 o 1 (conocido como máscara binaria) para cada una de las posiciones disponibles.

Posteriormente, las posiciones con resultante 1 son copiadas primero y directamente desde el imperialista hacia la colonia asimilada, mientras las posiciones con resultante 0, son copiadas luego desde la colonia original hacia la colonia asimilada según sea necesario. Mediante la figura 10 se detalla el procedimiento de asimilación.

**Figura 10.** Asimilación del imperialista por parte de las colonias.

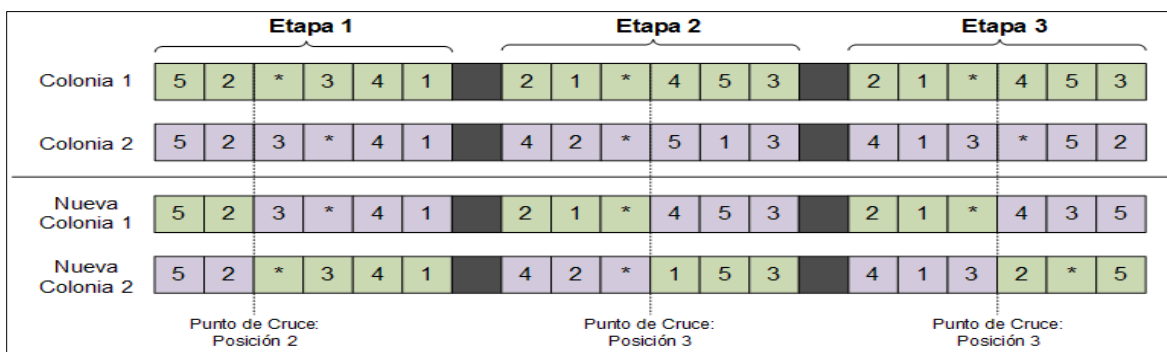


En la figura 10 podemos observar que para la estación 1, los trabajos 2, 3 y (\*) del cambio de máquina son tomados directamente del imperialista hacia la colonia asimilada puesto que los valores en la máscara binaria para estas posiciones fueron de 1, mientras que los demás trabajos (el 5, 4 y 1) son copiados según sea necesario desde la colonia original hacia la colonia asimilada con base en el orden que tienen en la colonia original. El procedimiento se repite para cada una de las etapas y al final, la nueva colonia asimilada generada reemplaza siempre a la antigua colonia, esto con el fin de mantener constante el número de países.

**5.1.3 Información compartida entre las colonias.** Para este paso, buscando mejorar y así derrocar a su imperialista las colonias comparten información entre ellas.

Para el procedimiento de información compartida entre las colonias se adecúan operadores de cruce (de un punto) tomados desde los algoritmos genéticos que trabajan sobre un porcentaje de la población que intercambia características y está dado por un PCrossover (PC).

**Figura 11.** Procedimiento de información compartida entre las colonias.



En el ejemplo presentado en la figura 11 se puede observar que los puntos de cruce para las etapas 1, 2 y 3 fueron las posiciones 2, 3 y 3 respectivamente, dichos puntos de cruce fueron elegidos de forma completamente aleatoria para cada estación.

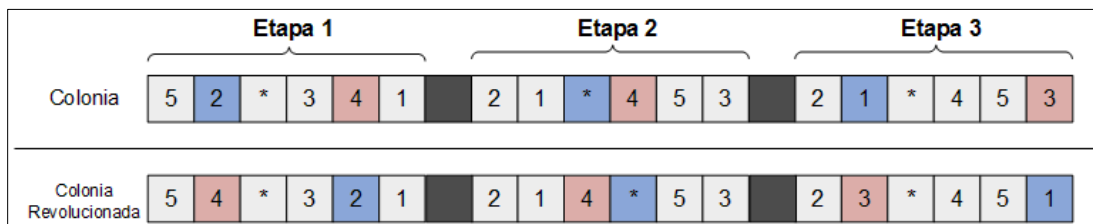
Para el caso de la nueva colonia 1 del ejemplo, en la etapa de procesamiento 1, se copia la información idéntica de la colonia 1 hasta el punto de cruce de la posición 2, luego, desde el punto de cruce hasta el final de la etapa se copian los valores de la colonia 2 según sea necesario. Este procedimiento se repite para ambas colonias nuevas (para la nueva colonia 2 se copia hasta el punto de cruce la información de la colonia 2 y luego del mismo de la colonia 1 según sea necesario) y para cada una de las estaciones.

- **Actualización del imperio comparando colonias antiguas y recién generadas.** Tras realizar los cruces entre colonias y que las mismas compartan información entre ellas se procede a calcular el costo o *makespan* de las colonias originales que compartieron información y las nuevas colonias generadas, esto se realiza con el objetivo de escoger las dos de mejor *makespan* y esas que tomen los lugares de las colonias iniciales, mientras las otras dos se eliminan y así no aumentar el tamaño de la población.

**5.1.4 Revolución de los países colonia.** Algunas colonias con el fin de ser las mejores de su imperio deciden revolucionarse, para el caso del algoritmo sería mejorar varias de sus características.

Para la revolución se hace uso del operador de mutación de salto tomado de nuevo de los algoritmos genéticos, para explicar el funcionamiento de este paso se presenta a continuación la figura 12 con un ejemplo.

**Figura 12.** Procedimiento de revolución de los países colonia.



Como se puede observar en la figura 12, en la estación 1 las posiciones 2 y 5 que representan los trabajos 2 y 4 respectivamente fueron elegidas de manera

aleatoria para aplicar el operador de salto. Luego, en la colonia revolucionada los trabajos 2 y 4 aparecen, pero con posiciones intercambiadas con respecto a la colonia original, el procedimiento se repite para las etapas 2 y 3 donde intercambian posiciones los trabajos 4 y el cambio de máquina (\*) para la estación 2 y los trabajos 1 y 3 para la etapa 3. Finalmente, la colonia revolucionada toma el puesto dentro del imperio de la colonia antigua y la reemplaza; la colonia original se elimina con el fin de mantener constante el número de países dentro del algoritmo.

- **Intercambio de posiciones del imperialista con las colonias.** Para esta parte se debe realizar el cálculo del *makespan* de todas y cada una de las colonias dentro del imperio con el fin de que, si se encuentra alguna con un mejor costo o menor *makespan* que el imperialista, intercambien posiciones y la colonia pase a ser el nuevo imperialista mientras que el viejo imperialista se une al grupo de colonias del imperio.

- **Cálculo del costo total de los imperios.** Con base en la ecuación presentada en la sección 5.1.2 y como secuela de los cambios en la secuenciación de las colonias por los procesos de asimilación, información compartida y revolución que han sufrido las mismas dentro de los distintos imperios, se debe volver a calcular el costo total de todos los imperios ya que no es el mismo que se obtuvo al comienzo.

**5.1.5 Guerra entre los imperios y supresión del imperio más débil.** Buscando ser cada vez más poderosos, los imperios más fuertes comienzan el proceso de despojar de sus colonias al imperio más débil para así poder eliminarlo.

Se considera el imperio más débil a aquel cuyo costo es el más elevado entre los diferentes imperios existentes. En dado caso que existan dos imperios débiles o con el mismo costo superior al del resto de imperios, el imperio a suprimir será el

que tenga el peor imperialista; si de igual manera persiste la igualdad entre ambos imperios débiles entonces se elegirá al azar cuál de los dos eliminar.

La cantidad de colonias que cada imperio fuerte quitará al más débil dependerá directamente del costo del imperio fuerte, de tal manera que entre mejor costo tenga el imperio fuerte más colonias despojará del imperio a suprimir. Para realizar este procedimiento se utilizan las dos ecuaciones presentadas a continuación:

$$DifCost_n = TC_{id} - TC_n \quad (28)$$

Dónde:

$DifCost_n$  es la diferencia en el costo entre el  $n$ -ésimo imperio fuerte y el imperio más débil.

$TC_{id}$  es el costo del imperio más débil.

$TC_n$  es el costo del  $n$ -ésimo imperio fuerte.

$$\#C_n = \frac{DifCost_n}{\sum_{i=1}^N DifCost_i} * Col_{id}$$

Dónde:

$\#C_n$  es el número de colonias que el  $n$ -ésimo imperio fuerte tomará del más débil.

$DifCost_n$  es la diferencia en el costo entre el  $n$ -ésimo imperio fuerte y el imperio más débil.

$N$  es el número de imperios con mejor costo que el imperio más débil.

$DifCost_i$  es la diferencia en el costo entre el  $i$ -ésimo imperio fuerte y el imperio más débil.

$Col_{id}$  es el número de países (incluye colonias e imperialista) dentro del imperio más débil.

- **Eliminación del imperio más débil.** Tras determinar el número de colonias que cada imperio tomará del imperio más débil, se efectúa el despejo de dichas colonias.

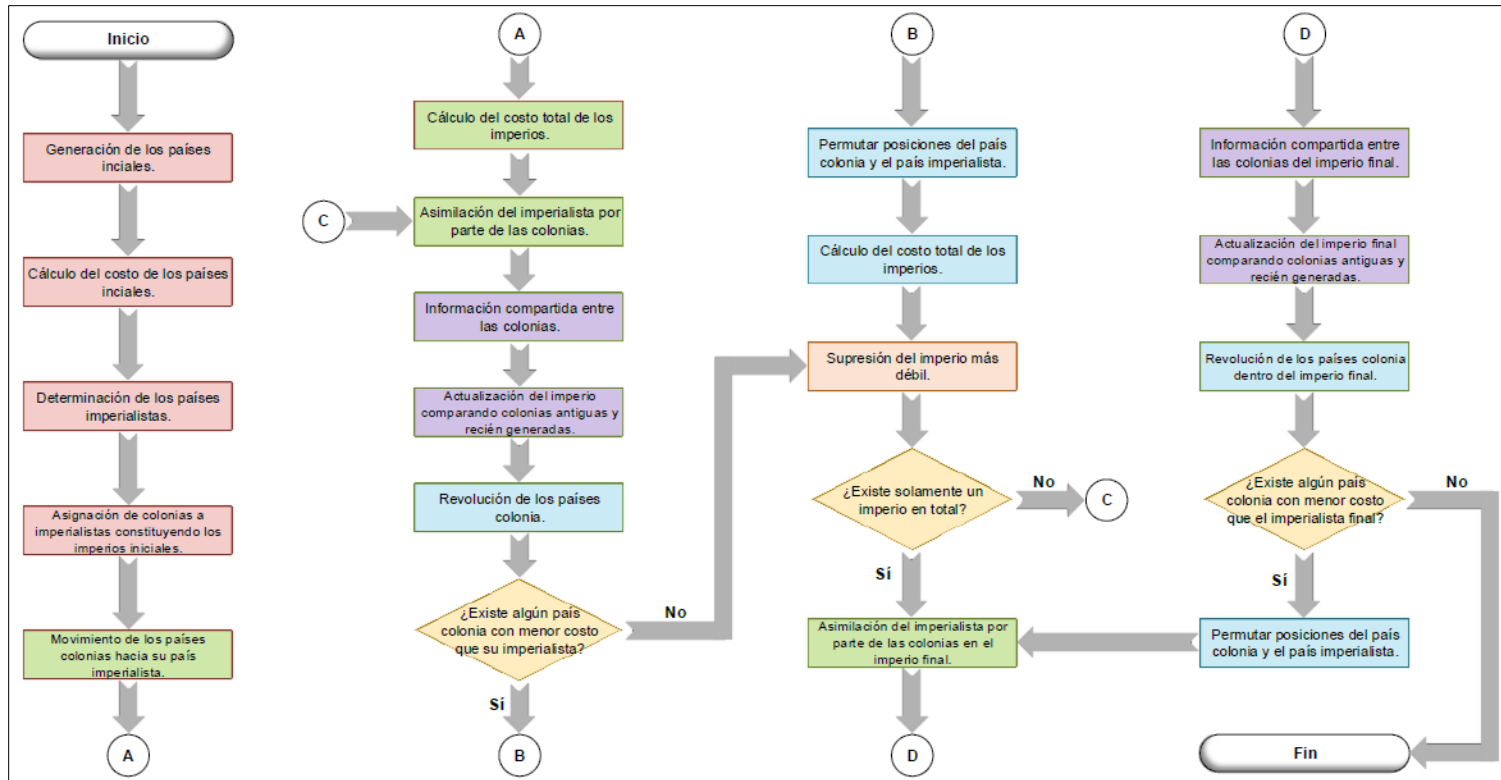
Para establecer cuáles colonias va a tomar cada imperio del imperio a eliminar, cada uno de los imperios fuertes toma al azar la cantidad de colonias que le corresponden con base en lo determinado al comienzo del paso 5.1.5.

Desde este momento se realizan las  $n$  iteraciones necesarias hasta lograr obtener un solo imperio que posea todas las colonias. Cada iteración consiste en repetir los procedimientos del paso de asimilación del imperialista por parte de las colonias hasta la eliminación del imperio más débil.

**5.1.6 Condición de parada.** Al momento de tener un solo imperio que domina todas las colonias, se realizan una asimilación final, un proceso de compartir información entre colonias final y una revolución de colonias también final y se guarda la respuesta.

Mediante la figura 13 de la siguiente página se presenta el diagrama de flujo de la metaheurística.

Figura 13. Diagrama de flujo de la metaheurística propuesta.



Fuente: Moratto y Pérez.

Los pasos con color rojo pertenecen al proceso de generación de países iniciales, con color verde se presentan los pertenecientes al proceso de movimiento de los países colonia hacia el imperialista, hacen parte del proceso de información compartida entre las colonias los de color morado, los pasos de color celeste hacen parte del proceso de revolución de los países colonia, finalmente, el de color rosa hace parte de la guerra entre los imperios.

## 6. PROCESO DE VALIDACIÓN Y RESULTADOS DE LA METAHEURÍSTICA

El desempeño de la metaheurística planteada es evaluado a través de 20 instancias (que se detallan en el anexo B) con el fin de comparar la eficiencia del mismo frente a las respuestas obtenidas mediante el método exacto. El pseudocódigo de la metaheurística y del algoritmo junto a sus adaptaciones para el problema con *buffers* limitados puede examinarse en el anexo C.

La codificación del algoritmo fue escrita en lenguaje de programación *Matlab* en su versión R2012a. Los parámetros incluidos en el proceso de validación y evaluación del algoritmo son los siguientes: la cantidad de países, el porcentaje de países imperialistas, el porcentaje de países que comparten información entre sí (también conocido como porcentaje de cruce de colonias), el peso de las colonias en el cálculo del poder total de imperio ( $\% \xi$ ) y finalmente, el porcentaje de revolución de los países colonia.

Para el proceso de comparación con los resultados del método exacto se obtiene el promedio de las 5 réplicas de las respuestas obtenidas para cada una de las 32 combinaciones posibles de tratamientos en un diseño experimental  $2^k$  con base en las combinaciones de los 5 parámetros presentados en la tabla 5 y se comparan mediante porcentaje de diferencia con la solución obtenida mediante *GAMS*. El detalle de las 5 réplicas, para cada una las 32 combinaciones posibles del diseño experimental en las 20 instancias evaluadas se pueden detallar en el anexo D.

Además, se realiza una comparación mediante una prueba de hipótesis de medias con los resultados obtenidos en el proyecto de grado de Moratto y Pérez<sup>121</sup> que plantean también una metaheurística basada en el *ICA* para el *HFS* con máquinas paralelas no relacionadas y tiempos de alistamiento dependientes de la secuencia,

---

<sup>121</sup> MORATTO CHIMENTY, ESELENDIS; PÉREZ FIGUEREDO, LUIS EDUARDO. Metaheurística basada en el algoritmo competitivo imperialista (*ICA*) aplicada a la solución del problema del *flow shop híbrido (HFS)* con máquinas paralelas no relacionadas. Trabajo de grado. Ingeniería Industrial. Universidad Industrial de Santander. 2016.

pero asumen *buffers* de tamaño ilimitado esto con el fin de comparar la efectividad de ambas metaheurísticas en instancias similares y tener una idea del impacto de los *buffers* reducidos sobre el *makespan* de las soluciones arrojadas por el algoritmo. El detalle y análisis de los resultados obtenidos en la comparación se puede detallar en el anexo E.

También, se realiza un diseño experimental para determinar los factores más influyentes en el desempeño de la metaheurística, para eso, se escoge un diseño factorial fraccionado  $2^{k-1}$  donde se tienen en cuenta los 5 factores (k) y dos niveles de cada factor presentados en la tabla 5. El detalle del diseño experimental junto a los resultados y análisis del mismo se presenta en el anexo F, mientras los datos utilizados para la realización del experimento se enseñan en el anexo G.

Mediante la tabla 6 presentada a continuación se realiza la relación de los factores que tienen influencia en el rendimiento del algoritmo junto a sus respectivos niveles, los valores corresponden a los utilizados por Rabiee et al.<sup>122</sup> quienes fueron los desarrolladores del híbrido del algoritmo competitivo imperialista con elementos de algoritmos genéticos y los utilizaron para dar solución a un problema de *flow shop* híbrido flexible con dos etapas de procesamiento y máquinas paralelas no relacionadas dentro de cada etapa. Ese mismo híbrido es el principal componente de la metaheurística utilizada en el presente trabajo.

**Tabla 5.** Factores y niveles del *ICA*.

Factor	Nivel bajo	Nivel alto
Número de países	100	250
% Imperialistas	0.10	0.25
% $\xi$	0.30	0.50
% Cruce	0.40	0.60
% Revolución	0.20	0.40

Todas las corridas experimentales que se tienen en cuenta para todas las validaciones y comparaciones se ejecutaron en una computadora con procesador *Intel Core i5-4570*, 3.2 GHz con una memoria *RAM* de 8 *Gigabytes* y un sistema

<sup>122</sup> RABIEE, M. et al. Op cit, p. 1229-1245.

operativo de 64 *bits*. En total se corre la metaheurística cerca de 3400 veces en total para realizar los todos los experimentos.

Para cada una de las instancias se presenta un detalle de los resultados junto a un análisis de la mejor respuesta obtenida y la combinación de niveles con la mejor respuesta en promedio, también se enseña un resumen de los resultados de la comparación con el *ICA* para *HFS* con *buffers* ilimitados y del diseño de experimentos para determinar los factores más influyentes en los resultados de cada instancia.

## 6.1 DETALLE Y ANÁLISIS DE LOS RESULTADOS

Desde la tabla 6 hasta la tabla 25 se detallan los resultados obtenidos en cada una de las instancias utilizadas para el problema del *HFS* con *buffers* limitados. Para ello se tienen en cuenta los parámetros presentados a continuación:

- **CP:** Cantidad inicial de países del algoritmo.
- **% Imp:** Porcentaje de imperialistas que tendrá la corrida.
- **%  $\xi$ :** Influencia de las colonias en el cálculo del poder total de los imperios.
- **% Cr:** Porcentaje de cruce o de colonias que comparte información.
- **% Rev:** Porcentaje de revolución dentro de las colonias.
- **SC:** Solución conocida obtenida a través del método exacto (en unidades de tiempo).
- **Med:** Media de las cinco réplicas de cada combinación de tratamientos establecida (en unidades de tiempo).
- **SD:** Desviación estándar de las cinco réplicas de cada combinación de tratamientos establecida (en unidades de tiempo).
- **% Dif:** Diferencia en porcentaje entre la solución conocida mediante el método exacto (SC) y la media de las cinco réplicas para cada combinación de tratamientos (Med).
- **MO:** Mejor solución obtenida por la metaheurística para cada una de las combinaciones (en unidades de tiempo).

- **Med-t:** Media de los tiempos de obtención de las respuestas para las cinco réplicas de cada combinación de tratamientos establecida (en segundos).
- **SD-t:** Desviación estándar de los tiempos de obtención de las respuestas para las cinco réplicas de cada combinación de tratamientos establecida (en segundos).

**Tabla 6.** Resultados obtenidos para la instancia i7j2k3-1.

#	CP	%Imp	% $\xi$	%Cr	%Rev	SC [ut]	MO [ut]	Med [ut]	%Dif	SD [ut]	Med-t [s]	SD-t [s]
1	100	0.1	0.3	0.4	0.2	873	882	895.8	2.612%	8.4083	6.68189	0.4445
2	250	0.1	0.3	0.4	0.2	873	903	909.6	4.192%	6.7676	12.9073	0.1078
3	100	0.25	0.3	0.4	0.2	873	887	894.6	2.474%	8.1731	8.63927	0.0415
4	250	0.25	0.3	0.4	0.2	873	901	910.2	4.261%	9.2304	23.4442	0.0654
5	100	0.1	0.3	0.6	0.2	873	893	903.2	3.459%	7.328	6.30868	0.0167
6	250	0.1	0.3	0.6	0.2	873	892	906.2	3.803%	10.986	12.2904	0.0627
7	100	0.25	0.3	0.6	0.2	873	889	906	3.780%	9.9499	8.14358	0.0201
8	250	0.25	0.3	0.6	0.2	873	895	907.8	3.986%	8.9833	26.04	0.1244
9	100	0.1	0.3	0.4	0.4	873	901	905	3.666%	3.1623	6.90768	0.0535
10	250	0.1	0.3	0.4	0.4	873	891	906.4	3.826%	11.803	13.8579	0.0624
11	100	0.25	0.3	0.4	0.4	873	890	899.6	3.047%	6.5422	8.8936	0.0527
12	250	0.25	0.3	0.4	0.4	873	911	927.6	6.254%	12.641	26.5964	0.0772
13	100	0.1	0.3	0.6	0.4	873	901	903.4	3.482%	3.2863	6.94455	0.0436
14	250	0.1	0.3	0.6	0.4	873	884	897.8	2.841%	10.305	14.1306	0.2872
15	100	0.25	0.3	0.6	0.4	873	910	914.2	4.719%	3.1145	9.01916	0.058
16	250	0.25	0.3	0.6	0.4	873	891	916.4	4.971%	18.022	27.3508	0.1324
17	100	0.1	0.5	0.4	0.2	873	894	901.4	3.253%	5.9833	6.8163	0.0375
18	250	0.1	0.5	0.4	0.2	873	906	910.2	4.261%	3.8987	12.9913	0.1041
19	100	0.25	0.5	0.4	0.2	873	890	901.4	3.253%	9.3702	8.88456	0.1632
20	250	0.25	0.5	0.4	0.2	873	896	904.2	3.574%	6.7231	25.8197	0.4366
21	100	0.1	0.5	0.6	0.2	873	885	891.8	2.153%	3.9623	7.05883	0.137
22	250	0.1	0.5	0.6	0.2	873	889	897.8	2.841%	7.0498	13.3649	0.1063
23	100	0.25	0.5	0.6	0.2	873	893	904.8	3.643%	8.6429	8.95481	0.1036
24	250	0.25	0.5	0.6	0.2	873	889	905	3.666%	10.677	26.5055	0.4517
25	100	0.1	0.5	0.4	0.4	873	887	897	2.749%	7.746	6.9707	0.0623
26	250	0.1	0.5	0.4	0.4	873	891	907.6	3.963%	9.6333	14.1584	0.1004
27	100	0.25	0.5	0.4	0.4	873	912	915.2	4.834%	2.1679	8.91712	0.0273
28	250	0.25	0.5	0.4	0.4	873	914	926.4	6.117%	10.262	30.5141	0.534
29	100	0.1	0.5	0.6	0.4	873	882	891.2	2.085%	5.5408	7.84753	0.1347
30	250	0.1	0.5	0.6	0.4	873	894	901.6	3.276%	7.3689	15.8666	0.2789
31	100	0.25	0.5	0.6	0.4	873	890	906.8	3.872%	10.826	13.0164	0.40612
32	250	0.25	0.5	0.6	0.4	873	926	932.6	6.827%	5.9414	31.9562	0.0993

Como se puede observar en la tabla 6, para la instancia i7j2k3-1 (7 trabajos, 2 máquinas por estación, 3 etapas de procesamiento, tamaño de *buffer* 1) la metaheurística basada en el *ICA* no encuentra ninguna respuesta igual o mejor a la obtenida mediante el método exacto, la más cercana a la solución conocida es 882 u.t. que se obtiene para las combinaciones 1 y 29 de niveles de tratamientos. Esto puede ser causado porque a pesar de que *GAMS* debido a la baja complejidad computacional de las instancias más pequeñas arroja respuestas de muy buena calidad.

A pesar de esto, la metaheurística en promedio obtiene respuestas con porcentaje de diferencia inferior al 7% con la solución conocida en todas las posibles combinaciones de niveles y, las respuestas más cercanas a la obtenida mediante el método exacto se consiguen con la combinación 29 con una distancia del 2,085%.

Para el caso de la instancia actual, se encuentra que la metaheurística basada en el *ICA* para el *HFS* con *buffers* limitados tamaño uno obtiene secuencias con *makespan* superiores a las de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado (ver anexo E). También se determina que el factor más influyente en la variable respuesta es la interacción entre el % de imperialistas y el % de revolución, mientras el que menos influye es la interacción entre el número de países y el %  $\xi$  (ver anexo F).

También vale la pena destacar que la metaheurística obtiene respuestas de buena calidad en tiempos en promedio inferiores a 32 s. para todas las posibles combinaciones de tratamientos frente a los 10800 s. del método exacto.

Para la instancia *i7j2k3-2* (7 trabajos, 2 máquinas por estación, 3 etapas de procesamiento, tamaño de *buffer* 2) presentada en la tabla 7 se observa que no se obtuvo ninguna respuesta igual o mejor a la arrojada por *GAMS*, siendo la más cercana 853 u.t. emanada de la combinación de niveles 9. Como en la instancia anterior, puede ser causado por la buena calidad de la respuesta obtenida mediante el método exacto debido a la baja complejidad computacional a pesar de sobrepasar los límites de tiempo establecidos.

De nuevo, a pesar de esto, el algoritmo obtuvo respuestas con porcentajes de diferencia inferiores al 10% con respecto a la solución conocida, destacando la combinación de niveles de tratamientos 14 con una diferencia de 4,751%. Es importante destacar la buena calidad de las respuestas obtenidas pues se comparaban con una solución conocida que presentaba un *gap* relativo de menos del 10% (ver tabla 1).

**Tabla 7.** Resultados obtenidos para la instancia i7j2k3-2.

#	CP	%Imp	%ξ	%Cr	%Rev	SC [ut]	MO [ut]	Med [ut]	%Dif	SD [ut]	Med-t [s]	SD-t [s]
1	100	0,1	0,3	0,4	0,2	842	878	889,6	5,653%	12,621	6,95509	0,5907
2	250	0,1	0,3	0,4	0,2	842	883	896,8	6,508%	15,531	12,94526	0,1097
3	100	0,25	0,3	0,4	0,2	842	872	898,4	6,698%	15,978	8,708076	0,1054
4	250	0,25	0,3	0,4	0,2	842	892	909,4	8,005%	15,931	23,62636	0,1011
5	100	0,1	0,3	0,6	0,2	842	886	891,2	5,843%	3,8987	6,312026	0,0151
6	250	0,1	0,3	0,6	0,2	842	890	897,2	6,556%	6,0992	12,2644	0,0584
7	100	0,25	0,3	0,6	0,2	842	876	883,4	4,917%	6,6558	8,190032	0,0559
8	250	0,25	0,3	0,6	0,2	842	875	913,4	8,480%	23,671	24,24608	0,0983
9	100	0,1	0,3	0,4	0,4	842	853	882,2	4,774%	17,398	6,415768	0,0145
10	250	0,1	0,3	0,4	0,4	842	896	902,2	7,150%	5,4955	12,74502	0,0292
11	100	0,25	0,3	0,4	0,4	842	876	891,6	5,891%	13,831	8,28755	0,0399
12	250	0,25	0,3	0,4	0,4	842	901	920,4	9,311%	15,915	24,95314	0,0909
13	100	0,1	0,3	0,6	0,4	842	891	898	6,651%	4,899	6,43967	0,0083
14	250	0,1	0,3	0,6	0,4	842	877	882	4,751%	5,1478	12,86214	0,0535
15	100	0,25	0,3	0,6	0,4	842	881	893	6,057%	8,86	8,39842	0,0185
16	250	0,25	0,3	0,6	0,4	842	894	911,6	8,266%	10,286	25,94788	0,4299
17	100	0,1	0,5	0,4	0,2	842	879	891,8	5,914%	9,2033	6,301048	0,0279
18	250	0,1	0,5	0,4	0,2	842	881	902,6	7,197%	14,926	12,28728	0,2106
19	100	0,25	0,5	0,4	0,2	842	875	896	6,413%	14,018	8,150968	0,0808
20	250	0,25	0,5	0,4	0,2	842	888	900	6,888%	11,247	23,9436	0,4486
21	100	0,1	0,5	0,6	0,2	842	889	900,4	6,936%	6,9857	6,41644	0,1943
22	250	0,1	0,5	0,6	0,2	842	883	890,8	5,796%	7,225	12,54546	0,1399
23	100	0,25	0,5	0,6	0,2	842	884	897,2	6,556%	10,257	8,247062	0,0438
24	250	0,25	0,5	0,6	0,2	842	890	909,8	8,052%	11,925	24,664	0,3208
25	100	0,1	0,5	0,4	0,4	842	885	891	5,819%	4,4721	7,75298	0,1291
26	250	0,1	0,5	0,4	0,4	842	881	886,2	5,249%	6,7602	15,68382	0,3484
27	100	0,25	0,5	0,4	0,4	842	869	882,2	4,774%	12,235	10,095668	0,2058
28	250	0,25	0,5	0,4	0,4	842	900	909,4	8,005%	7,8294	27,50042	0,6834
29	100	0,1	0,5	0,6	0,4	842	872	888,4	5,511%	10,31	7,117718	0,1946
30	250	0,1	0,5	0,6	0,4	842	867	885,6	5,178%	11,127	14,12004	0,2207
31	100	0,25	0,5	0,6	0,4	842	869	887	5,344%	12,145	9,159118	0,057
32	250	0,25	0,5	0,6	0,4	842	891	908,6	7,910%	13,612	28,08546	0,7035

En la presente instancia, se determina también que la metaheurística basada en el *ICA* para el *HFS* con *buffers* limitados tamaño dos obtiene secuencias con *makespan* superiores a las de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado (ver anexo E). A su vez, se determina que el factor más influyente en la variable respuesta es el % de imperialistas mientras el que menos influye es el % de cruce de información entre colonias (ver anexo F).

Finalmente, con respecto a los tiempos de obtención de respuesta, de nuevo ninguno supera los 29 s. en promedio para todas las combinaciones posibles frente a los 10800 s. del método exacto.

Con respecto a la instancia presentada en la tabla 8, es decir, la instancia i7j3k3-1 (7 trabajos, 3 máquinas por estación, 3 etapas de procesamiento, tamaño de *buffer* 1) la metaheurística no logró encontrar respuesta igual o mejor a de la

solución conocida siendo la más cercana a este valor la presentada en la combinación 20 con 697 u.t. En contraste con las dos instancias anteriores, esta vez la metaheurística tuvo varias combinaciones con porcentaje de diferencia superior al 10% con respecto a la mejor conocida, pero a su vez siempre inferiores al 12%.

**Tabla 8.** Resultados obtenidos para la instancia i7j3k3-1.

#	CP	%Imp	%ξ	%Cr	%Rev	SC [ut]	MO [ut]	Med [ut]	%Dif	SD [ut]	Med-t [s]	SD-t [s]
1	100	0.1	0.3	0.4	0.2	657	718	723.2	10.076%	4.3243	6.91413	0.2479
2	250	0.1	0.3	0.4	0.2	657	715	723.8	10.167%	8.0436	11.8787	0.0538
3	100	0.25	0.3	0.4	0.2	657	723	727.8	10.776%	5.2631	8.41379	0.0446
4	250	0.25	0.3	0.4	0.2	657	710	723.4	10.107%	10.714	22.7116	0.0437
5	100	0.1	0.3	0.6	0.2	657	707	712.2	8.402%	4.5497	6.80627	0.0614
6	250	0.1	0.3	0.6	0.2	657	711	714.2	8.706%	3.0332	12.1561	0.3341
7	100	0.25	0.3	0.6	0.2	657	701	710.8	8.189%	7.0498	8.49645	0.0541
8	250	0.25	0.3	0.6	0.2	657	717	721.4	9.802%	3.2094	23.413	0.243
9	100	0.1	0.3	0.4	0.4	657	705	714.6	8.767%	6.269	6.78261	0.0664
10	250	0.1	0.3	0.4	0.4	657	708	718.6	9.376%	5.9833	11.9814	0.0369
11	100	0.25	0.3	0.4	0.4	657	712	719.8	9.559%	4.7645	8.43265	0.0609
12	250	0.25	0.3	0.4	0.4	657	706	730	11.111%	14.195	22.5596	0.1657
13	100	0.1	0.3	0.6	0.4	657	711	714.6	8.767%	2.881	6.95057	0.0378
14	250	0.1	0.3	0.6	0.4	657	718	720.6	9.680%	2.7019	12.1771	0.1128
15	100	0.25	0.3	0.6	0.4	657	713	721.2	9.772%	6.0166	9.25921	1.4373
16	250	0.25	0.3	0.6	0.4	657	730	732.4	11.476%	2.4083	23.2703	0.2048
17	100	0.1	0.5	0.4	0.2	657	707	714	8.676%	4.7958	6.91353	0.0248
18	250	0.1	0.5	0.4	0.2	657	707	717.4	9.193%	9.2087	11.9481	0.0984
19	100	0.25	0.5	0.4	0.2	657	699	711.8	8.341%	8.167	8.57989	0.0592
20	250	0.25	0.5	0.4	0.2	657	697	725	10.350%	16.462	23.0873	0.1716
21	100	0.1	0.5	0.6	0.2	657	699	711.2	8.250%	8.4083	6.89522	0.0347
22	250	0.1	0.5	0.6	0.2	657	722	727.8	10.776%	5.7184	12.123	0.1044
23	100	0.25	0.5	0.6	0.2	657	718	721.4	9.802%	2.9665	8.67645	0.0772
24	250	0.25	0.5	0.6	0.2	657	706	718	9.285%	7.4498	24.9364	2.771
25	100	0.1	0.5	0.4	0.4	657	702	709	7.915%	6.1644	6.92591	0.0183
26	250	0.1	0.5	0.4	0.4	657	723	731.2	11.294%	4.9699	12.1417	0.1194
27	100	0.25	0.5	0.4	0.4	657	715	719.4	9.498%	4.0373	8.51515	0.0256
28	250	0.25	0.5	0.4	0.4	657	721	730	11.111%	5.4314	22.8234	0.2075
29	100	0.1	0.5	0.6	0.4	657	722	723.4	10.107%	1.3416	6.95146	0.0576
30	250	0.1	0.5	0.6	0.4	657	711	719.2	9.467%	5.1186	12.31	0.0882
31	100	0.25	0.5	0.6	0.4	657	716	718.8	9.406%	1.9235	9.30811	0.0373
32	250	0.25	0.5	0.6	0.4	657	721	728	10.807%	4.3012	24.6268	0.2639

La que mejor desempeño tuvo fue la combinación de niveles de tratamientos 25 con porcentaje de diferencia del 7,915%. Este cambio en la calidad de las soluciones se puede deber al manejo de tres máquinas por estación que aumenta el tamaño de los países y así mismo la cantidad posible de permutaciones de trabajos y cambios de máquina que pueden presentar las respuestas haciendo así para el algoritmo más difícil encontrar mejores soluciones.

También, se determina para la actual instancia que la metaheurística basada en el *ICA* para el *HFS* con *buffers* limitados tamaño uno obtiene secuencias con *makespan* superiores a las de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado. (ver anexo E). Por su parte, se determina que el factor más influyente en la variable respuesta para la presente instancia es el número de países, mientras el menos influyente es la interacción entre el número de países y el % de revolución (ver anexo F).

Por último, los tiempos promedio de obtención de respuestas para esta instancia son excelentes, encontrando respuesta siempre debajo de los 25 s. en promedio para cualquier combinación de niveles versus los 10800 s. del método exacto.

En la tabla 9 se muestran los resultados para la instancia *i7j3k3-2* (7 trabajos, 3 máquinas por estación, 3 etapas de procesamiento, tamaño de *buffer 2*) donde se puede evidenciar que la metaheurística no encuentra para ninguna combinación de niveles de tratamientos una respuesta mejor a la obtenida mediante el método exacto. La más cercana es la presentada como mejor solución para la combinación 5 con 692 u.t.

Al igual que para la instancia anterior, las respuestas en promedio encontradas son de menor calidad que para instancias con dos máquinas por estación. Todas las combinaciones presentan un porcentaje de diferencia entre el 12% y el 17%, destacándose sobre las demás la combinación 5 con una distancia promedio a la solución conocida del 12,23%. El incremento en el porcentaje de diferencia de las respuestas se puede deber tanto al manejo de tres máquinas como también a la buena calidad de la respuesta obtenida por el método exacto para la instancia actual (*i7j3k3-2*) que presenta un *gap* relativo del 12,4158% (ver tabla 1).

**Tabla 9.** Resultados obtenidos para la instancia i7j3k3-2.

#	CP	%Imp	% $\xi$	%Cr	%Rev	SC [ut]	MO [ut]	Med [ut]	%Dif	SD [ut]	Med-t [s]	SD-t [s]
1	100	0.1	0.3	0.4	0.2	628	708	718.4	14.39%	6.269	7.0378	0.1156
2	250	0.1	0.3	0.4	0.2	628	721	724.8	15.41%	2.3875	12.3363	0.2521
3	100	0.25	0.3	0.4	0.2	628	706	713.6	13.63%	5.3666	8.7217	0.21
4	250	0.25	0.3	0.4	0.2	628	716	723	15.13%	5.099	23.3585	0.1993
5	100	0.1	0.3	0.6	0.2	628	692	704.8	12.23%	9.8843	7.04545	0.2133
6	250	0.1	0.3	0.6	0.2	628	706	713.6	13.63%	5.8992	12.2978	0.1097
7	100	0.25	0.3	0.6	0.2	628	720	724.2	15.32%	2.3875	8.81631	0.1519
8	250	0.25	0.3	0.6	0.2	628	712	721	14.81%	6.2849	24.3396	0.3594
9	100	0.1	0.3	0.4	0.4	628	709	713.6	13.63%	3.8471	6.99214	0.034
10	250	0.1	0.3	0.4	0.4	628	711	714.4	13.76%	3.0496	12.166	0.1149
11	100	0.25	0.3	0.4	0.4	628	709	718.8	14.46%	5.8052	8.57157	0.0695
12	250	0.25	0.3	0.4	0.4	628	723	729.4	16.15%	5.0299	22.7222	0.0992
13	100	0.1	0.3	0.6	0.4	628	708	718.2	14.36%	5.8907	6.92167	0.045
14	250	0.1	0.3	0.6	0.4	628	711	717.8	14.30%	5.8052	12.0665	0.1067
15	100	0.25	0.3	0.6	0.4	628	709	717.2	14.20%	5.4955	8.65154	0.0347
16	250	0.25	0.3	0.6	0.4	628	709	723.4	15.19%	9.2087	23.2785	0.1108
17	100	0.1	0.5	0.4	0.2	628	703	717.4	14.24%	8.0808	6.89044	0.029
18	250	0.1	0.5	0.4	0.2	628	714	720.2	14.68%	4.6583	11.9745	0.0724
19	100	0.25	0.5	0.4	0.2	628	709	713.4	13.60%	3.8471	8.50983	0.0479
20	250	0.25	0.5	0.4	0.2	628	723	728.8	16.05%	5.4037	22.7457	0.4113
21	100	0.1	0.5	0.6	0.2	628	709	712.6	13.47%	3.9115	6.88999	0.0534
22	250	0.1	0.5	0.6	0.2	628	709	712	13.38%	1.7321	12.0735	0.0805
23	100	0.25	0.5	0.6	0.2	628	717	723.2	15.16%	3.7014	8.63373	0.0522
24	250	0.25	0.5	0.6	0.2	628	715	719.8	14.62%	3.3466	23.5845	0.1752
25	100	0.1	0.5	0.4	0.4	628	716	718.2	14.36%	1.3038	7.01409	0.1739
26	250	0.1	0.5	0.4	0.4	628	713	722.6	15.06%	5.5045	12.2154	0.311
27	100	0.25	0.5	0.4	0.4	628	696	711	13.22%	9.7468	8.53885	0.084
28	250	0.25	0.5	0.4	0.4	628	717	725.6	15.54%	6.8044	22.6709	0.1683
29	100	0.1	0.5	0.6	0.4	628	715	718.4	14.39%	2.3022	6.86133	0.0164
30	250	0.1	0.5	0.6	0.4	628	723	724.8	15.41%	1.6432	11.9983	0.0556
31	100	0.25	0.5	0.6	0.4	628	708	714.6	13.79%	5.3666	8.7092	0.0694
32	250	0.25	0.5	0.6	0.4	628	720	726.25	15.64%	4.272	23.5169	0.099

Para el caso de la instancia actual, se demuestra también que la metaheurística basada en el *ICA* para el *HFS* con *buffers* limitados tamaño dos obtiene secuencias con *makespan* superiores a las de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado. (ver anexo E). Se determina a su vez que el factor más influyente en la variable respuesta para la instancia actual es el número de países y el menos influyente es la interacción entre el % de imperialistas y el  $\xi$  (ver anexo F).

Se destaca el rendimiento en tiempo que tuvo la metaheurística para esta instancia dando respuestas siempre debajo de los 25 segundos contra los 10800 segundo tomados por el método exacto.

**Tabla 10.** Resultados obtenidos para la instancia i7j3k3-3.

#	CP	%Imp	% $\xi$	%Cr	%Rev	SC [ut]	MO [ut]	Med [ut]	%Dif	SD [ut]	Med-t [s]	SD-t [s]
1	100	0.1	0.3	0.4	0.2	633	703	707.8	11.82%	3.6332	6.92212	0.1117
2	250	0.1	0.3	0.4	0.2	633	707	713.8	12.76%	3.9623	11.8689	0.1215
3	100	0.25	0.3	0.4	0.2	633	710	715.2	12.99%	3.0332	8.5891	0.192
4	250	0.25	0.3	0.4	0.2	633	714	717.4	13.33%	2.6077	22.8002	0.2694
5	100	0.1	0.3	0.6	0.2	633	708	713	12.64%	4.6904	7.19422	0.0397
6	250	0.1	0.3	0.6	0.2	633	708	716.4	13.18%	4.7749	12.7472	0.4353
7	100	0.25	0.3	0.6	0.2	633	709	713.2	12.67%	2.49	9.30362	0.1226
8	250	0.25	0.3	0.6	0.2	633	698	715.2	12.99%	11.009	24.9593	0.1717
9	100	0.1	0.3	0.4	0.4	633	701	707.8	11.82%	4.8166	7.44625	0.0537
10	250	0.1	0.3	0.4	0.4	633	699	718.2	13.46%	10.78	12.8874	0.103
11	100	0.25	0.3	0.4	0.4	633	716	721.2	13.93%	3.2711	9.22661	0.0543
12	250	0.25	0.3	0.4	0.4	633	727	732.6	15.73%	4.2778	24.421	0.1768
13	100	0.1	0.3	0.6	0.4	633	711	712.4	12.54%	1.6733	7.44175	0.0456
14	250	0.1	0.3	0.6	0.4	633	700	705	11.37%	6	13.0956	0.5867
15	100	0.25	0.3	0.6	0.4	633	704	716	13.11%	7.2457	9.28186	0.0291
16	250	0.25	0.3	0.6	0.4	633	698	721.4	13.97%	13.903	24.589	0.1206
17	100	0.1	0.5	0.4	0.2	633	697	706.8	11.66%	6.0166	6.99592	0.1736
18	250	0.1	0.5	0.4	0.2	633	718	719.6	13.68%	1.1402	12.1033	0.1143
19	100	0.25	0.5	0.4	0.2	633	701	705.8	11.50%	3.8341	8.57642	0.0384
20	250	0.25	0.5	0.4	0.2	633	712	723.6	14.31%	6.8411	23.0897	0.0805
21	100	0.1	0.5	0.6	0.2	633	710	712.6	12.58%	2.4083	6.92003	0.0318
22	250	0.1	0.5	0.6	0.2	633	700	708.6	11.94%	5.9414	12.1046	0.1133
23	100	0.25	0.5	0.6	0.2	633	702	707	11.69%	3.937	8.81918	0.1655
24	250	0.25	0.5	0.6	0.2	633	702	711	12.32%	5.099	24.088	0.3605
25	100	0.1	0.5	0.4	0.4	633	696	705	11.37%	6.245	7.0201	0.0886
26	250	0.1	0.5	0.4	0.4	633	695	710.4	12.23%	8.9051	12.2652	0.3264
27	100	0.25	0.5	0.4	0.4	633	690	713.8	12.76%	13.442	8.57967	0.0317
28	250	0.25	0.5	0.4	0.4	633	713	721.6	14.00%	6.4653	24.8786	0.139
29	100	0.1	0.5	0.6	0.4	633	675	684.4	8.12%	10.188	7.54859	0.0538
30	250	0.1	0.5	0.6	0.4	633	710	711.2	12.35%	1.3038	14.0607	1.0795
31	100	0.25	0.5	0.6	0.4	633	696	703.6	11.15%	4.7749	10.7156	0.1886
32	250	0.25	0.5	0.6	0.4	633	704	722.8	14.19%	10.569	29.2638	0.287

Para el caso de la instancia i7j3k3-3 (7 trabajos, 3 máquinas por estación, 3 etapas de procesamiento, tamaño de *buffer* 3) de la tabla 10, la metaheurística no logró igual ni superar la solución conocida en ninguna combinación de niveles, siendo la más próxima las 675 u.t. obtenidas para la combinación 29.

La calidad promedio de las respuestas obtenidas por la metaheurística tuvo una mejora frente a la instancia inmediatamente anterior con porcentajes de diferencia entre el 8% mínimo y máximo el 16% frente a la solución conocida. La combinación de niveles identificada con el número 29 fue la que menor diferencia presentó distanciándose de la arrojada por *GAMS* en un 8,12% para una instancia en la que la respuesta obtenida con el método exacto tenía *gap* relativo del 10,5887% (ver tabla 1).

También se determina que la metaheurística basada en el *ICA* para el *HFS* con *buffers* limitados tamaño tres obtiene secuencias con *makespan* generalmente superiores a las de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado (ver anexo E). Para la instancia en cuestión, se determina también que el factor que mayor influencia tiene sobre la variable respuesta es el % de imperialistas, mientras el que menos influye es la interacción entre el % de imperialistas y el % de cruce (ver anexo F).

Con respecto al tiempo de respuesta de la metaheurística los resultados siguen siendo excelentes, pues la misma continúa arrojando respuesta siempre debajo de los 30 s. de tardanza frente a los 10800 s. del método exacto.

**Tabla 11.** Resultados obtenidos para la instancia i7j2k5-1.

#	CP	%Imp	% $\xi$	%Cr	%Rev	SC [ut]	MO [ut]	Med [ut]	%Dif	SD [ut]	Med-t [s]	SD-t [s]
1	100	0,1	0,3	0,4	0,2	1268	1196	1201,2	-5,27%	5,4037	11,8758	0,6563
2	250	0,1	0,3	0,4	0,2	1268	1201	1241,8	-2,07%	22,939	21,9929	1,2809
3	100	0,25	0,3	0,4	0,2	1268	1190	1211,6	-4,45%	12,582	15,90466	1,3041
4	250	0,25	0,3	0,4	0,2	1268	1234	1243,8	-1,91%	7,6616	43,54224	3,065
5	100	0,1	0,3	0,6	0,2	1268	1205	1211,6	-4,45%	4,4497	12,87014	0,8458
6	250	0,1	0,3	0,6	0,2	1268	1232	1237,6	-2,40%	3,7815	23,00748	1,494
7	100	0,25	0,3	0,6	0,2	1268	1215	1218,2	-3,93%	2,5884	15,67916	0,859
8	250	0,25	0,3	0,6	0,2	1268	1240	1245,2	-1,80%	4,2071	45,74406	2,7019
9	100	0,1	0,3	0,4	0,4	1268	1216	1218,8	-3,88%	1,9235	12,59836	0,8715
10	250	0,1	0,3	0,4	0,4	1268	1215	1232	-2,84%	12,629	23,83498	1,5246
11	100	0,25	0,3	0,4	0,4	1268	1222	1228,8	-3,09%	8,0436	16,11348	1,1421
12	250	0,25	0,3	0,4	0,4	1268	1246	1255,2	-1,01%	12,795	45,87234	2,9033
13	100	0,1	0,3	0,6	0,4	1268	1207	1210,2	-4,56%	2,7749	12,63132	0,8715
14	250	0,1	0,3	0,6	0,4	1268	1227	1241,8	-2,07%	8,3487	24,45092	1,6867
15	100	0,25	0,3	0,6	0,4	1268	1212	1223,2	-3,53%	9,654	15,68512	0,8081
16	250	0,25	0,3	0,6	0,4	1268	1234	1247,8	-1,59%	9,8843	45,06428	2,8216
17	100	0,1	0,5	0,4	0,2	1268	1208	1211,8	-4,43%	3,4205	12,04872	0,7646
18	250	0,1	0,5	0,4	0,2	1268	1229	1230,8	-2,93%	1,7889	23,3333	1,626
19	100	0,25	0,5	0,4	0,2	1268	1229	1231,2	-2,90%	2,2804	15,80274	0,909
20	250	0,25	0,5	0,4	0,2	1268	1222	1238	-2,37%	9,2466	44,98472	2,9715
21	100	0,1	0,5	0,6	0,2	1268	1207	1211,4	-4,46%	3,5071	12,72646	0,8795
22	250	0,1	0,5	0,6	0,2	1268	1223	1233,8	-2,70%	7,0143	24,41854	0,891
23	100	0,25	0,5	0,6	0,2	1268	1213	1214,8	-4,20%	1,4832	15,47292	0,9074
24	250	0,25	0,5	0,6	0,2	1268	1237	1242,8	-1,99%	5,6745	43,9149	2,3996
25	100	0,1	0,5	0,4	0,4	1268	1208	1212	-4,42%	4,6368	12,50094	0,8573
26	250	0,1	0,5	0,4	0,4	1268	1218	1228,8	-3,09%	9,6799	23,6214	1,6379
27	100	0,25	0,5	0,4	0,4	1268	1234	1246,2	-1,72%	7,6942	15,36696	0,8969
28	250	0,25	0,5	0,4	0,4	1268	1250	1266,8	-0,09%	16,208	43,55686	2,3295
29	100	0,1	0,5	0,6	0,4	1268	1205	1208,4	-4,70%	3,0496	12,68802	0,8722
30	250	0,1	0,5	0,6	0,4	1268	1238	1241,8	-2,07%	4,9699	25,40034	1,3696
31	100	0,25	0,5	0,6	0,4	1268	1230	1237,4	-2,41%	5,1284	16,3342	1,1764
32	250	0,25	0,5	0,6	0,4	1268	1224	1248,8	-1,51%	18,833	45,90306	2,3925

Como se puede observar en la tabla 11, para la instancia i7j2k5-1 (7 trabajos, 2 máquinas por estación, 5 etapas de procesamiento, tamaño de *buffer* 1) la

metaheurística basada en el *ICA* encuentra para todas las instancias evaluadas respuestas mejores a la arrojada por *GAMS*, siendo la mejor observada la de la combinación de niveles 4 con 1190 u.t.

También en promedio, la metaheurística presenta siempre respuestas mejores a la conocida, obteniendo un porcentaje de mejora entre el 0% y el 7% destacándose sobre las demás la primera combinación de niveles que tiene un progreso frente a la solución exacta del 6,27%.

Para el caso de la instancia actual, se encuentra el importante resultado que la metaheurística basada en el *ICA* para el *HFS* con *buffers* tamaño uno obtiene secuencias con *makespan* prácticamente iguales a las de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado (ver anexo E). También se determina que el factor más influyente en la variable respuesta es el número de países, por el contrario, el menos influyente es la interacción entre el número de países y el % de imperialistas (ver anexo F).

También vale la pena destacar que la metaheurística obtiene respuestas de buena calidad en tiempos en promedio inferiores a 46 s. para todas las posibles combinaciones de tratamientos frente a los 10800 s del método exacto.

Para la instancia *i7j2k5-2* (7 trabajos, 2 máquinas por estación, 5 etapas de procesamiento, tamaño de *buffer* 2) presentada en la tabla 12 a continuación, se observa que no se obtuvo ninguna respuesta igual o mejor a la arrojada por *GAMS*, siendo la más cercana 1168 u.t. conseguida mediante la combinación de niveles 27. La caída en la calidad de las respuestas para la presente instancia frente a la anterior se puede deber a la mejora en la respuesta dada por *GAMS* que pasó de tener un *gap* relativo del 43,9846% para la instancia anterior a un 30,1702% para la presente (ver tabla 1).

A pesar de esto, el algoritmo obtuvo respuestas con porcentajes de diferencia inferiores al 13% con respecto a la solución conocida, destacándose la combinación de niveles de tratamientos 25 con una diferencia de 8,69%.

**Tabla 12.** Resultados obtenidos para la instancia i7j2k5-2.

#	CP	%Imp	%x	%Cr	%Rev	SC	MO [ut]	Med [ut]	%Dif	SD [ut]	Med-t [s]	SD-t [s]
1	100	0,1	0,3	0,4	0,2	1086	1191	1200,4	10,53%	6,3875	12,41038	0,8974
2	250	0,1	0,3	0,4	0,2	1086	1208	1218	12,15%	5,9582	22,89834	1,3422
3	100	0,25	0,3	0,4	0,2	1086	1197	1201,6	10,64%	4,1593	15,22342	0,9785
4	250	0,25	0,3	0,4	0,2	1086	1221	1243,8	14,53%	16,285	44,34454	3,1119
5	100	0,1	0,3	0,6	0,2	1086	1192	1195,4	10,07%	3,9749	12,13734	0,7448
6	250	0,1	0,3	0,6	0,2	1086	1206	1223,6	12,67%	10,784	23,6323	1,7546
7	100	0,25	0,3	0,6	0,2	1086	1198	1209,4	11,36%	7,8294	16,3623	1,0159
8	250	0,25	0,3	0,6	0,2	1086	1204	1230,4	13,30%	15,469	45,09726	2,5225
9	100	0,1	0,3	0,4	0,4	1086	1185	1196,2	10,15%	6,9065	12,3781	0,8718
10	250	0,1	0,3	0,4	0,4	1086	1193	1201	10,59%	6,1237	23,26234	1,3929
11	100	0,25	0,3	0,4	0,4	1086	1193	1205,2	10,98%	6,8337	16,3313	0,831
12	250	0,25	0,3	0,4	0,4	1086	1219	1225	12,80%	8,124	46,44908	2,6811
13	100	0,1	0,3	0,6	0,4	1086	1182	1190,6	9,63%	8,0187	12,58404	0,9156
14	250	0,1	0,3	0,6	0,4	1086	1174	1190,2	9,59%	12,814	22,91504	0,1812
15	100	0,25	0,3	0,6	0,4	1086	1176	1196,6	10,18%	12,054	16,52464	0,869
16	250	0,25	0,3	0,6	0,4	1086	1216	1220	12,34%	3,7417	45,8221	2,3343
17	100	0,1	0,5	0,4	0,2	1086	1183	1189,4	9,52%	6,0249	12,3585	0,8208
18	250	0,1	0,5	0,4	0,2	1086	1185	1201,8	10,66%	13,255	22,84702	1,517
19	100	0,25	0,5	0,4	0,2	1086	1200	1205,4	10,99%	3,2094	14,93626	0,0636
20	250	0,25	0,5	0,4	0,2	1086	1212	1225,2	12,82%	9,6281	45,1891	2,4286
21	100	0,1	0,5	0,6	0,2	1086	1181	1186,8	9,28%	6,4187	12,7365	0,9222
22	250	0,1	0,5	0,6	0,2	1086	1190	1204	10,87%	14,265	22,57846	0,9619
23	100	0,25	0,5	0,6	0,2	1086	1189	1194,2	9,96%	4,8166	15,7475	1,0124
24	250	0,25	0,5	0,6	0,2	1086	1229	1231,4	13,39%	1,6733	46,39242	3,162
25	100	0,1	0,5	0,4	0,4	1086	1169	1180,4	8,69%	8,6197	12,13072	0,7655
26	250	0,1	0,5	0,4	0,4	1086	1196	1213,2	11,71%	9,985	23,27042	1,3011
27	100	0,25	0,5	0,4	0,4	1086	1168	1184	9,02%	9,6177	15,39976	0,8024
28	250	0,25	0,5	0,4	0,4	1086	1193	1196,4	10,17%	2,6077	44,45404	2,2839
29	100	0,1	0,5	0,6	0,4	1086	1191	1201	10,59%	5,6569	12,6284	0,8975
30	250	0,1	0,5	0,6	0,4	1086	1202	1211,6	11,57%	6,5038	23,3055	1,0842
31	100	0,25	0,5	0,6	0,4	1086	1196	1199,2	10,42%	3,6332	15,7422	0,7097
32	250	0,25	0,5	0,6	0,4	1086	1219	1227,2	13,00%	12,091	45,63906	2,5317

En la presente instancia y tal como era de esperarse con base en la anterior, se determina que la metaheurística basada en el *ICA* para el *HFS* con *buffers* limitados de tamaño dos obtiene secuencias con *makespan* de la misma calidad a las de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado (ver anexo E). A su vez, se determina que el factor más influyente en la variable respuesta es el número de países, y el factor interacción entre el % de cruce y % de revolución es el menos influyente (ver anexo F).

Finalmente, con respecto a los tiempos de obtención de respuesta, de nuevo ninguno supera los 47 s. en promedio para todas las combinaciones posibles que superan los 10800 s. tomados por *GAMS* para dar solución.

Con respecto a la instancia presentada en la tabla 13, es decir, la instancia *i7j3k5-1* (7 trabajos, 3 máquinas por estación, 5 etapas de procesamiento, tamaño de *buffer* 1) la metaheurística no logró encontrar respuesta igual o mejor a la solución conocida siendo la más cercana a este valor la presentada en la combinación 5 con 1024 u.t.

**Tabla 13.** Resultados obtenidos para la instancia *i7j3k5-1*.

#	CP	%Imp	% $\xi$	%Cr	%Rev	SC	MO [ut]	Med [ut]	%Dif	SD [ut]	Med-t [s]	SD-t [s]
1	100	0,1	0,3	0,4	0,2	937	1033	1045	11,53%	8,86	12,2138	0,6213
2	250	0,1	0,3	0,4	0,2	937	1051	1055,8	12,68%	2,8636	20,29362	1,5532
3	100	0,25	0,3	0,4	0,2	937	1062	1067,8	13,96%	4,6043	14,9538	1,0044
4	250	0,25	0,3	0,4	0,2	937	1072	1078	15,05%	4,062	37,54994	2,7662
5	100	0,1	0,3	0,6	0,2	937	1024	1050	12,06%	14,95	12,194	0,6557
6	250	0,1	0,3	0,6	0,2	937	1072	1076,8	14,92%	3,5637	19,64228	0,5575
7	100	0,25	0,3	0,6	0,2	937	1048	1052,6	12,34%	4,5056	15,15142	1,0769
8	250	0,25	0,3	0,6	0,2	937	1064	1076,6	14,90%	11,127	38,92394	3,2007
9	100	0,1	0,3	0,4	0,4	937	1075	1077,4	14,98%	1,8166	12,38146	0,9555
10	250	0,1	0,3	0,4	0,4	937	1071	1078	15,05%	4	20,45232	1,4942
11	100	0,25	0,3	0,4	0,4	937	1068	1072,4	14,45%	5,8566	14,74852	1,1025
12	250	0,25	0,3	0,4	0,4	937	1064	1086,4	15,94%	13,012	35,12716	2,0749
13	100	0,1	0,3	0,6	0,4	937	1055	1056,6	12,76%	1,5166	12,7494	0,9064
14	250	0,1	0,3	0,6	0,4	937	1070	1076,4	14,88%	4,9295	20,14546	1,1877
15	100	0,25	0,3	0,6	0,4	937	1077	1078,2	15,07%	1,3038	14,48214	0,7839
16	250	0,25	0,3	0,6	0,4	937	1091	1097,8	17,16%	5,5408	36,8791	2,6672
17	100	0,1	0,5	0,4	0,2	937	1042	1049,8	12,04%	6,6106	12,08908	0,7469
18	250	0,1	0,5	0,4	0,2	937	1054	1058,6	12,98%	3,2094	20,80832	1,2892
19	100	0,25	0,5	0,4	0,2	937	1052	1057,6	12,87%	5,3666	14,49104	0,8962
20	250	0,25	0,5	0,4	0,2	937	1060	1065,6	13,72%	4,827	36,9824	1,9022
21	100	0,1	0,5	0,6	0,2	937	1039	1041,2	11,12%	1,9235	12,4823	1,0222
22	250	0,1	0,5	0,6	0,2	937	1048	1055,4	12,64%	6,3482	20,02574	1,2379
23	100	0,25	0,5	0,6	0,2	937	1052	1055	12,59%	3,7417	14,70718	0,8408
24	250	0,25	0,5	0,6	0,2	937	1070	1073,2	14,54%	3,5637	38,43448	2,5995
25	100	0,1	0,5	0,4	0,4	937	1070	1073	14,51%	2,4495	12,34216	0,6932
26	250	0,1	0,5	0,4	0,4	937	1068	1078,4	15,09%	5,8566	19,76776	1,5547
27	100	0,25	0,5	0,4	0,4	937	1070	1074,4	14,66%	2,9665	14,2323	0,781
28	250	0,25	0,5	0,4	0,4	937	1098	1101,6	17,57%	3,0496	35,93148	2,7218
29	100	0,1	0,5	0,6	0,4	937	1064	1066	13,77%	2,3452	12,73592	0,8427
30	250	0,1	0,5	0,6	0,4	937	1066	1077,4	14,98%	7,733	20,45574	1,3946
31	100	0,25	0,5	0,6	0,4	937	1070	1074	14,62%	3,6742	14,50066	0,926
32	250	0,25	0,5	0,6	0,4	937	1087	1094	16,76%	4,8477	37,77972	2,7171

En promedio la metaheurística encuentra respuestas entre el 11% y 18% de porcentaje de diferencia. La que mejor desempeño tuvo fue la combinación de niveles de tratamientos 21 con una diferencia promedio del 11,12%. Tal como

sucede para el caso de tres etapas de procesamiento, el incremento en la diferencia se puede deber al manejo de tres máquinas y sus consecuencias explicadas anteriormente.

También, se determina para la actual instancia que la metaheurística basada en el *ICA* para el *HFS* con *buffers* limitados tamaño uno obtiene secuencias con *makespan* generalmente superiores a las de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado (ver anexo E). Por su parte, se determina que el factor más influyente en la variable respuesta para la presente instancia es el %  $\xi$  mientras el factor menos influyente es el % de cruce (ver anexo F).

Por último, los tiempos promedio de obtención de respuestas para esta instancia son excelentes, encontrando respuesta siempre debajo de los 39 s. en promedio para cualquier combinación de niveles versus los 10800 s. del método exacto.

**Tabla 14.** Resultados obtenidos para la instancia i7j3k5-2.

#	CP	%Imp	% $\xi$	%Cr	%Rev	SC	MO [ut]	Med [ut]	%Dif	SD [ut]	Med-t [s]	SD-t [s]
1	100	0,1	0,3	0,4	0,2	925	1041	1044,8	12,95%	2,49	12,0242	0,8353
2	250	0,1	0,3	0,4	0,2	925	1035	1050,2	13,54%	9,9599	20,22276	1,1863
3	100	0,25	0,3	0,4	0,2	925	1034	1046	13,08%	8,9722	14,57068	1,0235
4	250	0,25	0,3	0,4	0,2	925	1034	1048,6	13,36%	10,761	38,2914	2,5388
5	100	0,1	0,3	0,6	0,2	925	1026	1034	11,78%	7,6485	12,44382	0,9328
6	250	0,1	0,3	0,6	0,2	925	1023	1041,2	12,56%	11,256	21,05278	1,1439
7	100	0,25	0,3	0,6	0,2	925	1031	1044,4	12,91%	11,48	16,27726	1,1581
8	250	0,25	0,3	0,6	0,2	925	1039	1044,2	12,89%	4,9699	40,1575	2,5913
9	100	0,1	0,3	0,4	0,4	925	1033	1039,2	12,35%	4,9699	12,47432	1,0419
10	250	0,1	0,3	0,4	0,4	925	1041	1046,4	13,12%	3,7148	20,01726	1,5528
11	100	0,25	0,3	0,4	0,4	925	1039	1045,8	13,06%	5,5408	14,4518	0,84
12	250	0,25	0,3	0,4	0,4	925	1044	1055,6	14,12%	7,7006	35,2688	2,4098
13	100	0,1	0,3	0,6	0,4	925	1012	1028,2	11,16%	9,3113	12,4327	0,8662
14	250	0,1	0,3	0,6	0,4	925	1049	1054	13,95%	4,7958	19,28846	1,2889
15	100	0,25	0,3	0,6	0,4	925	1030	1038,8	12,30%	5,8052	15,02388	1,1803
16	250	0,25	0,3	0,6	0,4	925	1043	1055,4	14,10%	11,546	37,31746	2,946
17	100	0,1	0,5	0,4	0,2	925	1044	1046,6	13,15%	2,3022	12,35606	0,8443
18	250	0,1	0,5	0,4	0,2	925	1031	1040,4	12,48%	6,8044	20,3148	1,3412
19	100	0,25	0,5	0,4	0,2	925	1044	1052,4	13,77%	5,1769	15,59544	1,0822
20	250	0,25	0,5	0,4	0,2	925	1052	1057,4	14,31%	3,5777	39,04314	2,8656
21	100	0,1	0,5	0,6	0,2	925	1032	1038,2	12,24%	5,9749	12,55564	0,9093
22	250	0,1	0,5	0,6	0,2	925	1043	1051,2	13,64%	5,6745	20,06312	1,4886
23	100	0,25	0,5	0,6	0,2	925	1039	1041,4	12,58%	2,881	15,33878	1,0989
24	250	0,25	0,5	0,6	0,2	925	1060	1063,6	14,98%	3,9115	40,39434	3,0363
25	100	0,1	0,5	0,4	0,4	925	1031	1036,2	12,02%	4,3243	12,71924	0,888
26	250	0,1	0,5	0,4	0,4	925	1058	1061,6	14,77%	2,881	19,10932	0,9229
27	100	0,25	0,5	0,4	0,4	925	1042	1050,4	13,56%	4,98	14,45172	0,9475
28	250	0,25	0,5	0,4	0,4	925	1042	1047,2	13,21%	5,8907	35,38472	2,4582
29	100	0,1	0,5	0,6	0,4	925	1027	1037,8	12,19%	8,6429	12,35478	1
30	250	0,1	0,5	0,6	0,4	925	1050	1053,4	13,88%	3,5071	19,43204	1,2318
31	100	0,25	0,5	0,6	0,4	925	1043	1052,4	13,77%	6,3875	15,33076	1,0561
32	250	0,25	0,5	0,6	0,4	925	1073	1075,6	16,28%	2,6077	36,8175	2,533

En la tabla 14 se muestran los resultados para la instancia i7j3k5-2 (7 trabajos, 3 máquinas por estación, 5 etapas de procesamiento, tamaño de *buffer* 2) donde se puede evidenciar que la metaheurística no encuentra para ninguna combinación de niveles de tratamientos una respuesta mejor a la obtenida mediante el método exacto. La más cercana es la presentada como mejor solución para la combinación 13 con 1012 u.t.

Al igual que para la instancia anterior, las respuestas en promedio encontradas son de menor calidad que para instancias con dos máquinas por estación. Todas las combinaciones presentan un porcentaje de diferencia entre el 11% y el 17% que son bastante idénticos a los dados para la instancia anterior, destacándose sobre las demás la combinación 13 con una distancia con la solución conocida del 11,16%.

El incremento en el porcentaje de diferencia de las respuestas se puede deber tanto al manejo de tres máquinas (explicado en la instancia anterior) como también a la buena calidad de la respuesta obtenida por el método exacto para la instancia actual que presenta un *gap* relativo del 23,4133% (ver tabla 1).

Para el caso de la instancia actual, se demuestra también que la metaheurística basada en el *ICA* para el *HFS* con *buffers* limitados tamaño dos obtiene secuencias con *makespan* algo superiores a las de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado. (ver anexo E). Se determina a su vez que el factor más influyente en la variable respuesta para la instancia actual es el número de países, por otra parte, el menos influyente es la interacción entre el número de países y el % de imperialistas (ver anexo F).

Se destaca el rendimiento en tiempo que tuvo la metaheurística para esta instancia dando respuestas siempre debajo de los 41 segundos (comparándose con los 10800 segundos tomados por el método exacto).

Para el caso de la instancia i7j3k5-3 (7 trabajos, 3 máquinas por estación, 5 etapas de procesamiento, tamaño de *buffer* 3) mostrado en la tabla 15, la metaheurística no logra igual ni superar la solución conocida en ninguna combinación de niveles, siendo la más próxima a esto las 1035 u.t. obtenidas para la combinación 2.

**Tabla 15.** Resultados obtenidos para la instancia i7j3k5-3.

#	CP	%Imp	% $\xi$	%Cr	%Rev	SC	MO [ut]	Med [ut]	%Dif	SD [ut]	Med-t [s]	SD-t [s]
1	100	0,1	0,3	0,4	0,2	973	1038	1052,2	8,14%	8,0125	12,39966	0,8844
2	250	0,1	0,3	0,4	0,2	973	1035	1045,8	7,48%	7,5299	20,3192	1,6715
3	100	0,25	0,3	0,4	0,2	973	1058	1062,6	9,21%	5,4589	15,1805	1,1763
4	250	0,25	0,3	0,4	0,2	973	1065	1070	9,97%	3,3166	37,04066	2,1635
5	100	0,1	0,3	0,6	0,2	973	1038	1046,4	7,54%	5,7706	11,74134	0,101
6	250	0,1	0,3	0,6	0,2	973	1057	1063,4	9,29%	5,5045	25,19054	8,0233
7	100	0,25	0,3	0,6	0,2	973	1050	1053,8	8,30%	2,7749	15,68936	1,1336
8	250	0,25	0,3	0,6	0,2	973	1066	1069	9,87%	3,7417	39,4201	3,1628
9	100	0,1	0,3	0,4	0,4	973	1044	1049,8	7,89%	5,7184	12,2768	0,907
10	250	0,1	0,3	0,4	0,4	973	1054	1057,2	8,65%	2,5884	20,9956	1,2112
11	100	0,25	0,3	0,4	0,4	973	1069	1072	10,17%	2,2361	14,85616	0,9444
12	250	0,25	0,3	0,4	0,4	973	1073	1085,4	11,55%	12,422	39,19164	3,2126
13	100	0,1	0,3	0,6	0,4	973	1039	1045,6	7,46%	7,3007	12,25718	0,8679
14	250	0,1	0,3	0,6	0,4	973	1065	1069	9,87%	5,0498	19,44532	1,3739
15	100	0,25	0,3	0,6	0,4	973	1048	1051,6	8,08%	4,0988	14,502	0,8484
16	250	0,25	0,3	0,6	0,4	973	1067	1080	11,00%	7,3824	37,62932	2,0743
17	100	0,1	0,5	0,4	0,2	973	1054	1057,8	8,72%	3,2711	11,6459	0,0605
18	250	0,1	0,5	0,4	0,2	973	1066	1070,4	10,01%	3,8471	19,91972	1,3064
19	100	0,25	0,5	0,4	0,2	973	1057	1077	10,69%	11,979	15,37834	1,1555
20	250	0,25	0,5	0,4	0,2	973	1081	1085,8	11,59%	2,8636	37,27172	2,1852
21	100	0,1	0,5	0,6	0,2	973	1036	1049,2	7,83%	8,1056	12,70598	1,0561
22	250	0,1	0,5	0,6	0,2	973	1068	1072	10,17%	3,3912	20,9268	1,21
23	100	0,25	0,5	0,6	0,2	973	1037	1045,6	7,46%	6,229	15,01586	1,0458
24	250	0,25	0,5	0,6	0,2	973	1065	1070,4	10,01%	4,4497	39,65428	2,5614
25	100	0,1	0,5	0,4	0,4	973	1052	1056,2	8,55%	4,0866	12,9713	0,8212
26	250	0,1	0,5	0,4	0,4	973	1063	1067,6	9,72%	2,9665	19,92284	1,3692
27	100	0,25	0,5	0,4	0,4	973	1074	1079	10,89%	3,3912	14,38844	1,0057
28	250	0,25	0,5	0,4	0,4	973	1069	1076,6	10,65%	5,1284	37,2329	2,3208
29	100	0,1	0,5	0,6	0,4	973	1046	1067,8	9,74%	12,538	12,71188	0,87
30	250	0,1	0,5	0,6	0,4	973	1061	1076,2	10,61%	10,849	20,07094	1,3413
31	100	0,25	0,5	0,6	0,4	973	1041	1054,6	8,39%	9,1815	14,92388	1,1489
32	250	0,25	0,5	0,6	0,4	973	1071	1074	10,38%	2,9155	38,17884	2,6124

La calidad promedio de las respuestas obtenidas por la metaheurística tuvo una mejora frente a la instancia inmediatamente anterior con un porcentaje de diferencia entre el 7% mínimo y máximo el 12% frente a la solución conocida. Las combinaciones de niveles con el número 13 y 23 fueron las que menor diferencia presentaron distanciándose de la arrojada por *GAMS* en un 7,46%.

También se determina que la metaheurística basada en el *ICA* para el *HFS* con *buffers* limitados tamaño tres obtiene secuencias con *makespan* un poco

superiores a las de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado (ver anexo E). Para la instancia en cuestión, se determina también que el factor que mayor influencia tiene sobre la variable respuesta es el número de países, y el factor menos influyente es la interacción entre el % de cruce y el % de revolución (ver anexo F).

Con respecto al tiempo de respuesta de la metaheurística los resultados siguen siendo excelentes, pues la misma continúa arrojando respuesta siempre debajo de los 40 s. de tardanza contra los 10800 segundos tomados por *GAMS*.

**Tabla 16.** Resultados obtenidos para la instancia i9j2k3-1.

#	CP	%Imp	% $\xi$	%Cr	%Rev	SC [ut]	MO [ut]	Med [ut]	%Dif	SD [ut]	Med-t [s]	SD-t [s]
1	100	0.1	0.3	0.4	0.2	1014	1038	1044.4	2.998%	5.5045	8.68209	0.0537
2	250	0.1	0.3	0.4	0.2	1014	1037	1047	3.254%	8.3666	16.6396	0.1091
3	100	0.25	0.3	0.4	0.2	1014	1063	1065.8	5.108%	2.2804	11.2465	0.2853
4	250	0.25	0.3	0.4	0.2	1014	1060	1066.6	5.187%	5.2249	32.3154	0.1183
5	100	0.1	0.3	0.6	0.2	1014	1033	1047.6	3.314%	8.6487	8.77532	0.0633
6	250	0.1	0.3	0.6	0.2	1014	1056	1062.2	4.753%	4.6043	17.0013	0.1032
7	100	0.25	0.3	0.6	0.2	1014	1041	1049.6	3.511%	6.269	11.2878	0.091
8	250	0.25	0.3	0.6	0.2	1014	1054	1062	4.734%	7.6158	36.353	1.7324
9	100	0.1	0.3	0.4	0.4	1014	1028	1047	3.254%	11.895	9.75127	0.0957
10	250	0.1	0.3	0.4	0.4	1014	1051	1055.8	4.122%	4.0866	19.4557	0.048
11	100	0.25	0.3	0.4	0.4	1014	1051	1054.2	3.964%	2.9496	11.829	0.5457
12	250	0.25	0.3	0.4	0.4	1014	1067	1075	6.016%	6.2048	33.0214	0.8785
13	100	0.1	0.3	0.6	0.4	1014	1044	1048.4	3.393%	3.6469	8.51623	0.0493
14	250	0.1	0.3	0.6	0.4	1014	1048	1052.8	3.826%	3.1145	18.3354	2.409
15	100	0.25	0.3	0.6	0.4	1014	1038	1047	3.254%	5.7446	14.4818	0.1349
16	250	0.25	0.3	0.6	0.4	1014	1061	1068.2	5.345%	5.4955	43.7679	0.9979
17	100	0.1	0.5	0.4	0.2	1014	1020	1038	2.367%	11.683	10.813	0.0683
18	250	0.1	0.5	0.4	0.2	1014	1050	1052.2	3.767%	1.7889	20.3409	0.3225
19	100	0.25	0.5	0.4	0.2	1014	1032	1052.6	3.807%	11.632	13.6543	0.0924
20	250	0.25	0.5	0.4	0.2	1014	1052	1063.2	4.852%	6.7602	40.9559	1.5848
21	100	0.1	0.5	0.6	0.2	1014	1022	1043	2.860%	13.472	10.8456	0.2167
22	250	0.1	0.5	0.6	0.2	1014	1037	1047	3.254%	7.5166	21.2249	0.4928
23	100	0.25	0.5	0.6	0.2	1014	1044	1052.8	3.826%	5.5408	13.3869	0.3549
24	250	0.25	0.5	0.6	0.2	1014	1036	1051	3.649%	8.5147	39.7482	0.5128
25	100	0.1	0.5	0.4	0.4	1014	1043	1045.6	3.116%	2.0736	10.4533	0.0779
26	250	0.1	0.5	0.4	0.4	1014	1044	1048.6	3.412%	2.881	20.983	0.4965
27	100	0.25	0.5	0.4	0.4	1014	1045	1055.4	4.083%	7.5033	13.5871	0.1818
28	250	0.25	0.5	0.4	0.4	1014	1055	1074.4	5.957%	14.064	39.2883	1.5789
29	100	0.1	0.5	0.6	0.4	1014	1036	1043.6	2.919%	4.5056	10.0848	0.3373
30	250	0.1	0.5	0.6	0.4	1014	1036	1049.4	3.491%	8.2037	21.1747	1.1965
31	100	0.25	0.5	0.6	0.4	1014	1054	1064.8	5.010%	6.4576	13.9155	0.7479
32	250	0.25	0.5	0.6	0.4	1014	1066	1075	6.016%	6.7454	39.6368	0.642

En la tabla 16 se presentan los resultados de la instancia i9j2k3-1 (9 trabajos, 2 máquinas por estación, 3 etapas de procesamiento, tamaño de *buffer* 1). Para este caso, la metaheurística obtuvo muy buenos resultados, encontrando como mejor

solución las 1020 u.t. obtenidas en la combinación de niveles 17 y que se encuentra solo 6 u.t. encima de la solución encontrada con el método exacto.

Respaldando también los buenos resultados para esta instancia se encuentran los porcentajes de diferencia entre las respuestas obtenidas y la del método exacto que prácticamente no sobrepasan el 6% (solo dos combinaciones están encima del 6% con 6,016%) y de nuevo con la combinación 17 se destaca presentando un *gap* de solo el 2,367%.

Para esta instancia, se demuestra que la metaheurística basada en el *ICA* para el *HFS* con *buffers* limitados tamaño uno obtiene secuencias con *makespan* generalmente superiores a las de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado (ver anexo E). A su vez, se determina que el factor más influyente en la variante en la variable respuesta es el % de imperialistas, por otra parte, el que menos influye es la interacción entre el % de cruce y el % de revolución (ver anexo F).

Comparado con los 10800 segundos que toma el método exacto la metaheurística también destaca obteniendo respuestas de muy alta calidad en menos de 45 segundos para cualquier combinación.

Para el caso de la instancia i9j2k3-2 (9 trabajos, 2 máquinas por estación, 3 etapas de procesamiento, tamaño de *buffer* 2) presentada en la tabla 17, ninguna respuesta pudo igual o reducir el tiempo de la solución conocida, la solución más cercana a lograr esto fue la obtenida como mejor para la combinación 25 con 973 u.t.

El comportamiento promedio de las respuestas generadas por el algoritmo es bueno, pero de menor calidad que para la instancia inmediatamente anterior, presentando porcentajes de diferencia que oscilan entre el 4% y el 9% y siendo la mejor combinación de niveles la identificada con el número 25 con un 4,841% de porcentaje de diferenciación.

Tabla 17. Resultados obtenidos para la instancia i9j2k3-2.

#	CP	%Imp	% $\xi$	%Cr	%Rev	SC [ut]	MO [ut]	Med [ut]	%Dif	SD [ut]	Med-t [s]	SD-t [s]
1	100	0.1	0.3	0.4	0.2	942	997	999.2	6.072%	2.3875	8.93747	0.1056
2	250	0.1	0.3	0.4	0.2	942	1002	1006.4	6.837%	3.3615	17.1778	0.0561
3	100	0.25	0.3	0.4	0.2	942	993	1002.8	6.454%	7.2595	12.3342	0.0279
4	250	0.25	0.3	0.4	0.2	942	1011	1021.4	8.429%	8.961	38.0999	0.1393
5	100	0.1	0.3	0.6	0.2	942	1004	1005.6	6.752%	1.3416	10.179	0.0223
6	250	0.1	0.3	0.6	0.2	942	990	1001.6	6.327%	6.8775	18.8786	0.3375
7	100	0.25	0.3	0.6	0.2	942	996	1005.6	6.752%	5.7706	12.5633	0.2718
8	250	0.25	0.3	0.6	0.2	942	1011	1026.6	8.981%	9.7622	31.5258	0.1718
9	100	0.1	0.3	0.4	0.4	942	982	998.2	5.966%	13.142	8.24242	0.0404
10	250	0.1	0.3	0.4	0.4	942	1007	1013.4	7.580%	4.3932	16.6358	0.1417
11	100	0.25	0.3	0.4	0.4	942	993	1000.2	6.178%	4.1473	10.6832	0.026
12	250	0.25	0.3	0.4	0.4	942	1024	1029.4	9.278%	5.6833	33.9372	3.3728
13	100	0.1	0.3	0.6	0.4	942	992	997	5.839%	3.7417	10.162	0.0258
14	250	0.1	0.3	0.6	0.4	942	996	1003.2	6.497%	5.8907	20.6215	0.0939
15	100	0.25	0.3	0.6	0.4	942	996	1000.8	6.242%	4.3818	15.0785	1.127
16	250	0.25	0.3	0.6	0.4	942	1006	1012	7.431%	5.3385	47.8164	0.8571
17	100	0.1	0.5	0.4	0.2	942	986	995	5.626%	5.3852	11.653	0.1345
18	250	0.1	0.5	0.4	0.2	942	997	1011.6	7.389%	8.9889	23.2299	2.1411
19	100	0.25	0.5	0.4	0.2	942	996	1004.6	6.645%	5.0299	14.889	0.3467
20	250	0.25	0.5	0.4	0.2	942	991	1000.6	6.221%	5.6391	44.5413	1.0448
21	100	0.1	0.5	0.6	0.2	942	998	1001.2	6.285%	2.5884	11.6937	0.2451
22	250	0.1	0.5	0.6	0.2	942	991	997.8	5.924%	6.5345	22.7371	0.5286
23	100	0.25	0.5	0.6	0.2	942	998	1003.6	6.539%	5.0299	15.4541	0.305
24	250	0.25	0.5	0.6	0.2	942	1000	1012.8	7.516%	8.438	45.9078	1.2578
25	100	0.1	0.5	0.4	0.4	942	973	987.6	4.841%	10.807	10.5364	1.1845
26	250	0.1	0.5	0.4	0.4	942	1001	1009.8	7.197%	7.4632	19.5975	0.4803
27	100	0.25	0.5	0.4	0.4	942	1006	1012	7.431%	5.6125	12.38	0.1842
28	250	0.25	0.5	0.4	0.4	942	1003	1013.6	7.601%	7.6354	37.9223	0.9182
29	100	0.1	0.5	0.6	0.4	942	976	990	5.096%	8.2158	9.73348	0.1676
30	250	0.1	0.5	0.6	0.4	942	1003	1007.4	6.943%	4.3359	19.8852	0.1188
31	100	0.25	0.5	0.6	0.4	942	1017	1021.6	8.450%	2.7019	13.0384	0.5416
32	250	0.25	0.5	0.6	0.4	942	1012	1026.2	8.938%	13.554	39.7096	0.4225

La reducción en la calidad de la respuesta puede deberse a la mejora del *makespan* obtenido mediante *GAMS* ya que pasó de tener un *gap* relativo de más del 55% en la instancia anterior a uno de 46,3289% para la instancia actual (ver tabla 1).

Con la presente instancia se demuestra también el importante hecho que la metaheurística basada en el *ICA* para el *HFS* con *buffers* limitados tamaño dos obtiene secuencias con *makespan* algo superiores y para algunos pocos casos iguales a las de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado (ver anexo E).

También, se encuentra que el factor que más influye en la variable respuesta es el % de imperialistas, ahora bien, el factor % de cruce de información entre colonias es el que menos influye (ver anexo F).

En el caso de los tiempos se mantiene la tendencia de la metaheurística y en ese apartado sí hay un comportamiento parecido al de instancia i9j3k2-1 pues se arrojan respuestas en menos de 48 segundos máximo para cualquier combinación de niveles en los tratamientos contra los 10800 tomados por el método exacto.

**Tabla 18.** Resultados obtenidos para la instancia i9j3k3-1.

#	CP	%Imp	% $\xi$	%Cr	%Rev	SC [ut]	MO [ut]	Med [ut]	%Dif	SD [ut]	Med-t [s]	SD-t [s]
1	100	0,1	0,3	0,4	0,2	831	<b>815</b>	846,2	1,829%	17,824	9,587028	0,2466
2	250	0,1	0,3	0,4	0,2	831	844	851,2	2,431%	4,6583	17,50566	0,2358
3	100	0,25	0,3	0,4	0,2	831	<b>808</b>	844,4	1,613%	20,756	11,88508	0,1853
4	250	0,25	0,3	0,4	0,2	831	867	877,2	5,560%	6,6858	33,66484	0,455
5	100	0,1	0,3	0,6	0,2	831	851	855,2	2,912%	3,2711	9,62502	0,1239
6	250	0,1	0,3	0,6	0,2	831	859	865,2	4,116%	4,2661	20,46174	4,51
7	100	0,25	0,3	0,6	0,2	831	<b>820</b>	846,4	1,853%	15,947	12,19908	0,0731
8	250	0,25	0,3	0,6	0,2	831	864	875,2	5,319%	8,5849	34,44608	0,6701
9	100	0,1	0,3	0,4	0,4	831	<b>831</b>	842,6	1,396%	8,2037	9,56314	0,1749
10	250	0,1	0,3	0,4	0,4	831	<b>815</b>	853,2	2,671%	21,753	17,75296	0,3896
11	100	0,25	0,3	0,4	0,4	831	857	872,4	4,982%	9,2358	13,09646	0,3354
12	250	0,25	0,3	0,4	0,4	831	874	881,2	6,041%	8,9554	36,04848	0,131
13	100	0,1	0,3	0,6	0,4	831	<b>820</b>	838	0,842%	12,329	10,43202	0,1329
14	250	0,1	0,3	0,6	0,4	831	853	862,4	3,779%	5,4129	19,0015	0,2958
15	100	0,25	0,3	0,6	0,4	831	866	879,8	5,872%	8,3785	12,99774	0,2703
16	250	0,25	0,3	0,6	0,4	831	859	869	4,573%	10	36,12342	3,3781
17	100	0,1	0,5	0,4	0,2	831	<b>801</b>	852	2,527%	31,209	9,467634	0,7012
18	250	0,1	0,5	0,4	0,2	831	853	861,2	3,634%	7,2595	19,02598	1,3273
19	100	0,25	0,5	0,4	0,2	831	844	854,4	2,816%	8,4439	12,1985	1,149
20	250	0,25	0,5	0,4	0,2	831	852	864,8	4,067%	11,819	34,44214	3,5114
21	100	0,1	0,5	0,6	0,2	831	833	847,6	1,998%	12,402	9,901126	0,9445
22	250	0,1	0,5	0,6	0,2	831	846	860,6	3,562%	10,668	18,55604	1,6827
23	100	0,25	0,5	0,6	0,2	831	838	850,2	2,310%	8,1363	11,89606	0,8289
24	250	0,25	0,5	0,6	0,2	831	874	877,8	5,632%	3,4928	35,27572	3,3093
25	100	0,1	0,5	0,4	0,4	831	837	850,2	2,310%	8,4083	9,48818	0,7882
26	250	0,1	0,5	0,4	0,4	831	869	872,4	4,982%	2,7019	18,82072	1,6878
27	100	0,25	0,5	0,4	0,4	831	849	865,4	4,140%	9,2358	12,3848	1,199
28	250	0,25	0,5	0,4	0,4	831	876	887,6	6,811%	6,9498	37,13996	2,6471
29	100	0,1	0,5	0,6	0,4	831	<b>809</b>	832,6	0,193%	14,153	10,39739	0,5213
30	250	0,1	0,5	0,6	0,4	831	843	852,8	2,623%	6,2209	18,74686	1,2702
31	100	0,25	0,5	0,6	0,4	831	860	864,2	3,995%	3,7014	13,75454	0,2132
32	250	0,25	0,5	0,6	0,4	831	862	871	4,813%	6,9282	36,69816	3,9892

Con respecto a la instancia presentada en la tabla 18, es decir, la instancia i9j3k3-1 (9 trabajos, 3 máquinas por estación, 3 etapas de procesamiento, tamaño de *buffer* 1) la metaheurística logra encontrar respuesta igual o mejor que la solución conocida en ocho diferentes instancias, siendo la mejor observada las 801 u.t. de la combinación de niveles 17.

Sin embargo, la metaheurística no logra reducir en promedio los resultados de GAMS para ninguna de las instancias, pero presenta porcentajes de diferencia

cercanos al 0% y no superiores al 7%, siendo la más cercana la instancia 29 que presenta un *gap* de apenas el 0,193% con la solución conocida.

También, se determina para la actual instancia que la metaheurística basada en el *ICA* para el *HFS* con *buffers* limitados tamaño uno obtiene secuencias con *makespan* superiores a las de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado (ver anexo E). Por su parte, se determina que el factor más influyente en la variable respuesta para la presente instancia es el número de países y el que menos influye es la interacción entre el número de países y el % de imperialistas (ver anexo F).

Por último, los tiempos promedio de obtención de respuestas para esta instancia son excelentes, encontrando respuesta siempre debajo de los 38 s. en promedio para cualquier combinación de niveles comparados con los 10800 de *GAMS*.

**Tabla 19.** Resultados obtenidos para la instancia i9j3k3-2.

#	CP	%Imp	%ξ	%Cr	%Rev	SC [ut]	MO [ut]	Med [ut]	%Dif	SD [ut]	Med-t [s]	SD-t [s]
1	100	0,1	0,3	0,4	0,2	709	799	807,4	13,88%	7,0922	9,584066	0,7598
2	250	0,1	0,3	0,4	0,2	709	817	821,4	15,85%	6,5803	18,54226	1,605
3	100	0,25	0,3	0,4	0,2	709	838	846,8	19,44%	8,9275	12,36484	1,0389
4	250	0,25	0,3	0,4	0,2	709	852	864,4	21,92%	7,6354	35,71376	2,8053
5	100	0,1	0,3	0,6	0,2	709	840	847	19,46%	4,3012	10,234566	0,8911
6	250	0,1	0,3	0,6	0,2	709	819	832,6	17,43%	8,1425	18,44678	1,6369
7	100	0,25	0,3	0,6	0,2	709	824	837,4	18,11%	10,383	12,49848	1,198
8	250	0,25	0,3	0,6	0,2	709	849	861,4	21,50%	7,893	37,76856	2,6407
9	100	0,1	0,3	0,4	0,4	709	834	841,4	18,67%	5,3198	9,92214	0,8383
10	250	0,1	0,3	0,4	0,4	709	840	847,6	19,55%	7,7653	18,96934	1,7165
11	100	0,25	0,3	0,4	0,4	709	844	846,4	19,38%	1,9494	11,6381	0,1638
12	250	0,25	0,3	0,4	0,4	709	863	866,6	22,23%	3,8471	35,06872	3,7578
13	100	0,1	0,3	0,6	0,4	709	794	805,6	13,62%	10,31	9,926222	0,7161
14	250	0,1	0,3	0,6	0,4	709	859	863,8	21,83%	3,8987	17,86142	1,1689
15	100	0,25	0,3	0,6	0,4	709	810	814	14,81%	2,9155	11,95854	0,0253
16	250	0,25	0,3	0,6	0,4	709	858	864,8	21,97%	6,6106	36,11906	3,3318
17	100	0,1	0,5	0,4	0,2	709	832	833,8	17,60%	1,4832	10,96894	0,2458
18	250	0,1	0,5	0,4	0,2	709	783	824,8	16,33%	23,658	16,82204	0,182
19	100	0,25	0,5	0,4	0,2	709	773	820	15,66%	27,713	11,8973	0,1977
20	250	0,25	0,5	0,4	0,2	709	860	863,6	21,81%	2,881	35,21332	2,5842
21	100	0,1	0,5	0,6	0,2	709	807	823,6	16,16%	10,383	9,836056	0,8505
22	250	0,1	0,5	0,6	0,2	709	819	829,6	17,01%	6,3087	17,50744	1,372
23	100	0,25	0,5	0,6	0,2	709	841	848,4	19,66%	6,3482	12,8755	0,9527
24	250	0,25	0,5	0,6	0,2	709	856	860,6	21,38%	2,9665	35,53706	3,4388
25	100	0,1	0,5	0,4	0,4	709	812	815,8	15,06%	2,9496	9,616318	0,6853
26	250	0,1	0,5	0,4	0,4	709	850	854,2	20,48%	4,3818	18,374356	1,509
27	100	0,25	0,5	0,4	0,4	709	827	832	17,35%	4,7958	11,75742	0,1786
28	250	0,25	0,5	0,4	0,4	709	850	859,6	21,24%	8,5615	35,79452	3,3176
29	100	0,1	0,5	0,6	0,4	709	829	844	19,04%	8,6603	10,139674	0,891
30	250	0,1	0,5	0,6	0,4	709	812	847,2	19,49%	20,897	19,309822	1,4246
31	100	0,25	0,5	0,6	0,4	709	847	853	20,31%	5,0498	11,994366	0,0976
32	250	0,25	0,5	0,6	0,4	709	866	874	23,27%	6,442	34,12764	0,2018

En la tabla 19 se muestran los resultados para la instancia i9j3k3-2 (9 trabajos, 3 máquinas por estación, 3 etapas de procesamiento, tamaño de *buffer* 2) donde se puede evidenciar que la metaheurística no encuentra para ninguna combinación de niveles de tratamientos una respuesta mejor a la obtenida mediante el método exacto. La más cercana es la presentada como mejor solución para la combinación 19 con 773 u.t.

Todas las combinaciones presentan un porcentaje de diferencia entre el 13% y el 24%, destacándose sobre las demás la combinación 13 con una distancia promedio a la respuesta conocida del 13,62%. El incremento en el porcentaje de diferencia de las respuestas con respecto a la instancia anterior se puede deber a la buena calidad de la solución obtenida por el método exacto para la instancia actual que presenta un *gap* relativo del 36,5287% frente a los 50,4056% de la instancia anterior (ver tabla 1).

Para el caso de la instancia actual se demuestra que la metaheurística basada en el *ICA* para el *HFS* con *buffers* limitados tamaño dos obtiene secuencias con *makespan* superiores a las de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado (ver anexo E). Se determina a su vez que el factor más influyente en la variable respuesta para la instancia actual es el número de países mientras el factor interacción entre el número de países y  $\% \xi$  es el menos influyente (ver anexo F).

Se destaca el rendimiento en tiempo que tuvo la metaheurística para esta instancia dando respuestas siempre debajo de los 38 segundos, es decir, muchos menores a los 10800 segundos tomados por el método exacto.

Para el caso de la instancia i9j3k3-3 (9 trabajos, 3 máquinas por estación, 3 etapas de procesamiento, tamaño de *buffer* 3) de la tabla 20, la metaheurística no logra igual ni superar la solución conocida en ninguna combinación de niveles, siendo la más próxima las 796 u.t. obtenidas para la combinación 19.

La calidad promedio de las respuestas obtenidas por la metaheurística se mantuvo con respecto la instancia inmediatamente anterior con porcentajes de diferencia de máximo el 23% frente a la solución conocida. La combinación de niveles identificada con el número 5 fue la que menor porcentaje de diferencia presentó distanciándose de la arrojada por GAMS en un 16,63%.

**Tabla 20.** Resultados obtenidos para la instancia i9j3k3-3.

#	CP	%Imp	% $\xi$	%Cr	%Rev	SC [ut]	MO [ut]	Med [ut]	%Dif	SD [ut]	Med-t [s]	SD-t [s]
1	100	0,1	0,3	0,4	0,2	700	811	830	18,57%	11,358	9,312702	0,0999
2	250	0,1	0,3	0,4	0,2	700	818	836,4	19,49%	11,149	18,06998	1,6116
3	100	0,25	0,3	0,4	0,2	700	825	836,8	19,54%	7,0498	12,18536	0,943
4	250	0,25	0,3	0,4	0,2	700	831	843,8	20,54%	12,696	36,40422	0,4428
5	100	0,1	0,3	0,6	0,2	700	809	816,4	16,63%	7,3348	8,958018	0,1371
6	250	0,1	0,3	0,6	0,2	700	819	834,2	19,17%	12,029	16,95264	0,9846
7	100	0,25	0,3	0,6	0,2	700	819	826,2	18,03%	6,7231	13,3194	0,11
8	250	0,25	0,3	0,6	0,2	700	833	843,8	20,54%	12,153	35,34768	2,1343
9	100	0,1	0,3	0,4	0,4	700	803	820,8	17,26%	11,054	10,14071	0,603
10	250	0,1	0,3	0,4	0,4	700	811	832,4	18,91%	12,3	16,9532	0,347
11	100	0,25	0,3	0,4	0,4	700	827	831,2	18,74%	5,8481	12,58994	1,0882
12	250	0,25	0,3	0,4	0,4	700	838	849	21,29%	9,434	35,14454	2,7906
13	100	0,1	0,3	0,6	0,4	700	820	824,2	17,74%	2,8636	10,030066	0,7032
14	250	0,1	0,3	0,6	0,4	700	822	827,2	18,17%	5,4498	18,33324	1,3539
15	100	0,25	0,3	0,6	0,4	700	831	835,8	19,40%	4,0866	13,38226	0,2096
16	250	0,25	0,3	0,6	0,4	700	841	850,8	21,54%	7,791	35,39086	2,2111
17	100	0,1	0,5	0,4	0,2	700	823	826,8	18,11%	3,8987	9,389604	0,5999
18	250	0,1	0,5	0,4	0,2	700	827	829,4	18,49%	1,8166	17,37812	1,0065
19	100	0,25	0,5	0,4	0,2	700	796	830,8	18,69%	20,909	12,52742	0,9064
20	250	0,25	0,5	0,4	0,2	700	838	841,6	20,23%	4,0988	34,36248	2,4512
21	100	0,1	0,5	0,6	0,2	700	826	828,6	18,37%	2,7019	9,73088	0,7075
22	250	0,1	0,5	0,6	0,2	700	814	819,4	17,06%	5,9414	16,96472	1,0829
23	100	0,25	0,5	0,6	0,2	700	837	840,6	20,09%	3,3615	12,3614	1,0454
24	250	0,25	0,5	0,6	0,2	700	847	855,2	22,17%	8,6429	35,74366	2,531
25	100	0,1	0,5	0,4	0,4	700	817	832	18,86%	9,8234	9,700462	0,8009
26	250	0,1	0,5	0,4	0,4	700	831	850	21,43%	13,784	17,8587	1,0821
27	100	0,25	0,5	0,4	0,4	700	832	836,6	19,51%	4,98	12,31596	0,9698
28	250	0,25	0,5	0,4	0,4	700	821	841	20,14%	11,269	33,78358	2,1358
29	100	0,1	0,5	0,6	0,4	700	822	833,2	19,03%	7,2595	9,773736	0,7045
30	250	0,1	0,5	0,6	0,4	700	844	848,6	21,23%	5,8992	17,54704	1,1254
31	100	0,25	0,5	0,6	0,4	700	848	850,8	21,54%	1,9235	11,80474	0,1915
32	250	0,25	0,5	0,6	0,4	700	836	856,4	22,34%	12,522	35,31142	2,6204

También se determina que la metaheurística basada en el ICA para el HFS con buffers tamaño tres obtiene secuencias con makespan superiores a las de la metaheurística basada en el ICA para el HFS con buffers de tamaño ilimitado (ver anexo E). Para la instancia en cuestión, se determina también que el factor que mayor influencia tiene sobre la variable respuesta es el % de imperialistas y el factor interacción entre el número de países y el % de imperialistas es el que menor influencia tiene (ver anexo F).

Con respecto al tiempo de respuesta de la metaheurística los resultados siguen siendo excelentes, pues la misma continúa arrojando respuesta siempre debajo de los 36 s. de tardanza y también muy por debajo de los 10800 s. del método exacto.

**Tabla 21.** Resultados obtenidos para la instancia i9j2k5-1.

#	CP	%Imp	% $\xi$	%Cr	%Rev	SC [ut]	MO [ut]	Med [ut]	%Dif	SD [ut]	Med-t [s]	SD-t [s]
1	100	0,1	0,3	0,4	0,2		1432	1448,8		11,946	16,77472	0,9165
2	250	0,1	0,3	0,4	0,2		1463	1470,4		7,2319	28,92432	1,3552
3	100	0,25	0,3	0,4	0,2		1438	1451		14,629	19,12748	2,2922
4	250	0,25	0,3	0,4	0,2		1435	1466		21,26	57,19844	5,3581
5	100	0,1	0,3	0,6	0,2		1399	1442,4		25,501	16,3661	1,6663
6	250	0,1	0,3	0,6	0,2		1453	1465,8		8,8148	29,402	2,7977
7	100	0,25	0,3	0,6	0,2		1452	1462,2		12,194	19,79468	1,8349
8	250	0,25	0,3	0,6	0,2		1476	1485,8		7,1554	55,99204	4,3148
9	100	0,1	0,3	0,4	0,4		1444	1450,2		5,6303	15,61532	1,44
10	250	0,1	0,3	0,4	0,4		1453	1462,8		10,402	31,52188	3,2567
11	100	0,25	0,3	0,4	0,4		1450	1477,8		19,955	20,81838	2,0735
12	250	0,25	0,3	0,4	0,4		1486	1504,4		12,178	59,03024	5,4575
13	100	0,1	0,3	0,6	0,4		1443	1450,8		5,5857	15,22652	1,0563
14	250	0,1	0,3	0,6	0,4		1468	1478,8		8,8994	32,39396	3,4657
15	100	0,25	0,3	0,6	0,4		1478	1491,4		11,845	20,37456	2,1876
16	250	0,25	0,3	0,6	0,4		1510	1532		14,142	59,59674	5,3278
17	100	0,1	0,5	0,4	0,2		1418	1426,6		9,8387	15,72584	1,4896
18	250	0,1	0,5	0,4	0,2		1426	1456		18,166	29,78626	2,77
19	100	0,25	0,5	0,4	0,2		1468	1474,6		6,6182	19,68084	1,9757
20	250	0,25	0,5	0,4	0,2		1498	1507,2		7,4297	59,32524	4,5545
21	100	0,1	0,5	0,6	0,2		1449	1461,8		11,52	17,3485	1,7538
22	250	0,1	0,5	0,6	0,2		1450	1464,6		13,74	30,41116	3,7301
23	100	0,25	0,5	0,6	0,2		1424	1452,6		17,213	20,49042	2,3433
24	250	0,25	0,5	0,6	0,2		1459	1463,8		5,6303	63,3645	13,561
25	100	0,1	0,5	0,4	0,4		1443	1452		8,5147	16,75036	2,4946
26	250	0,1	0,5	0,4	0,4		1430	1458,2		21,557	30,24286	3,2707
27	100	0,25	0,5	0,4	0,4		1464	1471,8		6,0992	19,54712	1,777
28	250	0,25	0,5	0,4	0,4		1510	1529,4		16,103	56,82132	5,0553
29	100	0,1	0,5	0,6	0,4		1415	1444		23,42	15,73923	1,5406
30	250	0,1	0,5	0,6	0,4		1423	1442,8		14,446	33,47432	1,7659
31	100	0,25	0,5	0,6	0,4		1432	1456,8		23,637	21,05258	1,3683
32	250	0,25	0,5	0,6	0,4		1508	1524		11,747	63,1801	3,3953

Como se puede observar en la tabla 21, para la instancia i9j2k5-1 (9 trabajos, 2 máquinas por estación, 5 etapas de procesamiento, tamaño de *buffer* 1) la metaheurística basada en el *ICA* encuentra para todas las combinaciones de niveles evaluadas algún tipo de respuesta, mientras *GAMS* no arrojó ninguna dentro de los tiempos límites establecidos. Se destaca como mejor respuesta la observada en la combinación de niveles 17 con 1418 u.t.

No se pueden presentar resultados de diferencia de la metaheurística porque no existe una solución conocida con cuál comparar, sin embargo, las mejores

respuestas en promedio las entrega la combinación de niveles identificada con el número 17 de nuevo con 1426,6 u.t. en parte logrado gracias a obtener el mejor resultado observado.

Para el caso de la instancia i9j2k5-1 actual, se encuentra el destacado resultado que la metaheurística basada en el *ICA* para el *HFS* con *buffers* limitados tamaño uno obtiene secuencias con *makespan* prácticamente iguales a los de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado (ver anexo E). También se determina que el factor más influyente en la variable respuesta es el número de países y el menos influyente es la interacción entre el %  $\xi$  y % de cruce (ver anexo F).

También vale la pena destacar que la metaheurística obtiene respuestas de buena calidad en tiempos en promedio inferiores a 64 s. para todas las posibles combinaciones de tratamientos frente a los 18000 s. del método exacto que no arrojó repuesta.

Para la instancia i9j2k5-2 (9 trabajos, 2 máquinas por estación, 5 etapas de procesamiento, tamaño de *buffer* 2) presentada en la tabla 22, se observa que no se obtuvo ninguna respuesta igual o mejor a la obtenida mediante *GAMS*, siendo la más cercana 1402 u.t. originada mediante la combinación de niveles 23. El decaimiento en la calidad de las respuestas para la presente instancia frente a la anterior se puede deber a que *GAMS* ni siquiera encontró respuesta para la instancia anterior mientras para la actual sí encuentra (ver tabla 1). Además, de que se trata de una instancia muy grande, con muchos trabajos, varias etapas y una gran cantidad de permutaciones existentes para ordenar los trabajos y las máquinas lo cual dificulta un rendimiento óptimo de la metaheurística.

A pesar de esto, el algoritmo obtuvo respuestas con porcentajes de diferencia inferiores al 21% con respecto a la solución conocida, destacando la combinación de niveles de tratamientos 13 con una diferencia del 13%.

En la presente instancia y tal como era de esperarse con base en la anterior, que la metaheurística basada en el *ICA* para el *HFS* con *buffers* limitado tamaño dos obtiene secuencias con *makespan* de idéntica calidad a las de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado (ver anexo E). A su vez, se determina que el factor más influyente en la variable respuesta el número de países, mientras la interacción entre %  $\xi$  y % de cruce es el menos influyente (ver anexo F).

**Tabla 22.** Resultados obtenidos para la instancia i9j2k5-2.

#	CP	% Imp	% $\xi$	% Cr	% Rev	SC [ut]	MO [ut]	Med [ut]	% Dif	SD [ut]	Med-t [s]	SD-t [s]
1	100	0,1	0,3	0,4	0,2	1260	1427	1450,8	15,1%	15,205	16,59234	0,3805
2	250	0,1	0,3	0,4	0,2	1260	1430	1448,4	15,0%	15,534	30,70174	0,6672
3	100	0,25	0,3	0,4	0,2	1260	1425	1449,6	15,0%	16,577	21,39348	0,8438
4	250	0,25	0,3	0,4	0,2	1260	1458	1479,4	17,4%	14,398	60,02948	2,0659
5	100	0,1	0,3	0,6	0,2	1260	1433	1445,6	14,7%	10,04	16,9077	0,4524
6	250	0,1	0,3	0,6	0,2	1260	1469	1478,6	17,3%	8,6487	31,42404	0,8437
7	100	0,25	0,3	0,6	0,2	1260	1454	1462,6	16,1%	6,9857	21,6633	0,473
8	250	0,25	0,3	0,6	0,2	1260	1458	1472,4	16,9%	12,178	61,19746	1,2399
9	100	0,1	0,3	0,4	0,4	1260	1429	1438,4	14,2%	9,2358	16,86456	0,2772
10	250	0,1	0,3	0,4	0,4	1260	1474	1486,4	18,0%	8,0187	32,53366	0,7108
11	100	0,25	0,3	0,4	0,4	1260	1448	1455,8	15,5%	6,14	21,31254	0,4242
12	250	0,25	0,3	0,4	0,4	1260	1502	1512,6	20,0%	7,3348	61,54162	1,4006
13	100	0,1	0,3	0,6	0,4	1260	1406	1424,2	13,0%	12,276	16,95204	0,1252
14	250	0,1	0,3	0,6	0,4	1260	1411	1430,8	13,6%	14,237	33,27004	0,8029
15	100	0,25	0,3	0,6	0,4	1260	1438	1443,4	14,6%	4,6152	22,02632	0,4306
16	250	0,25	0,3	0,6	0,4	1260	1503	1509,8	19,8%	8,438	63,90956	1,6985
17	100	0,1	0,5	0,4	0,2	1260	1422	1432	13,7%	8,7464	16,71536	0,4735
18	250	0,1	0,5	0,4	0,2	1260	1430	1451,4	15,2%	19,932	31,31942	0,5324
19	100	0,25	0,5	0,4	0,2	1260	1445	1453,4	15,3%	9,3968	21,16686	0,491
20	250	0,25	0,5	0,4	0,2	1260	1490	1499,6	19,0%	9,9146	59,91326	0,5975
21	100	0,1	0,5	0,6	0,2	1260	1416	1427	13,3%	8,7464	17,4331	1,2142
22	250	0,1	0,5	0,6	0,2	1260	1436	1447,8	14,9%	10,569	31,9217	0,3309
23	100	0,25	0,5	0,6	0,2	1260	1402	1429,4	13,4%	21,197	21,66362	0,2873
24	250	0,25	0,5	0,6	0,2	1260	1440	1467,2	16,4%	15,77	61,32456	0,5003
25	100	0,1	0,5	0,4	0,4	1260	1428	1435,2	13,9%	7,3959	17,22916	0,2684
26	250	0,1	0,5	0,4	0,4	1260	1466	1474,2	17,0%	9,9348	33,40868	0,8484
27	100	0,25	0,5	0,4	0,4	1260	1445	1465,8	16,3%	14,89	21,61014	0,3665
28	250	0,25	0,5	0,4	0,4	1260	1505	1508,6	19,7%	3,9115	61,42422	0,7334
29	100	0,1	0,5	0,6	0,4	1260	1409	1421	12,8%	11,247	18,03362	0,7825
30	250	0,1	0,5	0,6	0,4	1260	1427	1453,6	15,4%	17,897	33,56086	0,5441
31	100	0,25	0,5	0,6	0,4	1260	1440	1443,8	14,6%	5,3572	21,9943	0,3837
32	250	0,25	0,5	0,6	0,4	1260	1498	1509,4	19,8%	6,8775	64,10198	1,1785

Finalmente, con respecto a los tiempos de obtención de respuesta, de nuevo ninguno supera los 65 s. en promedio para todas las combinaciones posibles frente a los 10800 tomados por el método exacto.

Con respecto a la instancia presentada en la tabla 23, es decir, la instancia i9j3k5-1 (9 trabajos, 3 máquinas por estación, 5 etapas de procesamiento, tamaño de

buffer 1) para la cual GAMS no halló solución, la metaheurística encuentra para las 32 combinaciones de niveles algún tipo de respuesta, siendo la más destaca la presentada en la combinación 3 con 1215 u.t.

Como sucede con la instancia i9j2k5-1 no se puede realizar análisis por porcentaje de diferencia ya que no existe con qué comparar las soluciones promedio de la metaheurística. De igual manera, se destaca la combinación 4 de nuevo con respuestas promedio de 1234,6 incluyendo la mejor respuesta observada.

**Tabla 23.** Resultados obtenidos para la instancia i9j3k5-1.

#	CP	% Imp	% ξ	% Cr	% Rev	SC [ut]	MO [ut]	Med [ut]	% Dif	SD [ut]	Med-t [s]	SD-t [s]
1	100	0,1	0,3	0,4	0,2		1253	1262		6,8191	16,7861	0,7964
2	250	0,1	0,3	0,4	0,2		1247	1273,4		17,43	27,3641	1,6927
3	100	0,25	0,3	0,4	0,2		1215	1234,6		17,416	20,17386	1,2081
4	250	0,25	0,3	0,4	0,2		1248	1263		13,134	51,23548	2,9142
5	100	0,1	0,3	0,6	0,2		1266	1275		7,3824	16,53868	0,8293
6	250	0,1	0,3	0,6	0,2		1249	1268,6		19,667	27,8667	2,1104
7	100	0,25	0,3	0,6	0,2		1246	1259		8,9722	20,19304	1,2071
8	250	0,25	0,3	0,6	0,2		1262	1283		19,429	54,65976	4,8385
9	100	0,1	0,3	0,4	0,4		1253	1265,6		10,55	16,49656	1,1272
10	250	0,1	0,3	0,4	0,4		1235	1260,4		18,16	28,39902	0,4041
11	100	0,25	0,3	0,4	0,4		1266	1277,6		6,8044	21,0988	0,4304
12	250	0,25	0,3	0,4	0,4		1294	1309,4		14,223	52,06244	0,8916
13	100	0,1	0,3	0,6	0,4		1257	1262,8		7,8549	16,77586	0,4791
14	250	0,1	0,3	0,6	0,4		1268	1279,6		10,502	30,05834	0,6362
15	100	0,25	0,3	0,6	0,4		1233	1276,6		25,383	20,91304	0,2511
16	250	0,25	0,3	0,6	0,4		1305	1319,8		12,112	53,65286	2,2711
17	100	0,1	0,5	0,4	0,2		1240	1248,2		7,6289	17,50678	0,4799
18	250	0,1	0,5	0,4	0,2		1260	1267,8		6,3403	28,05572	0,3488
19	100	0,25	0,5	0,4	0,2		1267	1276		7,3824	20,66174	0,2322
20	250	0,25	0,5	0,4	0,2		1265	1291,8		15,723	52,11776	0,729
21	100	0,1	0,5	0,6	0,2		1238	1247		9,9247	16,90732	0,1255
22	250	0,1	0,5	0,6	0,2		1255	1274,6		12,442	28,47252	0,4372
23	100	0,25	0,5	0,6	0,2		1262	1270,4		9,8641	21,00232	0,3929
24	250	0,25	0,5	0,6	0,2		1272	1283,6		10,334	54,39378	1,0422
25	100	0,1	0,5	0,4	0,4		1231	1259,4		17,097	17,00112	0,25
26	250	0,1	0,5	0,4	0,4		1241	1269,2		16,498	28,3563	0,4854
27	100	0,25	0,5	0,4	0,4		1299	1308,4		5,8992	20,4228	0,4169
28	250	0,25	0,5	0,4	0,4		1326	1344,8		12,755	50,86218	1,3968
29	100	0,1	0,5	0,6	0,4		1261	1273,8		8,8713	17,32092	0,0765
30	250	0,1	0,5	0,6	0,4		1261	1278,4		14,926	29,08562	0,6559
31	100	0,25	0,5	0,6	0,4		1285	1288,8		2,9496	20,60614	0,2355
32	250	0,25	0,5	0,6	0,4		1306	1325,2		15,531	53,26162	0,7652

También, se determina para la actual instancia el interesante resultado que la metaheurística basada en el ICA para el HFS con buffers limitados tamaño uno obtiene secuencias con makespan un poco superiores a los de la metaheurística basada en el ICA para el HFS con buffers de tamaño ilimitado (ver anexo E). Por su parte, se determina que el factor más influyente en la variable respuesta para la

presente instancia es el % de revolución, a su vez, el factor menos influyente está compuesto por la interacción entre %  $\xi$  y % de cruce (ver anexo F).

Por último, los tiempos promedio de obtención de respuestas para esta instancia son excelentes, encontrando respuesta siempre debajo de los 55 s. en promedio para cualquier combinación de niveles contra los 18000 tomados por GAMS para no arrojar respuesta.

**Tabla 24.** Resultados obtenidos para la instancia i9j3k5-2.

#	CP	% Imp	% $\xi$	% Cr	% Rev	SC [ut]	MO [ut]	Med [ut]	% Dif	SD [ut]	Med-t [s]	SD-t [s]
1	100	0,1	0,3	0,4	0,2	1016	1247	1253,6	23,39%	7,2319	17,27536	0,2115
2	250	0,1	0,3	0,4	0,2	1016	1258	1262,8	24,29%	4,4385	28,57616	0,5173
3	100	0,25	0,3	0,4	0,2	1016	1242	1251,6	23,19%	10,065	20,57074	0,2116
4	250	0,25	0,3	0,4	0,2	1016	1247	1259,2	23,94%	10,872	52,48062	0,7408
5	100	0,1	0,3	0,6	0,2	1016	1168	1231	21,16%	36,104	17,26834	0,1819
6	250	0,1	0,3	0,6	0,2	1016	1228	1246	22,64%	10,124	30,13042	1,9688
7	100	0,25	0,3	0,6	0,2	1016	1245	1250,2	23,05%	5,2631	21,32066	0,3002
8	250	0,25	0,3	0,6	0,2	1016	1243	1252,8	23,31%	6,5345	54,71396	0,5075
9	100	0,1	0,3	0,4	0,4	1016	1224	1240	22,05%	14,37	17,28414	0,5417
10	250	0,1	0,3	0,4	0,4	1016	1245	1256,2	23,64%	9,2304	28,27604	0,7892
11	100	0,25	0,3	0,4	0,4	1016	1251	1261,6	24,17%	9,8641	20,80092	0,3064
12	250	0,25	0,3	0,4	0,4	1016	1224	1262,2	24,23%	21,948	52,31484	0,6908
13	100	0,1	0,3	0,6	0,4	1016	1186	1229	20,96%	24,218	17,58496	0,3304
14	250	0,1	0,3	0,6	0,4	1016	1237	1253,4	23,37%	11,866	29,16618	0,7453
15	100	0,25	0,3	0,6	0,4	1016	1242	1248,6	22,89%	5,7706	20,88698	0,2222
16	250	0,25	0,3	0,6	0,4	1016	1279	1298,2	27,78%	13,142	53,7462	1,728
17	100	0,1	0,5	0,4	0,2	1016	1208	1232	21,26%	19,609	16,66758	0,7333
18	250	0,1	0,5	0,4	0,2	1016	1252	1257,6	23,78%	4,8785	27,55727	1,0161
19	100	0,25	0,5	0,4	0,2	1016	1203	1240,8	22,13%	22,264	20,12052	0,861
20	250	0,25	0,5	0,4	0,2	1016	1291	1306	28,54%	13,038	52,19565	3,522
21	100	0,1	0,5	0,6	0,2	1016	1221	1241,4	22,19%	13,722	16,46064	0,7905
22	250	0,1	0,5	0,6	0,2	1016	1254	1260,2	24,04%	5,4037	28,5473	1,452
23	100	0,25	0,5	0,6	0,2	1016	1218	1231,2	21,18%	14,653	20,62488	1,2219
24	250	0,25	0,5	0,6	0,2	1016	1233	1249	22,93%	12,59	52,1558	3,877
25	100	0,1	0,5	0,4	0,4	1016	1232	1242,2	22,26%	9,0111	16,43462	0,7636
26	250	0,1	0,5	0,4	0,4	1016	1259	1268,8	24,88%	7,6289	27,59688	1,0119
27	100	0,25	0,5	0,4	0,4	1016	1227	1242,8	22,32%	10,281	20,0745	0,8144
28	250	0,25	0,5	0,4	0,4	1016	1300	1309,4	28,88%	7,733	50,29564	3,2858
29	100	0,1	0,5	0,6	0,4	1016	1225	1242,2	22,26%	13,33	16,93508	0,8373
30	250	0,1	0,5	0,6	0,4	1016	1240	1251,6	23,19%	10,526	28,59082	1,7295
31	100	0,25	0,5	0,6	0,4	1016	1253	1258,2	23,84%	5,2154	20,55696	1,3157
32	250	0,25	0,5	0,6	0,4	1016	1297	1304,6	28,41%	6,8044	49,42014	0,2707

En la tabla 24, se muestran los resultados para la instancia i9j3k5-2 (9 trabajos, 3 máquinas por estación, 5 etapas de procesamiento, tamaño de *buffer* 2) donde se puede evidenciar que la metaheurística no encuentra para ninguna combinación de niveles de tratamientos una respuesta mejor a la obtenida mediante el método exacto. La más cercana es la presentada como mejor solución para la combinación 5 con 1168 u.t.

Todas las combinaciones presentan porcentaje de diferencia entre el 20% y el 29%, destacándose sobre las demás la combinación 13 con una distancia a la solución conocida del 20,96%. El incremento en el porcentaje de diferencia de las respuestas se puede deber a los mismos hechos que se explicaron para la instancia i9j2k5-2 presentada antes.

Para el caso de la instancia i9j3k5-2 actual, se demuestra también que la metaheurística basada en el *ICA* para el *HFS* con *buffers* limitados tamaño dos obtiene secuencias con *makespan* prácticamente iguales a los de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado (ver anexo E). Se determina a su vez que el factor más influyente en la variable respuesta para la instancia actual es el número de países, mientras tanto, el factor menos influyente la interacción entre % de imperialistas y % de revolución (ver anexo F).

Se destaca el rendimiento en tiempo que tuvo la metaheurística para esta instancia dando respuestas siempre debajo de los 55 segundos frente a los 10800 segundos del método exacto.

Finalizando con el caso de la instancia i9j3k5-3 (9 trabajos, 3 máquinas por estación, 5 etapas de procesamiento, tamaño de *buffer* 3) de la tabla 25, la metaheurística no logra igual ni superar la solución conocida en ninguna combinación de niveles, siendo la más próxima las 1187 u.t. obtenidas para la combinación 11.

La calidad promedio de las respuestas obtenidas por la metaheurística mantuvo la tendencia de la instancia inmediatamente anterior con porcentajes de diferencia entre el 20% mínimo y máximo el 29% frente a la solución conocida. La combinación de niveles identificada con el número 13 fue la que menor diferencia presentó distanciándose de la arrojada por *GAMS* en un 20,22%.

También se determina que la metaheurística basada en el *ICA* para el *HFS* con *buffers* limitados tamaño tres obtiene secuencias con *makespan* idénticas a los de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado (ver anexo E). Para la instancia en cuestión, se determina también que el factor que mayor influencia tiene sobre la variable respuesta es de nuevo el número de países, mientras el factor  $\% \xi$  es el menos influyente en la variante respuesta.

Con respecto al tiempo de respuesta de la metaheurística los resultados siguen siendo excelentes, pues la misma continúa arrojando respuesta siempre debajo de los 53 s. de tardanza contra los 10800 tomados por el método exacto.

**Tabla 25.** Resultados obtenidos para la instancia i9j3k5-3.

#	CP	%Imp	$\% \xi$	%Cr	%Rev	SC [ut]	MO [ut]	Med [ut]	%Dif	SD [ut]	Med-t [s]	SD-t [s]
1	100	0,1	0,3	0,4	0,2	1005	1212	1224,8	21,87%	8,5849	16,41028	0,6464
2	250	0,1	0,3	0,4	0,2	1005	1228	1240,6	23,44%	9,8894	26,9221	0,7317
3	100	0,25	0,3	0,4	0,2	1005	1225	1238	23,18%	8,3066	20,03768	0,3314
4	250	0,25	0,3	0,4	0,2	1005	1253	1261,4	25,51%	7,5366	51,39184	0,721
5	100	0,1	0,3	0,6	0,2	1005	1227	1235,6	22,95%	6,9138	16,2998	0,3627
6	250	0,1	0,3	0,6	0,2	1005	1233	1248,8	24,26%	9,4446	27,9079	0,5217
7	100	0,25	0,3	0,6	0,2	1005	1220	1234	22,79%	9,8234	20,1571	0,5469
8	250	0,25	0,3	0,6	0,2	1005	1220	1251,6	24,54%	22,3	51,69338	1,5593
9	100	0,1	0,3	0,4	0,4	1005	1199	1212,4	20,64%	13,557	16,12386	0,6531
10	250	0,1	0,3	0,4	0,4	1005	1204	1236,8	23,06%	18,873	25,36244	4,3627
11	100	0,25	0,3	0,4	0,4	1005	1187	1221,6	21,55%	20,082	20,07234	0,3607
12	250	0,25	0,3	0,4	0,4	1005	1251	1264,6	25,83%	7,9246	50,81638	0,4842
13	100	0,1	0,3	0,6	0,4	1005	1200	1208,2	20,22%	6,9426	16,52474	0,4961
14	250	0,1	0,3	0,6	0,4	1005	1200	1224,2	21,81%	21,811	27,48334	0,8303
15	100	0,25	0,3	0,6	0,4	1005	1236	1242	23,58%	7,2457	19,81378	0,6977
16	250	0,25	0,3	0,6	0,4	1005	1237	1252,2	24,60%	12,518	50,556	1,5554
17	100	0,1	0,5	0,4	0,2	1005	1201	1217,4	21,13%	9,7108	16,50596	0,4965
18	250	0,1	0,5	0,4	0,2	1005	1221	1235,4	22,93%	11,845	27,15136	0,4191
19	100	0,25	0,5	0,4	0,2	1005	1195	1223,2	21,71%	15,849	19,95594	0,3611
20	250	0,25	0,5	0,4	0,2	1005	1193	1234	22,79%	23,654	51,82994	0,4729
21	100	0,1	0,5	0,6	0,2	1005	1216	1226,4	22,03%	8,0187	16,67198	0,3755
22	250	0,1	0,5	0,6	0,2	1005	1214	1240	23,38%	20,359	28,1805	0,4269
23	100	0,25	0,5	0,6	0,2	1005	1213	1223	21,69%	8,1548	20,52136	0,4265
24	250	0,25	0,5	0,6	0,2	1005	1247	1260,4	25,41%	10,945	52,0694	0,1558
25	100	0,1	0,5	0,4	0,4	1005	1220	1230,6	22,45%	10,237	16,61592	0,3852
26	250	0,1	0,5	0,4	0,4	1005	1239	1250,2	24,40%	9,3648	26,22012	0,3861
27	100	0,25	0,5	0,4	0,4	1005	1228	1231,8	22,57%	4,9699	20,52729	0,3127
28	250	0,25	0,5	0,4	0,4	1005	1284	1294,2	28,78%	6,0166	52,12413	0,5355
29	100	0,1	0,5	0,6	0,4	1005	1209	1219,8	21,37%	7,9812	17,02181	0,4341
30	250	0,1	0,5	0,6	0,4	1005	1221	1242,4	23,62%	19,476	27,94205	0,3182
31	100	0,25	0,5	0,6	0,4	1005	1201	1223,8	21,77%	15,928	20,3952	0,4905
32	250	0,25	0,5	0,6	0,4	1005	1281	1290,4	28,40%	11,393	51,04341	0,2644

Para finalizar el análisis, se presenta a continuación una tabla con el resumen de los resultados obtenidos por el *ICA* y una serie de gráficas también de resumen

para mostrar las tendencias de las soluciones obtenidas, todo se analiza con base en el tamaño del *buffer*.

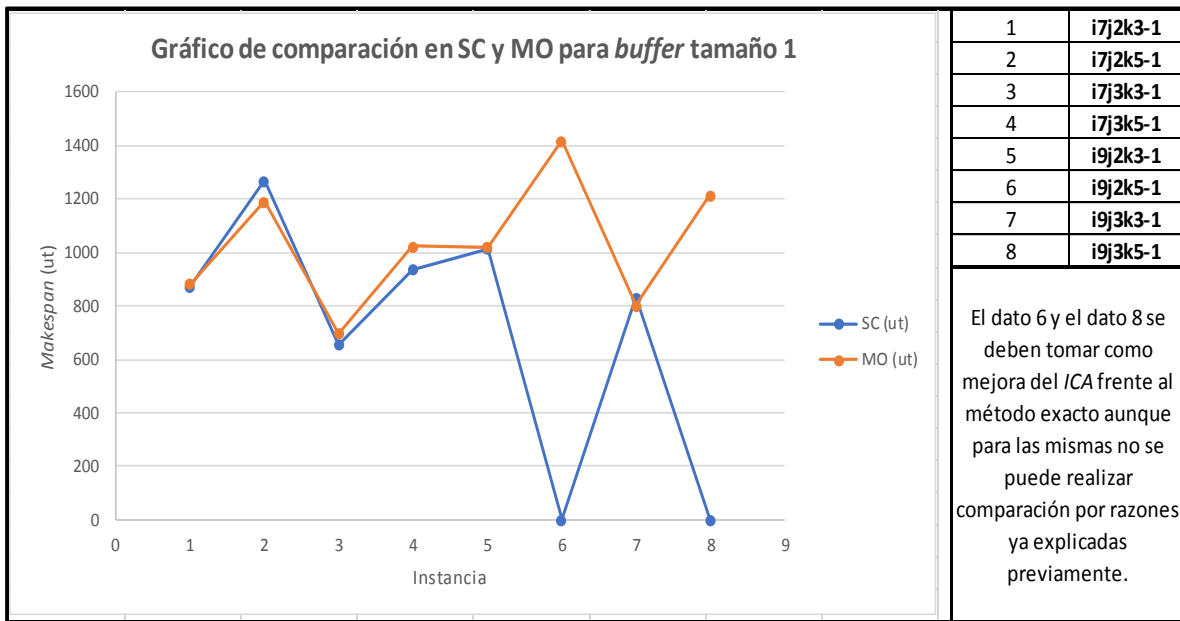
**Tabla 26.** Resumen de los resultados obtenidos por el *ICA* ordenados por tamaño de *buffer*.

Instancia	SC (ut)	MO (ut)	% Dif	Med-t GAMS (s)	Med-t ICA (s)
i7j2k3-1	873	882	2,085%	10800	32
i7j2k5-1	1268	1190	-6,270%	10800	46
i7j3k3-1	657	697	7,915%	10800	25
i7j3k5-1	937	1024	11,120%	10800	39
i9j2k3-1	1014	1020	2,367%	10800	45
i9j2k5-1	---	1418	---	18000	64
i9j3k3-1	831	801	0,193%	10800	38
i9j3k5-1	---	1215	---	18000	55
i7j2k3-2	842	853	4,751%	10800	29
i7j2k5-2	1086	1168	8,690%	10800	47
i7j3k3-2	628	692	12,230%	10800	25
i7j3k5-2	925	1012	11,160%	10800	41
i9j2k3-2	942	973	4,841%	10800	48
i9j2k5-2	1260	1402	13,000%	10800	65
i9j3k3-2	709	773	13,620%	10800	38
i9j3k5-2	1016	1168	20,960%	10800	55
i7j3k3-3	633	675	8,120%	10800	30
i9j3k3-3	700	796	16,630%	10800	36
i7j3k5-3	973	1035	7,460%	10800	40
i9j3k5-3	1005	1187	20,220%	10800	53

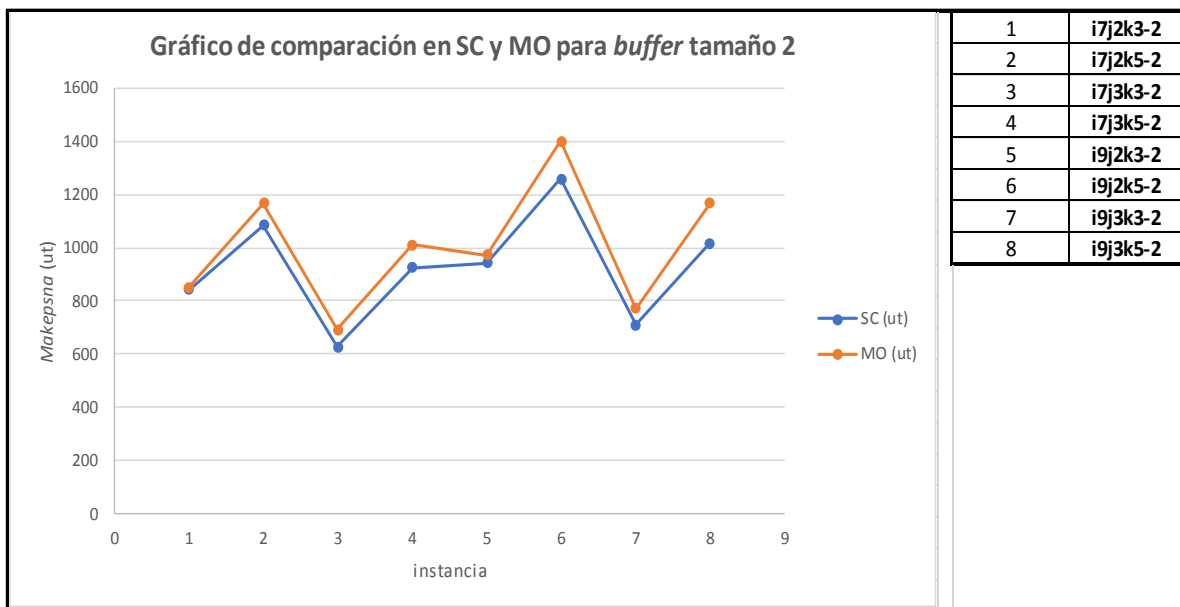
Las instancias resaltadas con verde para la columna MO (ut) son aquellas que obtuvieron mejores resultados que el SC (ut) en alguna de las cinco réplicas.

Mientras tanto, en la columna % Dif. aquellas que tienen un resalto verde oscuro son las que en promedio para las cinco réplicas mejoraron el SC (ut), las de un verde tono medio estuvieron entre el 0% y 5% encima del SC (ut), finalmente las de verde claro están encima de la SC (ut) entre el 5% y el 10%.

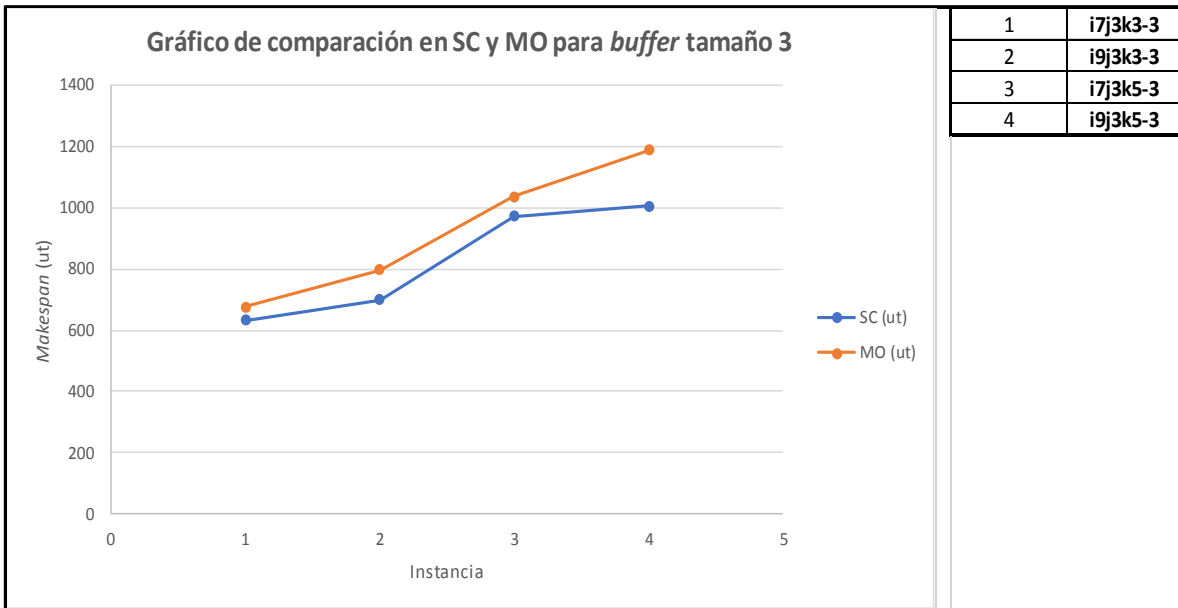
**Gráfica 9.** Gráfica de resumen de resultados obtenidos por el ICA para la mejor solución observada en instancias con tamaño de *buffer* 1.



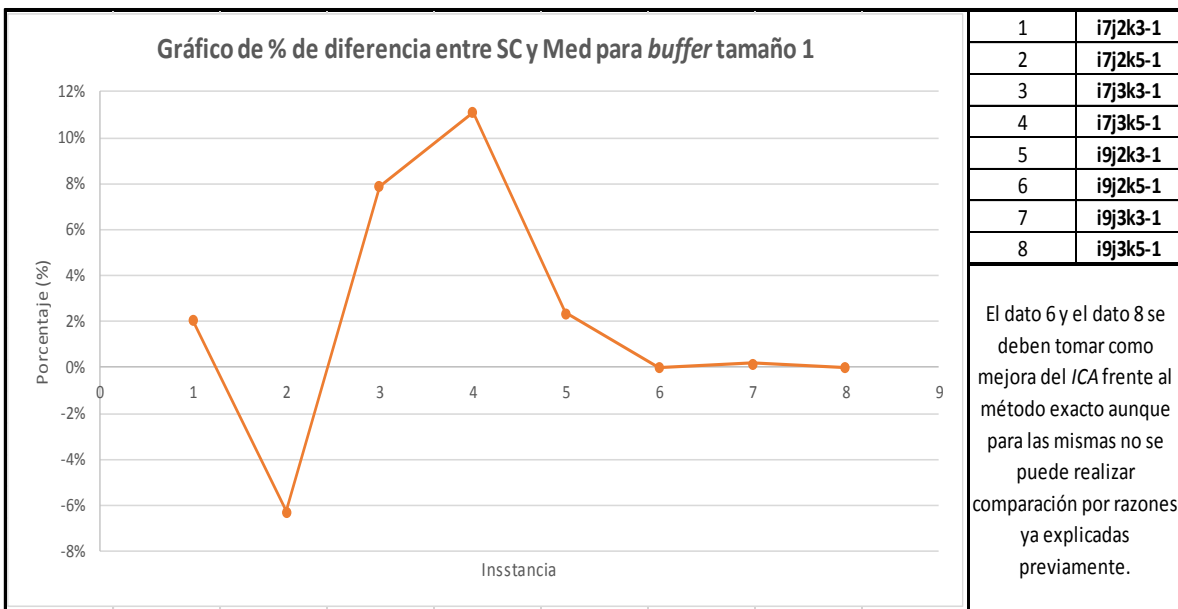
**Gráfica 10.** Gráfica de resumen de resultados obtenidos por el ICA para la mejor solución observada en instancias con tamaño de *buffer* 2.



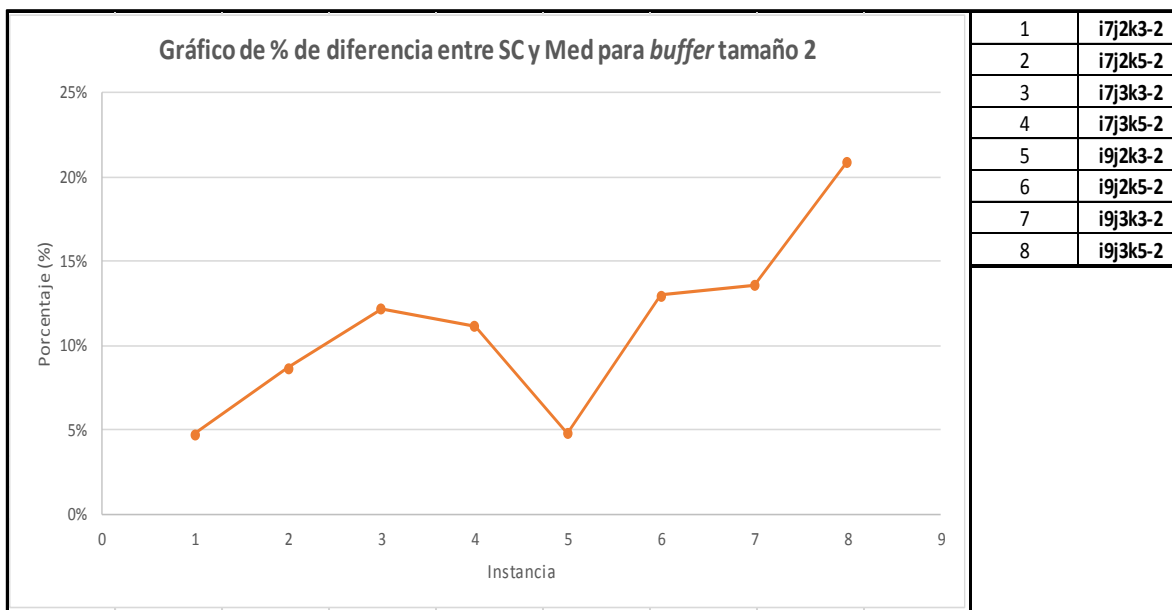
**Gráfica 11.** Gráfica de resumen de resultados obtenidos por el ICA para la mejor solución observada en instancias con tamaño de *buffer* 3.



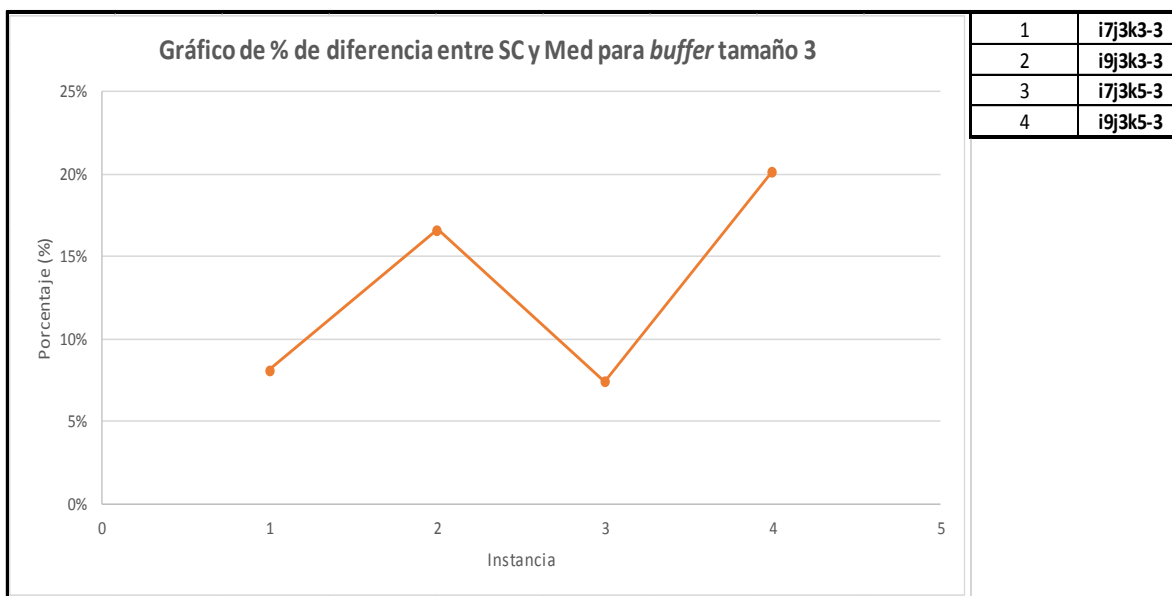
**Gráfica 12.** Gráfica de resumen de porcentaje de diferencia en instancias con tamaño de *buffer* 1.



**Gráfica 13.** Gráfica de resumen de porcentaje de diferencia en instancias con tamaño de *buffer* 2.



**Gráfica 14.** Gráfica de resumen de porcentaje de diferencia en instancias con tamaño de *buffer* 3.



## 7. CONCLUSIONES

- Al utilizar el método exacto, se pudo comprobar la alta complejidad computacional del problema del *HFS* con *buffers* limitados, ya que el software *GAMS* no encuentra para ninguna de las instancias planteadas solución con porcentaje de *gap* relativo igual a cero pasados 10800 segundos para 18 de las 20 instancias y 18000 segundos para las instancias *i9j2k5-1* e *i9j3k5-1* en las cuales no converge en una solución acorde al contexto del problema, todo esto frente a los máximo 65 segundos que toma *Matlab* para arrojar respuesta en todas las instancias.
- Con base en las soluciones arrojadas por el método exacto, se evidencia una relación inversa entre el tamaño del *buffer* y la calidad de la respuesta obtenida, ya que para instancias con igual cantidad de trabajos, máquinas y etapas se obtiene soluciones con un *gap* relativo mayor para aquellas con tamaño de *buffer* pequeño frente a aquellas con tamaño *buffer* más grande, en otras palabras, a menor tamaño de *buffer* mayor porcentaje de *gap* relativo.
- Partiendo de los resultados obtenidos en el diseño experimental fraccionado, se observa que el cambio de nivel en el factor número de países es el que mayor influencia tiene en las respuestas y, en segundo lugar, el porcentaje de imperialistas. Por otra parte, los cambios de nivel en el factor de cruce de información entre las colonias es el que menos influye en el *makespan* de las secuencias.
- A pesar de la complejidad del problema, la metaheurística basada en el algoritmo competitivo imperialista para el problema del *HFS* con *buffers* limitados obtiene buenas respuestas para las instancias planteadas, logrando superar la solución dada por *GAMS* (o arrojando respuesta para casos donde no se obtuvo con el método exacto) en cuatro instancias, todas ellas con tamaño de *buffer* uno (las más complejas) y obteniendo soluciones con un porcentaje de diferencia

menor al 5% con la solución conocida en otras cuatro instancias, y entre el 5% y el 10% de diferencia en otras cuatro, en tiempos mucho más reducidos a los tomados por *GAMS*.

- La metaheurística basada en el algoritmo competitivo imperialista obtiene respuestas de mejor calidad con respecto a las arrojadas por *GAMS* en instancias con tamaño de *buffer* pequeño, es decir, a medida que aumenta el tamaño de *buffer* la calidad de la respuesta disminuye. A su vez, la calidad de la solución también decrece para instancias grandes, en especial cuando se aumenta el número de trabajos.
- Finalmente, y con base en el comparativo entre los resultados obtenidos por la metaheurística basada en el *ICA* para *buffers* con tamaño limitado y para *buffers* con tamaño ilimitado se concluye que la primera metaheurística generalmente tiene respuestas con *makespan* mayores a los de la segunda, el incremento se puede deber en parte a algún impacto de los bloqueos y estancias en el *buffer* que hace que la metaheurística genere secuencias diferentes con *makespan* superiores. También se pudo observar que la metaheurística basada en el *ICA* para *buffers* con tamaño limitado presenta una variabilidad mucho menor en sus respuestas frente a las arrojadas por la misma metaheurística para el caso con *buffers* ilimitados.

## 8. RECOMENDACIONES

- Cambiar el método de generación de la población inicial por otros como *shortest processing time* o *large processing time* y comparar si la metaheurística basada en el *ICA* tiene mejores resultados para el *HFS* con *buffers* limitados que con una población inicial *FIFO*.
- Modificar la metaheurística basada en el *ICA* para que luego de los pasos de asimilación del imperialista por parte de las colonias y revolución de los países colonia compare entre las nuevas colonias y las antiguas y seleccione la mejor y no siempre la nueva, y determinar si este cambio conlleva a mejores respuestas para el caso del *HFS* con *buffers* limitados.
- Variar los niveles de los parámetros de asimilación del imperialista por parte de las colonias, información compartida entre las colonias y revolución de los países colonia tenidos en cuenta para el algoritmo competitivo imperialista en esta investigación y comparar los resultados del *HFS* con *buffers* limitados con esos nuevos niveles con los obtenidos con los actuales y establecer si existe una mejora en la calidad del *makespan*.
- Desarrollar un nuevo método metaheurístico para resolver el problema del *hybrid flow shop* con máquinas paralelas no relacionadas, tiempos de alistamiento dependientes de la secuencia y *buffers* de tamaño limitado y comparar los resultados con los obtenidos por el *ICA* y establecer la eficiencia del mismo.
- Determinar de forma más detallada, si en las instancias planteadas para la actual investigación donde el algoritmo competitivo imperialista para el problema del *HFS* con *buffers* limitados tiene secuencias con un *makespan* mayor que el *ICA* para *HFS* con *buffers* ilimitados el incremento se debe al rendimiento de la metaheurística o al impacto en el *makespan* de la reducción del tamaño de *buffer*.

- Promover la investigación del tema del *flow shop* híbrido con máquinas paralelas no relacionadas y *buffers* de tamaño limitado mediante el planteamiento de nuevos modelos que incluyan nuevas restricciones como tiempos de transporte entre estaciones de procesamiento, tiempos de alistamiento dependientes de la secuencia anticipatorios y procesamiento de trabajos por lotes.

## BIBLIOGRAFÍA

ALMEDER, Christian y HARTL, Richard F. A metaheuristic optimization approach for a real-world stochastic flexible flow shop problem with limited buffer. En: International Journal of Production Economics, 2013, vol. 145, no 1, p. 88-95.

ARANGO-MARÍN, Jaime Antero, GIRALDO-GARCÍA, Jaime Alberto y CASTRILLÓN-GÓMEZ, Omar Danilo. Applying TOC Heuristics to Job Scheduling in a Hybrid Flexible Flow Shop. En: Dyna, 2014, vol. 81, no 186, p. 113-119.

ATASHPAZ-GARGARI, Esmaeil y LUCAS, Caro. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. En Evolutionary computation, 2007. CEC 2007. IEEE Congress on. IEEE, 2007. p. 4661-4667.

ATTAR, S. F., MOHAMMADI, M. y TAVAKKOLI-MOGHADDAM, R. Hybrid flexible flowshop scheduling problem with unrelated parallel machines and limited waiting times. En: The International Journal of Advanced Manufacturing Technology, 2013, vol. 68, no 5-8, p. 1583.

BŁAŻEWICZ, Jacek, et al.. Handbook on scheduling: from theory to applications. Springer Science & Business Media, 2007.

BOZORGIRAD, Mir Abbas y LOGENDRAN, Rasaratnam. A comparison of local search algorithms with population-based algorithms in hybrid flow shop scheduling problems with realistic characteristics. En: The International Journal of Advanced Manufacturing Technolo.

BOZORGIRAD, Mir Abbas y LOGENDRAN, Rasaratnam. Hybrid flowshop scheduling problem with a bi-criteria objective and group technology assumption. En: IIE Annual Conference. Proceedings. Institute of Industrial Engineers-Publisher, 2013. p. 3004.

BOZORGIRAD, Mir Abbas y LOGENDRAN, Rasaratnam. Sequence-dependent group scheduling problem on unrelated-parallel machines. En: Expert Systems with Applications, 2012, vol. 39, p. 9021-9030.

CASTRILLÓN, GIRALDO y SARACHE. Técnicas inteligentes y estocásticas en scheduling. En: Un enfoque en la Prod. y las operaciones, 2010, Unibiblios.

CHASE, R.B., JACOBS, F. R. y AQUILANO, N.j. Administración de Operaciones, Producción y cadena de suministros, Editorial: McGraw Hill, 2009, 12va. Edición, p. prefacio.

CHÁVEZ, Marco Antonio Cruz, BERNAL, Pedro Moreno y ABARCA, Jesús del Carmen Peralta. Aplicación de la teoría de la complejidad en optimización combinatoria. *Inventio*, la génesis de la cultura universitaria en Morelos, 2014, no 20, p. 35.

COSTA, Antonio, CAPPADONNA, Fulvio Antonio y FICHERA, Sergio. A novel genetic algorithm for the hybrid flow shop scheduling with parallel batching and eligibility constraints. En: *The International Journal of Advanced Manufacturing Technology*, 2014, vol. 7.

CUI, Z y GU, X. An improved discrete artificial bee colony algorithm to minimize the makespan on hybrid flow shop problems. En: *Neurocomputing*, 2015, vol. 148, p. 248-259.

DE FATIMA MORAIS, Marcia, et al.. Multicriteria hybrid flow shop scheduling problem: literature review, analysis, and future research. En *Independent Journal of Management & Production*, 2014, vol. 5, no 4, p. 1004-1031.

DE SIQUEIRA, Eduardo Camargo, DE SOUZA, Sergio Ricardo y SOUZA, Marcone Jamilson Freitas. Um algoritmo iterated greedy search aplicado à minimização do makespan no problema flowline híbrido e flexível, 2013, p.44-55.

DEFERSHA, Fantahun Melaku y CHEN, Mingyuan. Mathematical model and parallel genetic algorithm for hybrid flexible flowshop lot streaming problem. En: *The International Journal of Advanced Manufacturing Technology*, 2012, vol. 62, no 1-4, p. 249-265.

DUARTE MUÑOZ, Abraham, PANTRIGO FERNÁNDEZ, J. J. y GALLEGO CARRILLO, M. Metaheurísticas. Madrid: Dykinson, 2007. p 1.

EBRAHIMI, M. et al.. Hybrid flow shop scheduling with sequence dependent family setup time and uncertain due dates. En: *Applied Mathematical Modelling*, 2014, vol. 38, p. 2490-2504.

FATTAHI, P., HOSSEINI, S. M. H., JOLAI, F., TAVAKKOLI-MOGHADDAN, R. A branch and bound algorithm for hybrid flow shop scheduling problema with setup time and assembly operations. En: *Applied Mathematical Modelling* 38. 2014. Páginas: 119-134.

FIGIELSKA, Ewa. A genetic algorithm and a simulated annealing algorithm combined with column generation technique for solving the problem of scheduling in the hybrid flowshop with additional resources. En: *Computers & Industrial Engineering*, 2009, vol. 56.

FIGIELSKA, Ewa. A heuristic for scheduling in a two-stage hybrid flowshop with renewable resources shared among the stages. En: *European Journal of Operational Research*, 2014, vol. 236, no 2, p. 433-444.

FIGIELSKA, Ewa. A new heuristic for scheduling the two-stage flowshop with additional resources. En: *Computers & Industrial Engineering*, 2008, vol. 54, no 4, p. 750-763.

FIGIELSKA, Ewa. Heuristic algorithms for preemptive scheduling in a two-stage hybrid flowshop with additional renewable resources at each stage. En: *Computers & Industrial Engineering*, 2010, vol. 59, no 4, p. 509-51.

FISHBACK, Paul E. *Linear and nonlinear programming with Maple: an interactive, applications-based approach*. CRC Press, 2009.

GÓMEZ-GASQUET, Pedro, ANDRÉS, Carlos y LARIO, Francisco-Cruz. An agent-based genetic algorithm for hybrid flowshops with sequence dependent setup times to minimize makespan. En: *Expert Systems with Applications*, 2012, vol. 39, no 9, p. 8095-8107.

HEIZER, J. y RENDER, B., *Dirección de la producción y de operaciones, Decisiones tácticas*, Editorial: Pearson Educación, 2008, 8va. Edición, p. 203.

IGLESIAS MARTÍNEZ, José y formales., y otros. *Complejidad Computacional*. Grado Ingeniería informática. Teoría de Autómatas y Lenguas.

JENABI, Masoud, et al.. Two hybrid meta-heuristics for the finite horizon ELSP in flexible flow lines with unrelated parallel machines. En: *Applied Mathematics and Computation*, 2007, vol. 186, no 1, p. 230-245.

JIMÉNEZ, Mario de J. Pérez y CAPARRINI, Fernando Sancho. *Máquinas moleculares basadas en ADN*. Universidad de Sevilla, 2003. p. 23.

JOHNSON, S.M. Optimal two-and three-stage production schedules with setup times included. En: *Naval Research Logistics Quarterly*. 1954, no 1. p. 61-68.

JUN, Sungbum y PARK, Jinwoo. A hybrid genetic algorithm for the hybrid flow shop scheduling problem with nighttime work and simultaneous work constraints: A case study from the transformer industry. En: *Expert Systems with Applications*, 2015, vol. 42, no 1.

JUNGWATTANAKIT, Jitti, et al.. A comparison of scheduling algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. En: Computers & Operations Research, 2009, vol. 36, no 2, p. 358-378.

JUNGWATTANAKIT, Jitti, et al.. Algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. En: The International Journal of Advanced Manufacturing Technology, 2008, vol. 37, no 3-4, p. 354-370.

JUNGWATTANAKIT, Jitti, et al.. Constructive and simulated annealing heuristics for hybrid flow shops with unrelated parallel machines. En: Proceedings of the 3rd OR-CRN Operations Research Conference, Bangkok, Thailand. 2006. p. 110-121.

KHALILI, M. y NADERI, B. A bi-objective imperialist competitive algorithm for no-wait flexible flow lines with sequence dependent setup times. En: Int J Adv Manuf Technol, 2015, vol. 76, p. 461-469.

KIM, H. W. y LEE, D. H. Heuristic algorithms for re-entrant hybrid flow shop scheduling with unrelated parallel machines. Proceedings of the Institution of Mechanical Engineers, En: Part B: Journal of Engineering Manufacture, 2009, vol. 223, no 4, p. 433-4.

LI, Jun-qing y PAN, Quan-ke. Solving the large-scale hybrid flow shop scheduling problem with limited buffers by a hybrid artificial bee colony algorithm. En: Information Sciences, 2014, vol. 316, p. 487-502.

LI, Zhan-tao, et al.. Approximation algorithms for the three-stage flexible flow shop problem with mid group constraint. En: Expert Systems with Applications, 2015, vol. 42, no 7, p. 3571-3584.

LI, Zhan-tao, et al.. Scheduling rules for two-stage flexible flow shop scheduling problem subject to tail group constraint. En: International Journal of Production Economics, 2013, vol. 146, no 2, p. 667-678.

LINN, Richard y ZHANG, Wei. Hybrid flow shop scheduling: a survey. En: Computers & industrial engineering, 1999, vol. 37, no 1, p. 57-61.

LÓPEZ J.C et al.. Reducción del Tiempo de Terminación en la Programación de la Producción de una Línea de Flujo Híbrida Flexible (HFS). En: Información Tecnológica, 2015, vol. 26(3), p. 157-172.

LOW, Chinyao, HSU, Chou-Jung y SU, Chwen-Tzeng. A two-stage hybrid flowshop scheduling problem with a function constraint and unrelated alternative machines. En: Computers & Operations Research, 2008, vol. 35, no 3, p. 845-853.

LOW, Chinyao. Simulated annealing heuristic for flow shop scheduling problems with unrelated parallel machines. En: Computers & Operations Research, 2005, vol. 32, no 8, p. 2013-2025.

MARICHELVAM, M.K. et al.. Improved cuckoo search algorithm for hybrid flow shop scheduling problems to minimize makespan. En: Applied Soft Computing, 2014, vol. 19, p. 93-101.

MARTÍ, Rafael y REINELT, Gerhard. The linear ordering problem: exact and heuristic methods in combinatorial optimization. Springer Science & Business Media, 2011.

MARTI, Rafael. Procedimientos metaheurísticos en optimización combinatoria. Matemáticas, 2003, vol. 1, no 1, p. 3-62.

MENDOZA, Franklin, MAGO, Enrique, REQUENA, Durylin. Herramientas computacionales "CIAPI-1": Una alternativa para la solución de problemas en sistemas de distribución de energía eléctrica. En: Memorias de las IX Jornadas de Investigación 2011 – UNEXPO Pue.

MIRANDA LUGO, Pedro, et al. Algoritmo genético com busca local para a programação da produção em sistemas flow-shop híbridos. En: Congreso Latino-Iberoamericano de Investigación Operativa (44:24-28, septiembre, Rio de Janeiro, Brasil). Publicaciones. Universidad.

MIRANDA LUGO, P.L. et al.. Um modelo de programação inteira mista para a programação da produção em flowshop híbrido com buffers limitados. En: Simpósio Brasileiro de Pesquisa Operacional, 2013, p. 154-165.

MORATTO CHIMENTY, ESELENDIS; PÉREZ FIGUEREDO, LUIS EDUARDO. Metaheurística basada en el algoritmo competitivo imperialista (ICA) aplicada a la solución del problema del flow shop híbrido (HFS) con máquinas paralelas no relacionadas. Trabajo de grado. Ingeniería Industrial. Universidad Industrial de Santander. 2016.

MONTGOMERY, Douglas C. Diseño y análisis de experimentos. Limusa Wiley. 2da edición. 2004.

NADERI, B. y YAZDANI, M. A model and imperialist competitive algorithm for hybrid flow shops with sublots and setup times. En: Journal of Manufacturing Systems, 2014, vol. 33, p. 647-653.

NAGANO, Marcelo y MOCCELLIN, João. Flowshops paralelos com processadores não-relacionados. En: Processing, 2002, p. 1–8.

NEJATI, M. et al.. Multi-job lot streaming to minimize the weighted completion time in a hybrid flow shop scheduling problem with work shift constraint. En: The International Journal of Advanced Manufacturing Technology, 2014, vol. 70, issue 1-4, p. 501-5.

NIEBLES ATENCIO, FABRICIO ANDRÉS. Aplicación de la metaheurística colonia de hormigas para la resolución de problemas multi-objetivo de programación de la producción en flow shops híbridos (flexibles). Trabajo de grado. Magíster en Gerencia de Operaciones.

PARRA, Javier y MEJÍA, Gonzalo. Desarrollo de un modelo para la secuenciación de trabajos en la mediana industria ladrillera de la localidad XIX. 2006.

RABIEE, M., et al.. An intelligent hybrid meta-heuristic for solving a case of no-wait two-stage flexible flow shop scheduling problem with unrelated parallel machines. En: The International Journal of Advanced Manufacturing Technology, 2014, vol. 71, no.

RASHIDI, E., JAHANDAR, M., ZANDICH, M., An improved hybrid multi-objective parallel genetic algorithm for hybrid flow shop scheduling with unrelated parallel machines. En: Int J Adv Manuf Technol 49. 2010. Páginas: 1129-1139.

RIBAS, Imma, LEISTEN, Rainer, FRAMIÑAN, Jose. Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective. En: Computers & operations research 37. 2010. Páginas: 1439-1454.

RUIZ, Ruben y MAROTO, Concepción. A genetic algorithm for hybrid flow shops with sequence dependent setup times and machine eligibility. En: European Journal of Operational Research, 2006, vol. 169, no 3, p. 781-800.

RUIZ, Rubén, ŞERIFOĞLU, Funda Sivrikaya y URLINGS, Thijs. Modeling realistic hybrid flexible flowshop scheduling problems. En: Computers & Operations Research, 2008, vol. 35, no 4, p. 1151-1175.

- SALAZAR, Eduardo y SARZURI, René. Algoritmo genético mejorado para la minimización de la tardanza total en un flowshop flexible con tiempos de preparación dependientes de la secuencia. En: *Ingeniare. Revista chilena de ingeniería*, 2015, vol. 23, p. 118-127.
- SOLTANI, S. Abolfazl y KARIMI, Behrooz. Cyclic hybrid flow shop scheduling problem with limited buffers and machine eligibility constraints. En: *The International Journal of Advanced Manufacturing Technology*, 2014, p. 1-17.
- TANG, D. et al.. Energy-efficient dynamic scheduling for a flexible flow shop using an improved particle swarm optimization, En: *Computers in Industry*, 2015.
- TRAN, T.H y NG, K.M. A water-flow algorithm for flexible flow shop scheduling with intermediate buffers. En: *J Sched*, 2011, vol. 14, p. 483-500.
- UETAKE, Toshifumi, TSUBONE, Hitoshi y OHBA, Masaaki. A production scheduling system in a hybrid flow shop. En: *International journal of production economics*, 1995, vol. 41, no 1, p. 395-398.
- WANG, K y CHOI, S.H. A holonic approach to flexible flow shop scheduling under stochastic processing times. En: *Computers & Operations Research*, 2014, vol. 43, p. 157-168.
- WANG, Wenpeng. Review on Hybrid Flow Shop Scheduling. En: *Information Technology, Computer Engineering and Management Sciences (ICM), 2011 International Conference on. IEEE*, 2011. p. 7-10.
- WEI, Q. et al.. A FPTAS for a two-stage hybrid flow shop problem and optimal algorithms for identical Jobs. En: *Theoretical Computer Science*, 2014, vol. 524, p. 78-89.
- WEI, Qi. An Optimal Algorithm for a Special Flow Shop Problem with Infinite Buffer Capacity. En: *Applied Mechanics and Materials Vols. 101-102*. 2012. Páginas 379-382.
- XU, Ye, et al.. An effective shuffled frog leaping algorithm for solving hybrid flow-shop scheduling problem. En: *Advanced Intelligent Computing*. Springer Berlin Heidelberg, 2012. p. 560-567.
- YAGMAHAN, Betul y YENISEY, Mehmet Mutlu. Scheduling practice and recent developments in flow shop and job shop scheduling. En *Computational intelligence in flow shop and job shop scheduling*. Springer Berlin Heidelberg, 2009. p. 261-300.

YEURIMA, Victor, BURTEVA, Larisa y TCHERNYKH, Andrei. Hybrid flowshop with unrelated machines, sequence dependent setup time and availability constraints: An enhanced crossover operator for a genetic algorithm. Parallel processing and applied mathematics.

YEURIMA, Victor, BURTEVA, Larisa y TCHERNYKH, Andrei. Hybrid flowshop with unrelated machines, sequence-dependent setup time, availability constraints and limited buffers. En: Computers & Industrial Engineering, 2009, vol. 56, no 4, p. 1452-1463.

YU, Xinjie y GEN, Mitsuo. Introduction to evolutionary algorithms. Springer Science & Business Media, 2010. p. 263.

YU, Y y LI, T. Hybrid Algorithms for Hybrid Flow Shop Scheduling considering Limited Buffers and Due Dates. En: Advances in information Sciences and Service Sciences(AISS), 2013, vol. 5, num. 10, p. 861-868.

ZABIHZADEH, S.S y REZAEIAN, J. Two meta-heuristic algorithms for flexible flow shop scheduling problem with robotic transportation and release time. En: Applied Soft Computing, 2016, vol. 40, p. 319-330.

ZANDIEH, M., MOZAFFARI, E. y GHOLAMI, Mansour. A robust genetic algorithm for scheduling realistic hybrid flexible flow line problems. En: Journal of Intelligent Manufacturing, 2010, vol. 21, no 6, p. 731-743.

## ANEXOS

**Anexo A.** Detalle del modelo matemático utilizado para el problema del *HFS* con *buffers* limitados escrito en lenguaje *GAMS*.

A continuación, se muestra el detalle del pseudocódigo en lenguaje *GAMS* utilizado para obtener la respuesta exacta para la instancia *i7j2k3(5)-1*.

```

* DECLARACIÓN DE ÍNDICES Y CONJUNTOS
SETS
I Trabajos /0*7/           SETS: Presenta los diferentes subíndices usados.
k Etapas /1*5/           ALIAS: Habilita la posibilidad de declarar varios
j Máquinas /M1,M2/;      nombres para un mismo grupo (ej: grupo de
alias (h,i,g);           trabajos a procesar)

* DECLARACIÓN DE PARÁMETROS
DISPLAY I;
table
P(i,j,k) "Tiempo de procesamiento de la tarea i en la máquina j, de la estación k"
    1      2      3      4      5
0. M1     0      0      0      0      0
0. M2     0      5000  0      5000  0
1. M1    119     0      75     0     101
1. M2     92     5000  119    5000  118
2. M1     88     0      78     0      79
2. M2    123     5000  85     5000  117
3. M1    112     0      83     0     100
3. M2     96     5000  78     5000  104
4. M1    121     0     121     0      98
4. M2     90     5000  115    5000  117
5. M1    121     0      87     0     106
5. M2    122     5000  113    5000  77
6. M1    107     0     102     0     110
6. M2     89     5000  119    5000  96
7. M1    119     0     118     0      98
7. M2     96     5000  125    5000  88;

table
S(h,i,j,k) "Tiempo de alistamiento entre las tareas h e i, en la máquina j, de la estación k"
    1      2      3      4      5
0. 1. M1   48     0      66     0      73
0. 1. M2   26     5000  74     5000  51
0. 2. M1   58     0      67     0      29
0. 2. M2   28     5000  55     5000  54
0. 3. M1   69     0      26     0      40
0. 3. M2   43     5000  31     5000  48
0. 4. M1   41     0      29     0      73
0. 4. M2   69     5000  61     5000  53
0. 5. M1   54     0      37     0      41

```

**PARÁMETROS:** Son datos de entrada al problema como por ejemplo los tiempos de procesamiento y alistamiento para cada instancia o un número escalar utilizado en las restricciones.

0. 5. M2	70	5000	73	5000	51
0. 6. M1	28	0	73	0	69
0. 6. M2	42	5000	66	5000	73
0. 7. M1	47	0	48	0	52
0. 7. M2	68	5000	38	5000	46
1. 2. M1	63	0	28	0	41
1. 2. M2	43	5000	52	5000	72
1. 3. M1	34	0	73	0	55
1. 3. M2	32	5000	51	5000	49
1. 4. M1	68	0	40	0	43
1. 4. M2	66	5000	50	5000	44
1. 5. M1	61	0	49	0	31
1. 5. M2	39	5000	42	5000	29
1. 6. M1	69	0	40	0	59
1. 6. M2	41	5000	55	5000	59
1. 7. M1	64	0	65	0	25
1. 7. M2	73	5000	56	5000	70
2. 1. M1	27	0	32	0	63
2. 1. M2	74	5000	58	5000	60
2. 3. M1	26	0	52	0	59
2. 3. M2	72	5000	29	5000	49
2. 4. M1	30	0	67	0	47
2. 4. M2	31	5000	60	5000	46
2. 5. M1	63	0	56	0	47
2. 5. M2	58	5000	40	5000	68
2. 6. M1	72	0	57	0	38
2. 6. M2	41	5000	48	5000	38
2. 7. M1	50	0	34	0	66
2. 7. M2	62	5000	67	5000	61
3. 1. M1	64	0	28	0	36
3. 1. M2	44	5000	68	5000	62
3. 2. M1	37	0	26	0	63
3. 2. M2	42	5000	37	5000	49
3. 4. M1	71	0	32	0	69
3. 4. M2	57	5000	37	5000	62
3. 5. M1	44	0	62	0	64
3. 5. M2	66	5000	54	5000	61
3. 6. M1	60	0	46	0	67
3. 6. M2	52	5000	33	5000	52
3. 7. M1	60	0	58	0	56
3. 7. M2	44	5000	63	5000	38
4. 1. M1	34	0	25	0	30
4. 1. M2	73	5000	28	5000	53
4. 2. M1	65	0	48	0	62
4. 2. M2	55	5000	66	5000	65
4. 3. M1	37	0	39	0	26
4. 3. M2	66	5000	26	5000	57
4. 5. M1	35	0	40	0	54
4. 5. M2	54	5000	57	5000	34
4. 6. M1	65	0	51	0	33
4. 6. M2	27	5000	69	5000	51
4. 7. M1	36	0	57	0	70
4. 7. M2	51	5000	27	5000	68
5. 1. M1	38	0	73	0	52

5. 1. M2	49	5000	72	5000	52
5. 2. M1	34	0	59	0	69
5. 2. M2	34	5000	53	5000	62
5. 3. M1	30	0	66	0	72
5. 3. M2	51	5000	45	5000	28
5. 4. M1	49	0	61	0	28
5. 4. M2	57	5000	52	5000	36
5. 6. M1	25	0	51	0	26
5. 6. M2	69	5000	73	5000	33
5. 7. M1	58	0	60	0	52
5. 7. M2	27	5000	65	5000	32
6. 1. M1	27	0	31	0	48
6. 1. M2	43	5000	54	5000	58
6. 2. M1	74	0	48	0	67
6. 2. M2	55	5000	65	5000	46
6. 3. M1	28	0	38	0	31
6. 3. M2	47	5000	35	5000	63
6. 4. M1	28	0	28	0	45
6. 4. M2	32	5000	66	5000	34
6. 5. M1	25	0	34	0	55
6. 5. M2	69	5000	48	5000	52
6. 7. M1	74	0	66	0	43
6. 7. M2	41	5000	69	5000	68
7. 1. M1	69	0	28	0	73
7. 1. M2	62	5000	34	5000	61
7. 2. M1	41	0	72	0	44
7. 2. M2	67	5000	50	5000	47
7. 3. M1	70	0	59	0	48
7. 3. M2	70	5000	31	5000	31
7. 4. M1	45	0	40	0	53
7. 4. M2	36	5000	29	5000	66
7. 5. M1	52	0	56	0	28
7. 5. M2	34	5000	58	5000	26
7. 6. M1	47	0	71	0	35
7. 6. M2	56	5000	40	5000	70;

```
scalar B "Penalización" /50000/;
```

```
*DECLARACIÓN DE VARIABLES DEL PROBLEMA
```

```
VARIABLES
```

```
Y(h,i,j,k) Si la tarea h precede a la tarea i en la máquina j de la estación k
```

```
C(i,k) Tiempo de finalización de la tarea i en la estación k
```

```
E(i,k) Tiempo de salida de la tarea i de la estación k
```

```
CMAX Tiempo de Completamiento
```

```
MAKESPAN Makespan;
```

```
*CONDICIONES DE LAS VARIABLES DEL PROBLEMA
```

```
binary variable Y(h,i,j,k)
```

```
positive variable C(i,k)
```

```
positive variable E(i,k);
```

VARIABLES: Se presentan las variables que son necesarias que el GAMS devuelva en forma de respuesta junto a sus características y condiciones.

\*DECLARACIÓN DE LAS RESTRICCIONES DEL PROBLEMA

EQUATIONS EQUATIONS: Presenta las diferentes restricciones para el problema las cuales buscan caracterizarlo o evitar errores.

```
e0 Makespan
e1 Asegura que cada tarea debe ser precedida por únicamente otra tarea además garantiza la no división de trabajos en lotes de procesamiento
e2 Asegura que cada trabajo asignado debe tener a lo sumo un sucesor
e3 Asegura que una tarea procesada en una máquina asignada debe tener como máximo un predecesor dentro de la misma máquina
e4 Asegura la no existencia de precedencias cruzadas
e5 Asegura que el trabajo ficticio inicial cero debe tener como máximo un solo sucesor
e6 Asegura que el tiempo de completamiento del trabajo ficticio inicial sea cero
e7 Asegura que ninguna tarea sea procesada en una máquina asignada sin que el trabajo anterior dentro de la misma máquina haya salido
e8 Asegura que cada trabajo sea procesado de manera subsecuente en todas las etapas
e9 Cálculo del tiempo de salida de cada tarea en cada etapa
e10 Asegura que que ningún trabajo deje una estación antes que su procesamiento en la misma esté finalizado
e11 Asegura que cada trabajo salga del sistema cuando su procesamiento en la última estación esté finalizado
e12 Cálculo del makespan;

e0.. MAKESPAN=e=CMAX;
e1(i,k)$ (ord(i)>1) .. sum((j,h)$ (not sameas(h,i)), Y(h,i,j,k))=e=1;
e2(i,k)$ (ord(i)>1) .. sum((j,h)$ (not sameas(h,i) and (ord(h)>1)), Y(i,h,j,k))=1=1;
e3(h,i,j,k)$ ((not sameas(h,i) and (ord(h)>1) and (ord(i)>1)) .. sum((g)$ (not sameas(h,g) and (not sameas(i,g))), Y(g,h,j,k))=g=Y(h,i,j,k);
e4(h,i,k)$ ((not sameas(h,i) and (ord(i)>1)) .. sum((j), Y(h,i,j,k)+Y(i,h,j,k))=1=1;
e5(h,j,k)$ (ord(h)=1) .. sum((i)$ (not sameas(h,i) and (ord(i)>1)), Y(h,i,j,k))=1=1;
e6(i,k)$ (ord(i)=1) .. C(i,k)=e=0;
e7(h,i,j,k)$ ((not sameas(h,i) and (ord(i)>1)) .. C(i,k)+B*(1-Y(h,i,j,k))=g=E(h,k)+S(h,i,j,k)+P(i,j,k);
*e7(h,i,j,k)$ ((not sameas(h,i) and (ord(i)>1)) .. C(i,k)+B*(1-Y(h,i,j,k))=g=E(h,k)+(Y(h,i,j,k)*(S(h,i,j,k)+P(i,j,k)));
e8(h,i,j,k)$ ((not sameas(h,i) and (ord(i)>1) and (ord(k)>1)) .. C(i,k)+B*(1-Y(h,i,j,k))=g=C(i,k-1)+S(h,i,j,k)+P(i,j,k);
*e8(h,i,j,k)$ ((not sameas(h,i) and (ord(i)>1) and (ord(k)>1)) .. C(i,k)+B*(1-Y(h,i,j,k))=g=C(i,k-1)+(Y(h,i,j,k)*(S(h,i,j,k)+P(i,j,k)));
e9(i,k)$ (ord(i)>1 and (ord(k)>1)) .. E(i,k-1)=e=C(i,k)-sum((j,h)$ (not sameas(h,i)), (Y(h,i,j,k)*P(i,j,k)))-sum((j,h)$ (not sameas(h,i)), (Y(h,i,j,k)*S(h,i,j,k)));
e10(i,k)$ (ord(k)<5) .. E(i,k)=g=C(i,k);
e11(i,k)$ (ord(k)=5) .. E(i,k)=e=C(i,k);
e12(i,k)$ (ord(k)>1) .. CMAX=g=C(i,k);

*DECLARACIÓN DEL TIPO DE MODELO Y DE FUNCIÓN A OPTIMIZAR
*option MIP=convert;
*option reslim=10800;
MODEL FLOWSHOP /e0,e1,e2,e3,e4,e5,e6,e7,e8,e9,e10,e11,e12/
solve FLOWSHOP USING MIP MINIMIZING MAKESPAN;
```

La sección final presenta restricciones pero relacionadas con el software GAMS como tiempos límite de solución, restricciones a tomar en cuenta o no y tipo de problema a solucionar junto con su variable objetivo.

**Anexo B.** Tablas con los tiempos de procesamiento y alistamiento de las tareas para cada una de las instancias.

A continuación, se presenta el detalle de los tiempos utilizados para cada una de las 20 instancias con las que se valida el funcionamiento de la metaheurística basada en el *ICA* para el problema del *HFS* con *buffers* limitados.

Cabe la pena recordar que el subíndice *i* representa el número de trabajo, el *j* el número de máquinas por estación, la *k* el número de etapas y luego del guion (-) se presenta el tamaño del *buffer* para cada instancia.

En la segunda parte del anexo se presentan los tiempos de procesamiento y alistamiento utilizados por Moratto y Pérez<sup>123</sup> en su proyecto de grado y los cuales se utilizan el presente para la comparación entre los resultados arrojados por el *ICA* para el *HFS* con *buffers* de tamaño limitado y el *ICA* para *HFS* con *buffers* de tamaño ilimitado.

---

<sup>123</sup> MORATTO CHIMENTY, ESELENDIS; PÉREZ FIGUEREDO, LUIS EDUARDO. Op. Cit. p. 63.

**Tabla 27.** Tiempos de procesamiento y alistamiento de tareas para la instancia i7j2k3-1.

Tiempos de Alistamiento																																
k=	1							2							1							2										
j=	1							2							1							2										
h=	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1	48	█	27	64	34	38	27	69	26	█	74	44	73	49	43	62	66	█	32	28	25	73	31	28	74	█	58	68	28	72	54	34
2	58	63	█	37	65	34	74	41	28	43	█	42	55	34	55	67	67	28	█	26	48	59	48	72	55	52	█	37	66	53	65	50
3	69	34	26	█	37	30	28	70	43	32	72	█	66	51	47	70	26	73	52	█	39	66	38	59	31	51	29	█	26	45	35	31
4	41	68	30	71	█	49	28	45	69	66	31	57	█	57	32	36	29	40	67	32	█	61	28	40	61	50	60	37	█	52	66	29
5	54	61	63	44	35	█	25	52	70	39	58	66	54	█	69	34	37	49	56	62	40	█	34	56	73	42	40	54	57	█	48	58
6	28	69	72	60	65	25	█	47	42	41	41	52	27	69	█	56	73	40	57	46	51	51	█	71	66	55	48	33	69	73	█	40
7	47	64	50	60	36	58	74	█	68	73	62	44	51	27	41	█	48	65	34	58	57	60	66	█	38	56	67	63	27	65	69	█

k=	3															
j=	1			2												
h=	0	1	2	3	4	5	6	7								
1	73	█	63	36	30	52	48	73	51	█	60	62	53	52	58	61
2	29	41	█	63	62	69	67	44	54	72	█	49	65	62	46	47
3	40	55	59	█	26	72	31	48	48	49	49	█	57	28	63	31
4	73	43	47	69	█	28	45	53	53	44	46	62	█	36	34	66
5	41	31	47	64	54	█	55	28	51	29	68	61	34	█	52	26
6	69	59	38	67	33	26	█	35	73	59	38	52	51	33	█	70
7	52	25	66	56	70	52	43	█	46	70	61	38	68	32	68	█

Tiempos de Procesamiento						
k=	1		2		3	
j=	1	2	1	2	1	2
h=						
1	119	92	75	119	101	118
2	88	123	78	85	79	117
3	112	96	83	78	100	104
4	121	90	121	115	98	117
5	121	122	87	113	106	77
6	107	89	102	119	110	96
7	119	96	118	125	98	88

**Tabla 28.** Tiempos de procesamiento y alistamiento de tareas para la instancia i7j2k3-2.

Tiempos de Alistamiento																																
k=	1														2																	
j=	1							2							1						2											
h=	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1	74	█	70	46	31	32	50	68	67	█	72	45	74	67	68	50	50	█	73	60	55	47	71	46	48	█	72	51	39	49	31	38
2	61	44	█	38	63	60	59	73	58	69	█	44	58	53	51	52	61	39	█	41	70	45	44	59	74	33	█	30	72	44	49	48
3	71	65	73	█	42	66	55	53	67	59	66	█	43	45	45	52	72	42	52	█	49	71	34	34	67	59	50	█	34	39	26	29
4	59	65	43	26	█	43	59	31	28	55	41	36	█	31	51	70	43	47	49	71	█	29	72	56	58	28	28	68	█	52	59	37
5	27	40	65	62	48	█	58	33	58	44	72	46	28	█	68	52	71	43	72	38	55	█	67	55	68	28	58	26	57	█	47	33
6	27	41	26	59	59	67	█	65	31	38	57	40	39	71	█	52	29	60	73	49	34	61	█	57	28	35	60	62	71	43	█	68
7	52	42	47	54	64	33	28	█	36	73	42	25	35	72	31	█	26	44	68	25	60	73	43	█	73	43	30	58	62	49	58	█

k=	3															
j=	1							2								
h=	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1	52	█	68	70	55	39	53	51	32	█	43	62	37	37	38	49
2	33	59	█	48	68	30	62	32	32	61	█	40	33	53	36	39
3	42	66	46	█	48	61	58	74	36	56	49	█	61	58	59	67
4	42	41	72	64	█	74	32	62	63	61	56	44	█	67	74	48
5	31	44	49	31	34	█	68	32	59	73	66	45	40	█	60	34
6	61	45	38	52	62	59	█	30	27	61	44	50	47	61	█	71
7	49	65	31	57	73	61	69	█	33	53	51	29	43	34	59	█

Tiempos de Procesamiento						
k=	1		2		3	
j=	1	2	1	2	1	2
h=						
1	121	113	95	89	76	76
2	83	93	95	105	80	125
3	125	124	122	79	125	101
4	115	124	115	117	92	122
5	110	108	79	93	79	120
6	85	94	108	91	101	92
7	107	105	107	94	79	124

**Tabla 29.** Tiempos de procesamiento y alistamiento de tareas para la instancia i7j3k3-1.

Tiempos de Alistamiento																																																
k=	1							2							3							2							3																			
j=	1							2							3							1							2							3												
h=	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1	50	74	42	29	28	30	63	41	58	72	28	70	48	25	65	48	50	69	31	71	38	30	74	56	52	34	47	36	65	27	25	26	44	73	43	74	46	57	50	72	73	26						
2	60	29	65	33	61	25	73	62	63	48	35	63	28	67	54	38	28	46	66	42	48	61	38	55	68	45	52	55	37	58	38	53	67	68	29	65	39	68	40	37	41	65						
3	58	48	65	64	47	30	74	36	44	51	65	68	73	74	26	52	35	31	52	34	74	33	51	49	68	57	35	60	72	36	70	64	60	62	42	30	45	36	33	32	63	65						
4	67	72	41	38	74	54	30	62	30	27	70	33	34	59	47	58	37	46	37	27	40	72	42	62	26	42	66	65	44	63	50	63	55	45	37	38	59	53	28	26	43	37						
5	39	27	25	74	56	60	29	50	64	42	48	70	28	71	55	49	45	48	29	25	68	52	42	64	30	64	61	74	33	35	55	51	54	25	49	30	66	40	28	73	35	42						
6	60	64	41	45	67	64	26	37	40	49	47	28	72	72	42	57	74	49	56	46	46	71	60	65	55	52	71	64	59	56	65	32	60	44	28	54	45	31	53	42	48							
7	55	28	28	54	63	62	47	48	35	55	38	31	58	43	61	72	30	71	29	51	38	72	72	35	53	44	39	28	57	51	31	32	69	73	49	53	58	47	33	27	52	71						

k=	3																							
j=	1							2							3									
h=	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1	45	53	32	28	46	36	30	34	59	31	66	42	29	35	46	65	74	61	44	27	38			
2	38	41	60	53	61	43	51	47	27	70	55	68	29	74	72	55	26	45	59	48	31			
3	56	26	57	48	40	48	69	51	65	45	67	29	63	57	43	31	49	29	28	25	54			
4	51	60	51	74	73	27	41	62	74	46	44	72	32	33	35	61	42	65	57	39	43			
5	61	31	52	36	59	40	52	47	62	29	50	68	46	32	34	39	48	28	48	67	39			
6	27	25	50	70	64	52	49	73	41	49	57	42	65	38	71	73	47	30	50	56	71			
7	33	60	55	46	28	66	39	70	70	26	61	52	47	48	58	47	65	57	25	73	73			

Tiempos de Procesamiento									
k=	1			2			3		
j=	1	2	3	1	2	3	1	2	3
h=									
1	104	82	100	117	79	88	110	104	99
2	111	120	97	117	105	75	105	110	109
3	86	124	81	81	89	123	112	105	94
4	82	95	121	103	79	84	95	106	99
5	100	104	83	86	86	80	115	86	108
6	103	101	119	110	75	125	101	75	115
7	107	102	94	91	89	93	125	91	114

**Tabla 30.** Tiempos de procesamiento y alistamiento de tareas para la instancia i7j3k3-2.

Tiempos de Alistamiento																																																
k=	1							2							3							2							3																			
j=	1							2							3							1							2							3												
h=	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1	30	62	32	61	49	74	66	35	38	31	43	52	48	30	69	39	37	30	31	56	35	31	25	50	72	35	60	43	40	63	71	62	40	70	71	39	69	69	28	45	71	65						
2	74	41	63	34	69	41	73	52	44	31	36	63	31	64	44	45	72	52	50	55	60	72	42	42	46	45	31	50	45	43	47	65	34	36	29	26	62	32	68	54	50	64						
3	54	29	29	73	73	63	56	39	60	66	28	62	50	70	31	40	39	40	54	59	72	37	55	47	63	35	70	67	49	54	41	61	52	36	61	67	64	51	72	71	44	74						
4	35	63	48	52	68	58	28	57	69	52	57	69	35	60	30	38	25	33	31	67	52	55	32	60	70	61	70	61	56	64	49	43	52	38	30	39	35	52	55	68	45	63						
5	57	47	65	37	71	71	53	34	49	70	72	61	71	62	39	72	40	66	33	32	42	55	72	64	38	29	53	59	39	52	29	43	25	31	31	49	31	45	65	67	68							
6	32	32	66	49	59	49	72	54	70	49	63	50	47	32	60	72	66	58	34	55	39	40	50	38	73	63	31	68	72	37	59	61	52	66	28	50	50	58	63	46	53	65						
7	46	35	66	67	62	27	31	70	49	48	57	73	41	34	31	31	63	28	41	55	45	25	52	66	58	44	33	53	27	49	31	31	71	41	38	53	69	52	46	74	38	68						

k=	3																							
j=	1							2							3									
h=	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1	64	41	70	68	52	34	67	43	69	35	40	37	40	39	43	26	55	52	41	55	59			
2	39	71	74	54	70	30	57	28	63	64	60	45	32	43	44	57	51	47	68	60	44			
3	45	33	51	74	61	42	53	27	50	26	64	62	54	38	67	34	65	39	71	67	48			
4	69	26	36	62	39	58	69	69	60	60	38	72	55	45	29	29	25	60	33	30	26			
5	25	52	30	49	44	54	27	49	59	30	71	31	30	52	43	52	62	53	55	65	30			
6	33	48	31	40	35	55	44	38	46	65	58	42	42	41	66	63	37	45	67	64	68			
7	42	71	31	65	50	41	30	25	46	73	61	56	33	66	53	51	27	70	40	69	62			

Tiempos de Procesamiento									
k=	1			2			3		
j=	1	2	3	1	2	3	1	2	3
h=									
1	78	115	120	110	93	79	104	93	101
2	116	105	113	82	113	84	103	80	100
3	96	102	84	99	123	79	94	97	116
4	101	113	84	89	121	105	122	95	125
5	116	107	95	109	77	108	108	113	111
6	113	87	91	84	95	78	94	112	99
7	106	112	115	88	103	106	101	107	86

**Tabla 31.** Tiempos de procesamiento y alistamiento de tareas para la instancia i7j3k3-3.

Tiempos de Alistamiento																																																								
k=	1							2							3							1							2							3																				
j=	1							2							3							1							2							3																				
h=	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1	48	59	69	48	48	51	58	62	62	48	61	56	27	43	29	59	46	30	40	33	42	43	46	62	36	70	74	74	52	59	41	48	59	35	32	64	67	41	42	50	32	73														
2	28	64	66	26	49	56	44	50	47	70	40	67	40	46	52	53	41	34	56	43	52	56	53	51	29	35	42	65	65	32	42	61	55	63	70	54	67	62	25	30	53	27														
3	67	23	39	69	52	49	33	50	32	71	48	38	33	63	60	32	44	69	50	27	27	49	62	47	67	61	70	38	41	46	58	64	40	51	53	34	37	31	45	70	59	41														
4	31	58	26	49	59	45	36	70	58	40	45	38	30	26	39	44	27	46	38	61	52	70	26	48	66	51	51	62	37	40	63	71	70	67	50	62	62	41	25	70	52	34														
5	63	38	52	49	30	34	41	59	63	36	70	73	34	47	58	54	49	36	71	29	36	38	35	31	55	29	31	44	50	45	31	60	43	42	68	63	26	48	66	29	33	29														
6	40	31	35	32	50	25	40	52	66	27	31	34	52	52	33	41	66	38	48	65	66	67	39	40	67	53	73	28	27	41	64	62	68	48	25	35	66	42	56	65	73	32														
7	69	59	30	35	34	39	28	32	33	31	74	54	65	39	63	30	71	32	56	25	56	34	66	49	33	62	61	31	47	74	29	39	72	69	29	71	56	35	53	29	49	58														

k=	3																							
j=	1							2							3									
h=	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1	64	47	34	69	54	42	56	65	55	46	55	25	72	49	32	65	28	56	29	49	40			
2	71	29	49	66	26	59	53	33	73	74	70	66	72	69	25	42	53	66	67	34	49			
3	51	26	61	49	53	26	25	39	46	60	51	34	70	58	68	54	46	47	62	36	52			
4	30	74	48	29	67	52	32	59	33	31	52	73	30	25	69	65	31	61	54	72	39			
5	53	73	39	43	68	42	66	44	27	31	37	51	46	47	47	70	58	68	44	51	28			
6	55	36	26	72	73	59	50	47	41	71	59	62	69	27	57	74	33	58	66	25	69			
7	61	62	60	34	52	55	59	43	68	37	43	25	30	46	44	69	67	43	53	69	44			

Tiempos de Procesamiento									
k=	1			2			3		
j=	1	2	3	1	2	3	1	2	3
h=									
1	84	103	112	104	96	108	109	86	118
2	103	85	100	109	83	103	97	103	86
3	94	83	75	109	88	99	109	111	105
4	113	82	77	95	115	108	90	116	105
5	98	84	122	102	93	88	109	79	121
6	115	116	123	118	114	124	79	95	119
7	122	89	106	75	82	109	102	99	125

**Tabla 32.** Tiempos de procesamiento y alistamiento de tareas para la instancia i7j2k5-1.

Tiempos de Alistamiento																																
k=	1														2																	
j=	1							2							1						2											
h=	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1	36	52	30	26	49	65	42	34	61	56	27	72	29	49	74	47	26	67	60	58	52	70	45	50	66	65	30	70				
2	31	58	28	31	28	29	55	67	49	71	53	66	41	64	71	46	65	63	29	67	29	59	30	72	46	29	51	65				
3	41	69	41	53	29	42	48	42	43	59	69	51	36	32	34	72	51	45	36	72	31	40	61	72	44	68	59	33				
4	40	51	28	29	73	65	52	43	68	27	37	40	42	54	28	73	29	29	73	26	65	67	62	72	57	31	60	69				
5	69	43	50	44	70	38	61	28	56	26	52	51	60	26	65	27	31	42	62	72	34	25	52	38	51	47	48	41				
6	64	57	36	57	72	38	61	38	36	58	36	73	40	37	36	32	49	69	25	71	52	33	29	54	57	59	40	63				
7	56	25	57	73	55	25	27	50	50	73	39	65	26	28	25	48	68	29	67	30	58	73	25	68	58	35	74	67				

k=	3														4																	
j=	1							2							1						2											
h=	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1	50	52	29	59	49	67	41	49	58	37	67	43	47	58	40	43	28	39	52	25	65	57	32	37	42	32	40	51				
2	52	29	53	46	26	31	64	39	59	58	61	59	55	52	48	36	35	72	45	56	55	52	28	27	35	28	73	36				
3	32	61	74	40	52	57	38	53	74	28	54	71	71	36	37	60	33	30	34	52	52	54	58	32	36	67	58	36				
4	43	48	35	61	31	61	70	58	56	66	51	73	47	71	46	32	45	59	35	68	61	50	25	27	42	37	35	28				
5	27	38	59	35	27	27	41	26	51	53	47	57	30	59	64	25	69	49	51	71	26	45	47	69	44	74	28	37				
6	41	31	31	73	35	62	38	46	37	59	42	55	55	54	73	68	61	50	49	68	73	55	49	25	29	73	54	65				
7	46	70	56	35	36	57	61	66	27	44	62	65	39	54	27	64	58	36	44	43	32	49	42	25	48	70	29	32				

k=	5															
j=	1							2								
h=	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1	30	44	73	64	44	40	47	64	43	64	50	54	61	70		
2	52	43	60	46	56	40	36	52	49	41	69	54	51	39		
3	40	74	31	37	45	45	63	25	69	69	65	61	58	29		
4	40	46	41	29	67	62	63	31	60	66	27	39	61	68		
5	46	26	58	61	58	45	57	59	27	59	53	32	47	36		
6	55	50	72	45	62	35	68	55	28	41	51	72	55	60		
7	54	49	58	39	48	35	28	54	73	33	65	56	32	51		

Tiempos de Procesamiento											
k=	1		2		3		4		5		
j=	1	2	1	2	1	2	1	2	1	2	
1	76	115	117	100	90	98	109	84	90	116	
2	97	113	80	99	81	86	80	118	90	120	
3	123	117	88	94	83	96	96	107	117	108	
4	106	112	101	110	111	118	116	91	75	94	
5	89	82	121	125	121	118	91	124	99	86	
6	110	113	110	108	93	116	102	79	120	89	
7	86	104	91	75	111	103	98	103	112	97	

**Tabla 33.** Tiempos de procesamiento y alistamiento de tareas para la instancia i7j2k5-2.

		Tiempos de Alistamiento																															
k=		1														2																	
j=		1							2							1							2										
h=		0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1		44	37	67	34	38	53	73	51	39	49	40	40	45	26	48	51	73	37	43	33	54	42	50	40	73	35	46	57				
2		66	59	45	51	39	51	66	41	47	58	33	55	56	70	32	43	31	26	40	72	47	64	73	65	34	47	48	70				
3		33	25	59	71	40	39	62	72	59	43	28	40	27	47	70	74	52	62	62	65	62	47	32	61	65	29	59	59				
4		61	52	36	57	45	47	28	73	70	74	73	52	26	74	45	61	55	48	33	39	34	41	52	31	60	28	26	37				
5		62	32	46	71	45	25	61	28	59	40	70	72	50	37	74	49	29	32	51	45	68	45	61	43	41	29	49	54				
6		30	52	73	36	71	59	48	62	65	32	28	59	48	31	52	32	53	74	46	69	33	48	33	71	49	29	71	41				
7		40	58	73	48	52	38	34	67	48	50	31	36	59	39	66	66	46	25	30	40	25	53	26	43	38	64	61	34				

k=		3														4																	
j=		1							2							1							2										
h=		0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1		68	60	46	51	40	52	36	54	25	44	68	61	53	39	54	53	47	54	63	36	31	64	30	41	56	53	29	61				
2		41	62	70	34	54	34	30	49	36	28	63	39	63	25	64	42	38	70	74	70	61	30	65	27	57	33	43	50				
3		73	39	50	36	44	36	59	60	37	55	53	36	73	60	40	26	56	63	74	43	54	65	40	25	70	69	25	32				
4		32	68	50	61	52	65	60	34	48	64	39	52	52	53	47	56	47	60	25	56	53	26	26	25	48	64	27	58				
5		30	51	46	64	40	59	58	48	52	53	68	39	41	67	53	35	50	67	43	74	49	38	50	69	51	39	39	25				
6		37	44	27	27	61	57	59	26	30	71	56	35	55	70	65	65	29	58	35	53	29	66	26	63	73	26	40	28				
7		66	74	66	53	64	29	71	26	62	54	49	62	36	43	69	51	42	74	34	67	73	65	44	60	62	42	63	65				

k=		5															
j=		1							2								
h=		0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1		61	72	49	51	26	34	44	51	32	72	38	30	41	46		
2		32	38	53	61	48	64	56	45	28	31	62	33	49			
3		46	60	54	60	33	43	28	64	61	39	71	67	70	70		
4		60	37	46	65	39	68	52	53	71	69	44	70	58	68		
5		54	27	36	64	51	57	74	73	50	37	45	41	65	40		
6		73	38	53	25	38	43	71	30	30	39	32	36	73	31		
7		41	41	33	47	60	49	28	58	47	41	63	26	43	53		

		Tiempos de Procesamiento									
k=		1		2		3		4		5	
j=		1	2	1	2	1	2	1	2	1	2
h=											
1		85	105	102	121	96	81	121	112	84	84
2		80	79	101	79	97	99	103	75	107	99
3		124	114	122	3	123	118	107	92	75	99
4		90	124	90	106	123	100	111	77	79	85
5		108	83	125	95	113	81	110	78	94	93
6		116	114	78	80	120	104	111	76	79	81
7		98	80	96	108	102	87	109	75	89	123

**Tabla 34.** Tiempos de procesamiento y alistamiento de tareas para la instancia i7j3k5-1.

Tiempos de Alistamiento																																										
k=	1							2																																		
j=	1		2		3			1		2		3																														
h=	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7										
1	52	52	44	54	34	60	73	40	34	54	37	34	62	52	57	39	64	62	29	51	55	27	65	39	72	34	61	56	37	46	65	26	74	65	70	72	35	26	42	69	39	29
2	57	25	72	41	47	64	59	64	74	58	37	39	53	62	52	65	57	26	27	47	48	26	58	59	36	26	57	57	74	33	30	65	46	34	51	71	30	60	38	41	52	52
3	27	66	52	28	38	30	46	50	56	47	45	68	63	34	53	26	43	54	69	70	44	63	39	64	68	67	64	71	55	62	73	29	31	42	30	50	49	47	25	54	59	63
4	33	38	29	61	35	47	64	45	51	35	30	26	66	30	32	52	63	62	66	30	55	38	44	37	60	29	73	73	56	58	37	58	71	44	71	64	28	62	46	25	32	32
5	28	29	49	53	37	57	63	44	66	25	30	29	33	28	40	56	27	60	61	52	72	37	50	57	71	73	58	68	39	55	44	74	44	49	59	67	30	72	34	61	55	59
6	69	49	67	73	51	38	59	46	46	55	73	70	44	74	25	47	25	29	40	25	27	41	64	41	60	26	63	40	28	40	67	59	56	37	61	62	49	35	40	26	42	33
7	69	62	49	47	51	26	68	34	37	32	54	64	25	36	59	35	62	39	68	29	28	50	33	39	64	55	49	67	29	36	70	53	33	72	25	72	42	31	47	61	44	49

k=	3							4																																		
j=	1		2		3			1		2		3																														
h=	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7										
1	41	62	62	59	47	30	35	37	46	31	74	73	31	49	53	54	27	68	54	50	73	54	49	52	53	44	55	65	32	72	73	37	43	67	53	32	39	50	74	36	69	35
2	38	65	43	38	36	58	74	31	68	54	47	50	74	73	69	27	44	38	34	69	33	34	30	49	52	44	44	54	65	56	44	29	62	65	54	74	48	39	34	27	54	25
3	39	31	69	47	50	54	35	41	47	48	72	68	26	47	41	44	26	68	42	28	29	64	39	51	28	68	63	28	39	50	60	61	56	65	37	35	42	42	48	53	42	74
4	70	71	72	41	68	26	50	54	57	64	61	73	46	54	60	31	68	34	28	61	49	38	34	57	69	50	43	54	64	43	28	31	27	43	36	71	26	43	41	33	70	53
5	50	40	61	38	42	41	45	29	32	55	31	25	56	30	28	48	60	29	34	70	66	46	61	41	45	73	45	59	60	57	50	53	65	39	36	46	27	44	36	72	59	73
6	63	52	25	63	71	74	61	68	36	39	51	37	56	56	39	59	72	74	68	49	25	51	73	60	38	62	44	37	44	57	62	31	72	34	60	58	37	56	42	59	60	51
7	28	37	64	37	30	65	39	45	40	74	71	55	64	69	57	48	37	53	44	70	39	44	41	31	56	35	34	32	51	54	32	63	53	71	38	39	27	57	65	53	54	63

k=	5																				
j=	1		2		3																
h=	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7					
1	25	59	41	67	37	69	69	53	65	35	42	43	72	73	44	64	73	40	47	62	49
2	43	67	70	32	26	65	68	37	54	52	70	50	66	43	32	46	57	67	27	59	65
3	57	42	40	64	45	34	67	69	37	47	65	50	48	32	49	47	53	47	65	27	40
4	41	70	59	70	66	25	64	69	57	53	47	68	37	48	63	58	32	31	33	69	49
5	30	26	53	30	74	43	41	71	36	50	31	74	52	49	37	53	70	35	56	44	52
6	28	46	51	33	43	41	31	48	28	50	61	39	66	64	60	41	57	53	32	73	42
7	29	69	41	45	32	58	57	46	68	44	70	60	56	67	62	60	51	31	72	61	35

Tiempos de Procesamiento															
k=	1			2			3			4			5		
j=	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
h=															
1	108	115	89	125	102	111	117	118	115	119	95	120	101	94	84
2	77	95	96	78	116	91	124	79	83	98	75	92	98	122	86
3	123	114	125	76	76	105	99	122	120	93	89	117	78	93	88
4	100	95	112	100	84	92	89	104	102	83	115	87	86	121	122
5	77	120	87	97	111	94	101	96	123	85	113	123	96	93	105
6	122	107	75	96	94	99	75	88	112	85	104	89	85	123	100
7	121	112	91	122	78	123	120	82	121	109	0	85	101	95	80

**Tabla 35.** Tiempos de procesamiento y alistamiento de tareas para la instancia i7j3k5-2.

Tiempos de Alistamiento																																																
k=	1														2																																	
j=	1							2							3						1						2						3															
h=	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1	69	41	49	56	41	48	46	31	68	48	30	71	68	55	60	63	64	38	62	62	72	44	63	53	60	73	65	49	41	65	25	64	49	50	49	52	26	55	50	26	30	44						
2	74	31	62	41	39	66	27	34	29	33	30	50	32	44	58	50	70	31	49	70	31	31	44	36	49	54	65	26	69	60	28	59	52	38	37	66	74	25	44	60	59	28						
3	67	39	34	62	35	72	66	62	27	53	30	72	60	38	41	46	40	69	55	37	32	65	38	32	34	28	29	53	30	28	37	41	74	70	61	65	60	37	33	33	74	52						
4	56	70	66	72	33	58	51	47	26	51	60	46	29	61	70	36	73	45	58	46	31	64	51	50	67	37	31	36	42	66	73	33	52	57	63	69	28	37	47	59	38	46						
5	57	47	68	74	58	52	30	30	40	37	44	56	28	36	26	33	59	63	33	64	31	72	72	69	37	73	50	41	36	34	71	72	54	58	36	73	72	55	65	60	74	66						
6	58	40	29	41	41	45	53	35	59	66	61	35	44	74	70	50	35	25	34	66	41	62	70	45	29	52	30	29	50	29	52	70	66	62	56	67	38	41	53	61	50	45						
7	26	69	69	68	51	54	50	48	36	29	37	55	45	59	52	36	74	27	72	62	63	56	28	69	37	55	33	69	53	42	40	56	70	46	72	28	44	36	31	65	56	70						

k=	3														4																																	
j=	1							2							3						1						2						3															
h=	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1	59	30	42	27	59	39	52	49	27	61	51	35	55	51	67	35	36	47	48	58	53	45	25	47	59	43	46	45	41	45	54	33	57	52	46	68	54	49	26	70	52	41						
2	36	70	34	31	64	52	40	54	40	55	41	45	60	50	38	54	59	74	61	30	54	66	46	47	34	27	73	46	28	38	39	55	62	58	33	73	32	28	42	49	48	53						
3	53	47	35	60	36	60	50	53	72	27	34	39	34	30	51	69	68	56	66	41	50	52	63	50	31	47	59	29	71	72	48	25	48	35	27	65	69	47	31	34	40	69						
4	27	56	56	27	49	62	49	48	29	41	55	58	58	70	68	35	62	48	36	28	51	28	34	49	31	51	28	41	45	34	72	63	49	26	54	46	68	62	32	40	40	34						
5	60	60	29	64	27	51	55	46	66	44	68	50	59	37	32	56	50	53	52	52	42	36	68	62	34	59	42	30	74	52	60	46	25	25	58	27	31	60	53	45	63	58						
6	44	52	47	34	62	74	42	67	66	41	64	68	71	58	33	73	48	56	71	73	64	54	61	46	37	52	35	54	43	26	51	31	38	67	68	25	32	27	58	45	73	60						
7	28	48	44	36	66	30	36	25	45	42	26	41	46	49	58	67	49	49	69	56	44	35	55	30	45	56	35	68	44	61	72	30	35	67	41	51	70	69	40	57	59	25						

k=	5																							
j=	1							2							3									
h=	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1	64	62	31	27	56	54	74	73	45	41	59	47	53	42	53	53	49	38	26	48	60			
2	65	63	74	42	44	36	66	48	62	27	65	58	64	45	59	53	54	42	49	62	72			
3	62	69	35	41	34	67	59	38	71	26	27	50	66	36	61	38	63	54	45	48	27			
4	45	56	52	60	58	68	49	44	42	67	63	58	70	49	54	35	60	67	51	39	38			
5	25	38	36	34	66	54	52	59	51	71	52	34	69	39	70	56	40	44	54	54	55			
6	51	39	48	71	43	57	40	72	70	49	64	34	71	74	29	30	50	53	62	50	60			
7	55	55	37	62	25	43	47	56	63	38	31	66	34	32	26	65	72	54	68	46	50			

Tiempos de Procesamiento																
k=	1			2			3			4			5			
j=	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	
h=	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	
1	119	88	88	124	90	123	98	111	110	108	93	76	107	112	76	
2	102	105	118	113	76	102	123	84	75	107	84	91	101	121	95	
3	120	86	88	78	110	83	104	86	125	76	120	124	92	110	75	
4	77	77	119	93	113	114	76	120	114	122	98	83	100	88	94	
5	117	87	104	83	79	102	83	92	76	92	104	123	77	80	92	
6	95	102	78	90	98	77	89	82	90	84	123	94	77	118	88	
7	111	83	84	124	82	84	79	108	83	106	119	111	111	119	112	

**Tabla 36.** Tiempos de procesamiento y alistamiento de tareas para la instancia i7j3k5-3.

Tiempos de Alistamiento																																										
k=	1							2																																		
j=	1			2				3				1			2				3																							
h=	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7		
1	72	36	62	57	60	47	30	49	43	27	46	63	32	54	28	72	41	55	67	28	43	72	27	28	30	44	69	62	27	43	26	29	28	40	56	40	51	65	52	46	64	33
2	34	43	34	55	67	54	55	39	34	37	33	34	33	49	26	26	70	59	54	67	57	69	26	45	68	73	55	44	53	34	74	30	26	40	51	63	69	50	44	40	57	63
3	55	31	63	40	62	63	50	59	39	66	39	61	67	34	46	60	28	56	56	57	65	60	57	40	53	29	51	31	38	47	66	48	30	44	54	60	55	29	33	61	66	71
4	42	29	49	71	68	29	30	55	54	35	51	61	72	74	44	29	40	67	74	30	32	31	44	70	37	39	68	66	35	49	55	73	52	69	41	54	44	38	47	52	50	30
5	44	27	57	70	54	36	66	71	30	53	26	56	27	54	50	39	46	73	25	71	39	35	39	37	55	27	68	68	60	39	46	70	56	34	66	33	28	56	25	50	51	44
6	54	29	43	30	63	73	63	54	27	65	51	53	45	61	43	42	51	66	58	27	43	60	48	68	61	47	60	74	43	57	53	54	51	69	35	32	61	58	52	42	69	73
7	63	31	46	56	44	59	58	25	28	26	50	62	37	29	33	73	57	29	61	51	43	61	27	70	46	68	27	29	48	65	38	25	31	69	29	39	69	62	53	26	71	60

k=	3							4																																		
j=	1			2				3				1			2				3																							
h=	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7		
1	56	28	27	68	61	52	37	37	50	53	27	30	50	48	33	34	32	59	45	48	51	30	40	58	66	43	51	46	29	50	32	73	27	29	67	34	30	59	41	35	37	74
2	67	70	33	58	61	68	64	65	27	70	50	39	53	36	68	31	25	69	72	37	53	47	53	63	38	43	68	40	36	74	65	38	59	34	26	61	64	41	57	45	30	48
3	56	61	55	49	43	74	32	63	66	53	70	52	45	66	57	69	54	46	69	49	43	48	73	66	42	31	37	73	54	48	68	46	42	53	73	64	38	72	51	72	36	40
4	64	30	64	29	54	31	42	30	30	31	36	28	68	74	65	45	72	61	74	51	47	36	58	31	63	64	47	35	33	52	55	63	41	43	58	42	46	55	25	35	64	60
5	53	70	37	59	39	32	63	60	47	71	74	42	37	57	58	72	25	63	53	52	35	66	27	61	42	65	71	39	60	70	60	39	50	39	46	48	42	54	72	33	68	59
6	37	43	45	74	35	64	54	35	37	67	33	54	49	36	47	60	26	74	41	45	68	69	57	58	41	55	52	52	45	50	39	25	33	28	70	49	73	46	74	53	31	63
7	30	39	64	37	50	44	44	63	32	64	63	39	33	62	32	45	29	55	72	33	66	64	56	50	71	68	69	51	25	71	42	28	74	27	56	49	41	63	56	49	56	67

k=	5																							
j=	1			2				3																
h=	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7								
1	27	61	65	50	56	50	34	60	65	52	55	68	48	61	58	41	28	70	63	71	35			
2	47	65	31	57	71	50	38	35	41	63	43	54	48	44	34	35	40	46	73	49	51			
3	40	69	51	58	38	30	65	40	42	67	47	62	44	44	55	34	67	41	64	35	55			
4	44	59	69	37	37	44	63	43	25	38	34	69	47	37	55	71	66	26	53	34	28			
5	33	48	33	59	38	50	39	62	35	37	51	74	47	54	38	45	68	65	50	51	50			
6	66	43	73	34	70	64	70	45	25	58	33	74	55	38	32	56	34	63	41	32	66			
7	37	43	57	36	39	58	37	73	51	45	39	36	40	64	39	67	30	44	29	28	59			

Tiempos de Procesamiento																
k=	1			2			3			4			5			
j=	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	
h=																
1	122	96	82	94	109	119	79	78	102	77	109	105	109	95	109	
2	79	119	100	116	78	75	83	87	123	114	109	89	90	100	103	
3	90	83	82	99	80	75	108	124	99	125	99	113	100	115	116	122
4	108	113	122	103	121	94	76	81	95	105	115	90	119	105	120	
5	117	120	123	89	79	103	102	97	105	94	117	120	90	83	86	
6	102	112	84	75	84	123	76	101	96	91	87	92	110	101	90	
7	78	122	124	83	97	81	105	101	84	125	92	90	90	112	125	

**Tabla 37.** Tiempos de procesamiento y alistamiento de tareas para la instancia i9j2k3-1.

Tiempos de Alistamiento																																								
k=	1										2										2																			
j=	1										2										1										2									
h=	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
1	29	64	73	33	59	67	52	53	61	72	54	40	25	63	68	52	67	49	46	41	37	43	63	73	68	71	70	64	45	71	35	61	37	61	31	30				
2	45	58	71	28	42	62	60	41	66	31	48	65	32	66	58	41	53	53	45	69	31	36	62	73	67	42	28	41	33	62	40	55	74	58	52	46				
3	71	33	27	28	57	53	43	36	36	53	34	68	60	60	70	26	57	61	46	56	68	45	66	45	60	66	33	26	65	61	65	62	32	62	54	72				
4	66	45	60	72	44	70	62	71	32	65	66	29	57	71	44	46	57	52	29	50	72	35	63	39	33	38	38	63	68	30	56	25	48	71	29	68				
5	57	66	69	54	32	32	54	68	55	40	65	47	29	48	56	47	62	39	40	31	71	28	64	53	38	59	47	38	71	72	73	70	35	57	54	36				
6	69	44	25	55	61	36	42	57	59	30	70	31	60	72	71	55	27	60	68	35	36	52	38	49	42	38	34	69	52	49	39	62	37	71	47	42				
7	44	39	51	51	71	29	57	25	46	59	62	50	58	52	69	30	64	46	54	54	63	31	45	28	56	73	74	64	33	56	40	60	57	25	55	59				
8	27	52	66	57	73	40	54	61	49	37	58	65	37	58	57	43	46	34	67	39	39	53	71	36	60	72	57	29	43	28	65	32	72	70	48	71				
9	44	49	62	69	49	66	28	46	28	52	38	60	33	55	38	64	52	71	49	59	68	47	57	53	66	53	38	50	58	26	28	56	41	73	65	65				

k=	3																																					
j=	1										2																											
h=	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9																		
1	37	34	67	52	72	25	62	28	30	74	63	72	46	38	66	59	34																					
2	51	37	56	36	35	50	57	29	51	51	71	49	68	62	68	35	44	38																				
3	69	45	28	60	57	48	64	47	53	53	70	27	67	56	33	30	44	37																				
4	70	50	52	44	52	36	70	56	66	32	38	57	32	27	55	32	51	71																				
5	51	47	40	67	69	32	39	63	74	40	71	34	72	60	26	52	65	68																				
6	74	62	46	62	28	68	59	34	47	72	45	45	65	43	45	32	43	71																				
7	37	44	42	45	60	69	56	54	25	25	31	28	49	33	55	25	60	74																				
8	68	41	43	47	56	73	72	31	47	74	45	27	58	40	51	33	49	41																				
9	55	45	49	65	73	62	44	54	34	42	44	68	51	65	27	54	34	65																				

Tiempos de Procesamiento						
k=	1		2		3	
j=	1	2	1	2	1	2
1	108	97	119	107	105	108
2	98	125	84	101	75	92
3	116	115	110	98	119	125
4	81	116	109	86	75	91
5	106	123	87	111	100	98
6	98	98	91	119	93	103
7	116	83	98	92	92	118
8	107	88	119	105	77	123
9	103	105	109	106	124	83

**Tabla 38.** Tiempos de procesamiento y alistamiento de tareas para la instancia i9j2k3-2.

		Tiempos de Alistamiento																																							
k=		1									2									1									2												
j=		1									2									1									2												
h=		0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
1		46	67	27	29	69	31	34	55	25	52	47	70	44	31	42	65	40	31	32	32	50	53	73	27	26	47	30	70	33	57	62	45	45	66	44	43				
2		66	31	57	36	43	64	27	30	36	50	40	51	33	58	49	30	41	44	32	47	59	43	58	72	62	49	45	31	59	41	36	65	26	38	28	33				
3		25	60	25	70	49	63	50	45	43	49	53	45	27	54	47	28	49	51	56	57	45	31	53	50	30	36	63	46	73	73	30	44	48	53	52	57				
4		46	43	29	36	71	56	35	34	28	64	70	65	57	52	52	32	50	44	55	44	69	60	40	46	56	57	63	67	63	51	58	73	34	54	67	66				
5		40	63	28	45	55	64	26	56	72	28	32	43	57	68	74	33	48	49	48	56	49	25	67	48	26	44	72	43	63	33	39	53	30	58	46	41				
6		28	44	45	29	63	47	74	58	25	66	70	62	51	74	32	52	33	26	69	47	31	52	61	27	38	71	49	69	61	28	35	72	67	53	26	28				
7		30	30	57	30	66	31	57	33	48	37	74	35	68	54	29	25	33	35	66	74	45	56	66	61	54	69	30	36	37	65	57	58	55	31	65	27				
8		31	29	39	59	62	62	57	73	70	48	55	59	32	49	60	42	30	71	25	38	25	40	72	34	26	46	34	45	31	36	48	71	72	46	69	34				
9		39	65	73	55	39	32	42	38	46	47	48	46	28	39	45	41	62	64	38	25	31	70	72	48	46	65	59	46	62	34	32	25	49	49	63	38				

k=		3																			
j=		1									2										
h=		0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
1		54	67	31	71	59	70	51	29	27	67	60	41	68	71	72	29	57	42		
2		64	37	34	43	48	59	57	44	39	65	43	25	29	32	64	41	64	32		
3		30	26	65	27	47	28	43	34	33	69	41	54	50	25	34	40	51	62		
4		70	74	37	69	26	25	54	69	65	34	60	30	34	49	40	50	43	70		
5		51	57	25	49	62	58	65	70	65	42	25	60	28	27	49	65	72	73		
6		54	26	25	69	52	27	67	73	34	60	36	28	49	49	52	39	31	38		
7		72	42	69	64	28	31	41	58	28	33	30	30	52	41	28	38	35	40		
8		41	71	73	55	56	41	49	67	42	59	51	69	36	70	30	40	33	71		
9		33	34	73	67	66	62	29	58	44	74	69	36	66	53	25	54	39	43		

		Tiempos de Procesamiento					
k=		1		2		3	
j=		1	2	1	2	1	2
h=							
1		87	89	102	95	85	79
2		113	110	75	121	101	110
3		111	87	124	92	102	88
4		85	109	75	78	101	117
5		105	122	125	85	81	82
6		81	82	100	124	85	100
7		122	83	109	111	90	89
8		79	95	76	121	110	113
9		97	114	95	108	109	83

**Tabla 39.** Tiempos de procesamiento y alistamiento de tareas para la instancia i9j3k3-1.

Tiempos de Alistamiento																																																						
k=	1																		2																																			
j=	1						2						3						1						2						3																							
h=	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9				
1	38	59	74	26	51	70	31	35	61	69	62	60	50	67	40	50	51	65	26	30	71	70	56	61	66	33	63	34	35	42	55	26	47	43	73	41	28	42	73	27	54	38	72	65	49	39	73	49	33	55	74	50	64	40
2	25	26	30	57	64	53	56	32	36	57	72	74	54	48	72	44	63	44	28	25	26	50	71	35	60	41	40	58	61	57	39	44	26	60	70	44	68	44	72	56	58	48	61	72	25	40	57	36	26	35	52	49	67	64
3	28	74	51	28	62	30	58	48	34	35	56	71	65	25	71	74	66	62	29	39	47	35	67	50	73	49	34	68	45	59	40	41	67	73	36	36	52	59	67	39	45	40	43	70	53	61	39	63	25	31	33	45	36	41
4	42	47	42	54	69	62	64	62	53	73	28	51	55	49	56	64	66	45	71	32	26	46	30	49	41	44	48	35	47	38	59	37	56	35	68	56	46	41	70	50	32	56	56	68	62	54	56	35	48	54	71	53	48	54
5	27	30	66	33	30	51	63	70	40	67	60	61	45	41	44	32	61	46	56	34	48	55	25	74	56	50	36	42	27	35	35	46	42	59	33	65	55	53	28	43	32	41	51	69	57	44	40	62	71	50	33	48	73	72
6	33	31	55	45	27	60	27	36	53	62	42	47	53	52	26	45	45	58	54	60	63	39	63	68	45	52	32	46	38	61	30	41	35	59	49	35	47	67	73	71	33	45	25	57	48	66	56	56	32	29	70	67	43	
7	25	70	26	45	65	61	26	44	47	74	45	25	64	53	58	54	54	60	37	40	32	68	71	30	47	27	34	60	35	46	39	62	56	68	38	34	45	70	31	50	42	46	51	52	73	48	44	31	54	72	34	31	42	64
8	29	71	43	25	54	57	38	41	39	52	52	53	35	54	56	36	68	26	42	57	30	57	43	37	61	47	49	58	33	29	40	31	56	60	44	26	68	65	42	27	68	43	31	35	74	31	31	66	50	63	71	59	51	40
9	39	57	60	45	59	29	43	73	32	28	55	35	57	66	39	39	54	74	41	45	66	64	54	47	37	56	32	28	32	48	63	69	34	43	58	27	38	74	65	43	35	71	29	28	38	50	59	51	50	73	49	67	45	69

k=	3																													
j=	1						2						3																	
h=	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
1	74	28	52	38	33	52	47	62	27	42	26	59	42	67	26	31	42	39	55	29	40	62	67	47	52	59	29			
2	65	58	61	53	56	73	37	39	70	58	58	26	40	69	32	69	66	35	68	37	56	35	30	25	71	30	63			
3	58	68	25	30	31	25	44	66	56	52	63	25	38	25	63	52	28	61	46	57	26	53	47	50	27	41	63			
4	31	51	55	32	42	34	46	46	34	49	44	39	55	53	71	29	64	59	61	27	69	44	66	73	37	43	59			
5	56	38	48	28	33	35	36	63	64	30	74	36	52	55	45	35	47	58	69	56	26	53	72	57	54	72	42			
6	33	34	53	38	67	66	44	36	65	32	38	72	59	35	46	38	58	48	66	58	45	48	30	27	69	27	36			
7	39	36	58	70	44	33	67	32	67	45	31	52	35	29	29	62	52	34	49	46	25	50	45	68	60	56	69			
8	64	35	35	53	39	74	66	58	55	59	70	60	68	60	63	52	51	63	62	54	47	40	33	64	32	64	36			
9	43	69	73	71	44	73	33	37	67	36	66	54	42	70	73	48	73	30	62	47	48	59	36	31	29	48	31			

Tiempos de Procesamiento									
k=	1			2			3		
j=	1	2	3	1	2	3	1	2	3
1	110	119	87	125	98	82	90	111	83
2	108	123	80	123	110	121	77	80	101
3	112	117	79	86	116	100	76	75	106
4	107	76	123	123	90	122	119	118	84
5	101	122	121	98	101	88	97	87	106
6	122	104	105	97	101	120	100	87	104
7	94	96	109	117	83	85	79	117	114
8	78	95	80	121	116	121	106	86	76
9	106	116	78	81	94	120	9	101	82

**Tabla 40.** Tiempos de procesamiento y alistamiento de tareas para la instancia i9j3k3-2.

		Tiempos de Alistamiento																																																					
k=		1									2									3																																			
j=		1			2			3			1			2			3																																						
h=		0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9														
1		33	47	59	52	55	51	62	57	28	31	68	74	59	51	68	60	65	65	53	46	29	42	34	43	49	65	64	34	33	68	66	64	58	64	33	49	73	57	27	34	40	51	33	29	56	26	32	43	59	45	64	28	56	42
2		32	26	73	47	38	43	65	74	35	69	71	27	55	31	35	53	32	25	28	48	32	38	34	28	47	35	65	73	41	71	53	53	66	67	35	32	50	64	72	38	70	47	25	70	61	66	67	60	74	62	69	36	57	65
3		56	25	61	35	26	59	25	57	54	33	26	40	45	38	42	47	45	46	73	51	48	63	66	66	35	68	36	48	25	30	41	63	67	26	25	74	28	53	48	61	48	25	64	36	42	45	73	34	70	50	53	60	34	41
4		71	41	58	67	28	66	61	49	34	33	27	70	52	58	59	31	47	66	49	71	53	35	59	37	32	46	44	56	66	44	65	61	42	68	40	49	68	29	39	44	42	40	55	65	67	49	61	25	25	51	68	36	53	27
5		47	32	59	41	25	54	44	54	72	57	45	36	73	53	64	42	30	53	54	47	63	58	38	32	33	63	56	64	47	72	50	74	30	33	45	49	65	74	32	51	40	25	58	55	59	46	50	61	49	35	43	48	54	27
6		40	30	58	26	68	43	68	74	65	40	40	27	56	40	50	59	33	34	58	44	29	68	64	73	58	59	45	45	56	33	46	51	38	61	48	46	35	54	41	58	70	32	47	26	53	74	25	56	33	47	58	46	36	65
7		35	53	72	33	62	37	73	60	49	33	27	47	48	40	42	28	72	50	31	38	47	60	40	35	29	30	39	44	67	49	66	28	62	36	48	59	38	57	51	48	51	39	73	28	60	58	44	66	44	66	42	67	66	47
8		42	58	36	47	64	66	51	44	53	62	58	63	69	45	26	25	27	72	41	45	68	67	67	67	26	72	59	26	44	44	27	35	25	32	27	44	67	39	62	31	53	25	26	39	38	49	53	32	30	64	50	45	55	
9		46	31	31	71	33	39	36	61	70	29	56	26	70	41	41	38	60	46	35	59	61	32	72	37	70	39	34	70	52	33	29	52	26	60	74	72	74	72	68	65	30	65	66	53	46	62	63	40	27	42	57	44	41	27

k=		3																										
j=		1			2			3																				
h=		0	1	2	3	4	5	6	7	8	9																	
1		57	69	51	32	45	42	62	58	69	52	47	47	44	37	54	33	44	35	33	60	39	59	28	37	38	63	66
2		42	61	39	65	60	58	44	62	57	51	46	64	56	56	42	72	51	71	52	29	46	45	49	31	48	49	66
3		51	32	51	53	47	50	71	74	48	45	35	30	60	30	45	54	64	35	48	32	27	44	34	56	36	54	33
4		53	44	57	53	54	33	66	45	39	36	31	45	69	53	35	41	58	44	37	52	60	73	31	74	34	52	43
5		67	34	57	40	67	72	28	67	58	47	73	49	28	60	45	53	62	62	34	28	65	29	44	34	56	69	45
6		32	53	31	40	51	70	69	29	66	47	61	42	44	55	67	48	28	56	41	74	45	61	31	54	46	27	53
7		65	47	60	28	59	32	60	74	71	26	61	72	44	64	38	29	36	70	71	28	73	74	53	32	58	61	63
8		68	59	34	35	56	46	37	61	66	34	55	43	30	33	73	29	37	72	68	31	70	36	69	74	70	46	56
9		48	65	47	72	63	47	27	40	28	74	52	59	71	66	65	55	35	30	27	37	66	72	33	55	69	50	40

		Tiempos de Procesamiento								
k=		1			2			3		
j=		1	2	3	1	2	3	1	2	3
1		124	86	115	90	117	115	110	78	88
2		98	93	100	100	75	107	119	81	77
3		113	114	89	104	86	80	88	87	87
4		106	102	113	98	111	115	100	99	77
5		118	113	110	76	85	109	90	100	77
6		116	115	115	97	122	104	83	98	111
7		92	119	110	109	80	98	98	85	106
8		90	76	96	123	80	120	98	90	89
9		87	104	118	84	102	80	98	113	78

**Tabla 41.** Tiempos de procesamiento y alistamiento de tareas para la instancia i9j3k3-3.

Tiempos de Alistamiento																																																																																																			
k=	1																		2																																																																																
j=	1						2						3						1						2						3																																																																				
h=	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9																																							
1	35	40	54	25	37	53	43	51	50	40	26	39	42	38	68	50	66	45	61	49	61	31	35	37	33	35	38	32	60	41	36	41	67	69	36	65	44	56	36	41	45	37	69	73	39	55	65	65	40	38	57	60	41	41																																													
2	41	28	51	52	37	53	41	50	69	38	49	40	25	59	44	66	63	40	59	62	47	25	66	55	60	55	52	44	70	73	60	64	53	43	57	41	29	64	69	50	31	31	62	65	40	70	39	71	42	35	34	54	46	37																																													
3	69	61	50	36	70	66	60	70	39	57	69	47	27	66	35	63	40	63	46	27	47	50	46	32	68	61	38	46	31	65	70	29	44	61	62	55	35	72	31	48	61	70	62	63	64	66	42	57	42	30	30	50	65	39																																													
4	74	33	42	31	48	58	66	44	49	41	45	43	42	61	68	69	48	67	39	42	36	53	45	39	41	28	56	50	35	71	56	44	29	38	58	27	38	74	25	44	58	54	43	28	34	54	65	38	47	36	25	57	34	68																																													
5	60	40	35	32	73	67	26	62	36	69	71	67	72	25	72	60	32	46	28	27	27	62	39	34	36	45	70	69	46	72	64	43	67	54	65	59	46	41	32	56	39	41	51	55	59	42	66	40	39	32	31	47	32	57																																													
6	51	46	52	62	27	27	61	67	42	56	25	47	29	72	30	48	39	50	46	52	56	33	64	33	53	45	72	35	73	69	40	40	26	63	34	59	50	25	32	41	49	30	27	62	71	27	36	35	29	28	34	29	47	34																																													
7	43	35	56	33	71	25	71	31	26	37	28	25	58	64	72	30	52	59	37	74	33	64	25	64	34	65	27	63	34	51	42	35	49	41	35	54	68	65	52	59	42	36	61	73	69	62	27	40	65	51	36	42	51	55																																													
8	48	29	72	49	62	70	49	27	74	40	57	54	29	54	74	39	26	35	30	25	43	39	51	28	51	48	36	48	50	53	39	57	42	41	46	60	44	27	26	66	58	34	32	68	50	40	57	35	54	28	68	74	25	39																																													
9	28	32	49	39	68	66	65	60	41	40	64	49	58	32	66	61	74	34	63	53	61	63	45	44	51	72	53	61	28	48	31	62	40	55	45	50	26	53	40	69	39	49	31	61	40	53	72	56	73	54	37	68	53	70																																													

k=	3																													
j=	1						2						3																	
h=	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
1	28	59	25	58	70	53	27	70	54	35	41	27	40	73	46	29	67	32	37	63	56	28	33	74	49	53	34			
2	65	72	60	59	55	34	72	34	40	34	37	46	38	72	42	54	35	72	32	43	43	30	49	66	29	70	31			
3	64	69	61	37	63	55	74	67	56	68	74	38	42	48	27	64	56	27	63	58	68	46	68	39	30	47	26			
4	48	41	60	30	47	70	62	61	33	69	51	40	68	54	34	73	63	59	53	25	61	53	46	38	27	71	64			
5	52	30	56	41	55	60	34	73	42	57	39	32	32	74	52	59	61	49	29	54	48	74	62	63	26	35	64			
6	44	30	33	28	72	27	43	41	57	52	61	66	39	61	74	74	67	40	44	46	52	48	53	45	52	43	55			
7	28	45	66	41	65	40	70	32	74	59	58	32	39	62	71	46	60	68	28	47	56	35	50	60	29	28	38			
8	67	47	64	35	74	48	27	68	59	42	59	26	25	56	57	54	52	27	25	65	26	69	60	46	49	50	32			
9	40	60	66	69	33	71	38	53	49	35	54	58	67	57	47	65	66	34	62	66	28	45	58	37	72	51	40			

Tiempos de Procesamiento									
k=	1			2			3		
j=	1	2	3	1	2	3	1	2	3
1	82	109	76	113	101	103	100	115	109
2	103	113	89	122	104	75	121	91	79
3	106	114	77	90	99	75	123	82	78
4	101	125	106	75	119	124	125	120	95
5	123	75	123	82	118	89	116	114	77
6	101	75	93	110	107	108	100	95	108
7	125	113	90	88	98	120	75	107	94
8	97	101	114	99	96	112	89	110	93
9	101	89	121	94	122	107	93	86	76

Tabla 42. Tiempos de procesamiento y alistamiento de tareas para la instancia i9j2k5-1.

Tiempos de Alistamiento																																				
k=	1										2																									
j=	1					2					1				2																					
h=	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9						
1	72	26	59	43	46	36	36	61	29	61	73	50	45	43	44	33	64	28	63	28	65	58	55	59	25	53	59	29	54	66	47	25	52	67	38	53
2	48	26	30	62	31	39	55	51	57	72	56	26	45	68	58	37	45	54	70	60	63	64	37	36	30	39	35	29	31	58	29	68	46	57	70	34
3	43	47	66	56	54	49	30	47	50	28	33	66	56	32	64	29	72	33	45	47	68	52	30	62	50	48	57	44	72	31	44	52	31	49	68	60
4	66	42	67	40	25	39	61	54	68	42	51	29	36	51	46	49	41	63	58	29	55	46	52	54	68	49	36	27	37	49	43	37	58	29	68	57
5	63	43	30	51	29	51	51	25	60	42	48	33	66	33	50	34	27	57	25	39	69	42	72	31	40	57	45	72	52	47	56	40	63	39	29	47
6	55	62	30	50	64	71	45	33	27	67	68	61	30	66	30	52	58	60	32	31	66	41	64	73	74	50	50	69	52	41	35	56	58	39	33	54
7	36	41	29	33	66	46	67	38	66	60	63	28	40	33	64	68	26	41	25	34	70	48	59	65	67	36	55	47	46	45	28	29	59	67	39	45
8	57	29	29	69	57	69	46	72	37	52	59	42	74	37	41	72	39	43	70	36	45	43	29	31	37	66	51	50	49	32	37	69	34	46	38	50
9	74	65	30	36	53	30	47	73	62	43	55	74	52	49	56	50	38	72	54	37	56	25	30	48	46	50	57	30	44	41	34	47	38	69	51	73

k=	3										4																									
j=	1					2					1				2																					
h=	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9						
1	29	25	46	48	43	25	33	62	39	63	39	57	42	34	69	52	40	60	36	28	52	43	33	28	51	28	68	47	43	34	47	25	40	35	69	58
2	29	64	57	50	44	74	51	26	72	61	47	31	39	48	28	43	43	61	42	45	62	28	42	41	65	50	26	45	36	71	57	66	34	61	64	70
3	31	60	49	71	46	28	29	63	38	35	57	56	34	50	56	62	73	60	52	56	34	28	54	74	47	68	38	33	26	71	67	67	47	40	56	60
4	36	52	60	61	55	67	66	41	25	66	73	48	52	48	72	46	37	71	67	55	33	73	26	46	50	60	40	74	39	46	48	56	70	56	27	47
5	71	53	41	71	25	36	51	42	71	38	69	52	33	71	62	48	41	68	65	31	48	71	45	69	71	32	73	26	32	41	58	56	47	57	49	29
6	28	34	35	26	31	31	56	41	48	34	59	43	37	36	28	62	29	26	55	34	36	52	31	29	33	63	26	58	25	59	29	62	44	73	65	40
7	67	63	52	64	59	38	70	55	65	29	55	38	59	47	50	51	32	40	63	58	69	74	31	25	49	60	44	32	30	45	39	30	33	42	48	48
8	46	45	46	29	41	69	43	37	45	31	52	31	31	56	43	36	59	72	53	43	39	32	48	27	64	37	66	59	46	66	65	33	47	48	38	74
9	47	72	71	52	56	48	47	49	53	59	42	63	51	55	36	45	62	36	71	73	66	38	36	68	69	71	47	54	51	26	57	25	38	51	58	74

k=	5																			
j=	1					2														
h=	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
1	27	49	60	66	65	51	74	32	39	32	69	41	39	53	51	37	46	50		
2	36	60	38	40	34	29	40	45	34	43	36	61	53	53	26	43	40	50		
3	54	28	59	56	62	73	44	63	59	26	25	71	51	68	52	29	72	25		
4	55	70	51	69	47	71	37	46	57	64	52	72	71	55	55	28	59	26		
5	63	48	37	71	70	68	56	65	39	40	32	65	48	40	44	42	41	57		
6	44	62	33	42	66	69	53	46	46	56	58	63	38	35	57	67	47	54		
7	48	58	35	68	44	41	62	36	43	44	44	56	74	61	39	50	37	48		
8	35	40	47	57	72	38	74	35	57	41	74	65	61	53	41	54	38	48		
9	61	42	41	62	66	51	62	53	66	70	53	52	52	31	57	40	27	70		

Tiempos de Procesamiento											
k=	1		2		3		4		5		
j=	1	2	1	2	1	2	1	2	1	2	
h=	1	2	1	2	1	2	1	2	1	2	
1	87	118	81	85	86	100	84	101	93	97	
2	79	81	114	116	111	90	121	87	114	97	
3	102	119	86	115	112	85	106	125	96	88	
4	111	97	87	78	105	96	110	97	103	81	
5	111	98	88	103	85	124	112	84	81	121	
6	119	87	123	94	81	99	88	117	88	94	
7	125	110	102	109	82	110	101	112	100	108	
8	106	104	109	125	119	100	125	78	87	88	
9	82	95	96	118	102	109	79	77	76	87	

**Tabla 43.** Tiempos de procesamiento y alistamiento de tareas para la instancia i9j2k5-2.

Tiempos de Alistamiento																																				
k=	1										2																									
j=	1					2					1				2																					
h=	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9						
1	27	70	59	49	40	42	66	43	59	70	59	53	58	36	26	41	43	26	39	37	71	55	65	41	69	37	40	60	26	46	61	62	74	27	42	57
2	49	30	39	62	68	26	55	46	47	65	63	51	45	62	40	34	62	65	51	26	71	65	42	33	68	28	47	36	51	33	63	71	29	27	63	36
3	36	64	60	32	68	35	47	37	40	68	49	40	31	69	60	33	43	47	70	56	66	59	33	73	54	48	40	67	37	53	26	34	25	42	62	46
4	29	65	68	55	33	69	42	48	63	73	71	31	32	55	71	50	51	62	45	34	41	45	42	59	62	28	29	47	26	42	72	38	36	36	73	55
5	72	33	29	65	57	69	35	34	60	66	44	52	31	66	57	62	57	27	62	28	73	37	44	36	55	38	48	32	48	31	58	37	28	70	62	74
6	39	66	55	48	71	65	65	27	60	56	35	53	58	52	46	51	69	30	55	40	51	62	27	27	27	40	33	72	72	50	54	29	57	30	45	
7	30	38	29	67	34	69	34	66	64	44	53	62	73	61	65	56	45	74	62	57	50	33	45	69	41	74	36	73	73	44	27	48	69	28	36	
8	64	43	55	69	48	32	54	42	29	64	34	34	65	63	49	52	33	45	29	32	71	29	73	41	40	71	57	63	50	69	41	29	33	38	40	74
9	34	74	31	26	60	66	33	46	29	53	31	46	62	57	55	29	33	63	74	50	69	53	34	38	71	33	66	67	28	54	34	27	54	37	54	60

Tiempos de Alistamiento																																								
k=	3										4																													
j=	1					2					1					2																								
h=	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
1	46	70	68	73	50	61	54	60	74	32	51	68	74	73	42	60	68	54	34	60	33	51	60	31	35	25	40	53	38	31	38	53	51	54	57	55				
2	26	68	28	49	51	36	70	51	61	26	32	47	30	26	48	65	57	33	59	43	31	33	28	57	42	73	45	64	62	36	73	39	34	50	52	69				
3	58	34	44	60	30	44	49	55	71	42	36	38	66	50	30	40	73	63	35	40	50	47	74	60	64	28	52	27	60	35	27	38	31	45	35	57				
4	58	49	37	49	51	64	48	64	29	69	25	63	58	46	64	70	69	54	74	47	66	41	43	44	50	67	72	43	25	55	73	35	47	68	71	39				
5	57	40	45	55	71	36	58	53	42	28	68	61	26	54	36	53	34	69	41	51	25	51	67	60	71	34	69	38	44	34	35	38	57	35	57	28				
6	41	49	37	58	72	64	70	43	59	30	66	64	63	38	74	42	33	35	49	65	55	50	70	63	48	66	29	61	57	71	25	51	71	64	51	54				
7	27	32	44	54	37	61	26	27	35	26	35	74	34	52	27	67	55	51	47	46	67	29	43	60	60	38	37	44	45	71	57	46	38	28	28	70				
8	72	46	60	31	36	34	71	54	43	34	35	26	56	36	58	41	69	54	32	31	58	67	46	41	46	41	74	28	45	57	60	61	66	66	70	62				
9	52	70	26	35	37	40	31	51	57	29	70	70	34	67	25	59	74	36	69	30	65	33	70	34	40	27	67	64	43	46	49	63	53	44	25	65				

Tiempos de Procesamiento										
k=	1		2		3		4		5	
j=	1	2	1	2	1	2	1	2	1	2
h=	1	2	1	2	1	2	1	2	1	2
1	89	96	106	81	116	100	121	88	99	121
2	113	89	116	103	112	118	81	117	93	84
3	97	107	101	108	83	110	88	110	91	93
4	95	84	117	98	80	112	107	94	75	105
5	95	83	98	108	102	90	116	92	76	104
6	102	78	111	75	123	105	81	75	82	87
7	112	117	110	99	110	120	90	105	88	81
8	124	103	100	101	111	103	108	81	79	105
9	96	113	102	97	104	114	78	99	115	123

Tabla 44. Tiempos de procesamiento y alistamiento de tareas para la instancia i9j3k5-1.

		Tiempos de Alistamiento																																																						
k=		1															2															3																								
j=		1					2					3					1					2					3																													
h=		0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9					
1		28	33	30	48	41	32	31	38	49	71	50	43	48	70	52	59	29	71	31	49	36	57	44	73	58	61	64	65	43	33	57	71	55	25	38	32	26	26	46	32	74	25	36	44	63	60	33	56	30	49	45	66	53	57	
2		64	30	44	49	41	28	71	49	61	49	31	62	28	34	74	55	72	69	72	63	65	33	68	28	38	53	62	42	73	26	66	55	71	71	63	70	68	28	28	70	39	49	74	46	30	74	44	58	28	71	31	53	30	32	
3		59	42	32	54	44	34	29	47	54	65	71	51	26	74	63	42	41	32	37	46	74	49	27	59	56	65	68	30	43	57	28	46	26	50	30	53	32	27	25	69	43	45	46	36	66	31	70	45	65	30	67	58	26	30	
4		31	73	68	57	42	74	35	35	28	72	26	74	68	40	32	33	55	32	39	31	65	70	65	62	49	42	43	44	43	71	43	51	46	58	65	65	43	73	55	58	59	54	28	67	31	66	67	63	34	51	56	28	33	29	
5		44	64	71	37	38	45	50	70	58	45	64	61	30	25	47	43	56	61	43	58	30	67	69	41	34	56	43	41	45	67	63	74	52	32	30	47	55	33	26	27	47	68	41	28	26	38	32	41	49	48	34	54	37	60	
6		61	65	31	52	34	34	61	62	61	70	45	28	29	45	57	63	34	60	57	71	33	68	51	43	63	52	67	59	70	36	63	46	52	69	37	70	49	74	66	34	48	71	64	26	25	32	38	59	58	65	60	36	64	74	
7		59	56	43	69	26	60	47	48	47	67	43	58	61	66	32	47	38	74	64	50	39	25	50	59	52	47	29	26	66	64	44	68	71	38	48	43	30	43	68	54	60	35	30	59	62	50	63	52	34	52	53	41	48	54	45
8		52	52	65	69	72	42	48	31	67	35	50	52	25	30	48	49	56	61	66	46	47	54	66	48	69	31	42	56	46	27	34	55	27	64	41	50	59	30	49	42	35	61	70	61	52	74	60	71	64	40	53	44	58	60	
9		64	61	73	27	59	51	46	37	65	73	69	72	45	63	35	30	29	39	34	55	29	65	42	51	66	33	36	51	39	62	46	38	62	39	43	59	29	42	59	46	40	44	36	29	64	30	37	32	54	32	67	39	66	63	

		Tiempos de Alistamiento																																																					
k=		1															2															3																							
j=		1					2					3					1					2					3																												
h=		0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9				
1		59	67	72	42	34	73	63	60	63	26	28	50	40	46	73	32	58	29	31	64	34	45	62	44	61	39	34	35	38	69	56	25	28	56	65	36	50	64	60	46	67	42	27	37	64	37	67	50	35	37	44	46	54	36
2		35	49	31	38	68	69	33	52	68	50	46	28	26	51	66	67	67	30	32	63	65	69	52	43	69	59	58	74	37	46	74	65	26	70	64	41	34	69	72	46	33	62	46	39	56	51	40	29	32	66	30	35	36	69
3		51	43	67	58	50	42	70	66	56	31	53	45	33	74	66	38	62	30	69	68	66	44	33	53	48	38	48	74	39	45	64	58	31	28	36	30	51	33	63	51	28	35	60	74	26	35	61	53	73	34	62	48	40	36
4		50	69	71	59	48	45	70	37	54	63	67	29	56	51	56	45	70	48	58	49	28	71	53	72	54	64	37	39	53	25	28	72	59	26	73	28	71	34	26	69	49	37	73	70	44	36	27	25	66	32	34	51	69	32
5		41	65	53	35	40	72	31	31	54	64	28	68	37	29	27	57	65	25	54	29	35	46	62	44	51	37	59	40	52	40	73	53	25	50	31	57	37	73	58	52	64	33	62	68	45	73	41	50	49	55	37	72	63	43
6		26	47	44	63	45	31	26	61	30	43	74	60	25	26	47	62	42	68	47	74	57	35	65	60	49	36	71	52	74	72	46	46	67	31	62	64	40	35	72	63	26	47	57	30	64	53	37	42	73	73	67	37	59	73
7		34	45	53	54	66	68	49	28	27	63	48	47	47	38	38	60	67	57	69	51	31	72	28	72	50	43	33	67	25	31	73	30	47	47	59	59	64	34	42	33	64	56	29	54	54	49	55	66	61	40	37	41	27	36
8		54	36	52	32	71	43	29	34	62	43	70	47	28	41	56	74	33	73	65	29	64	64	40	30	72	34	74	26	38	25	35	37	43	43	25	71	42	29	57	35	43	30	58	35	71	55	44	26	55	73	58	32	39	28
9		40	41	47	31	56	25	59	60	45	33	55	42	74	39	51	47	43	62	29	49	31	43	32	31	48	65	37	66	64	48	51	65	44	59	32	63	43	59	41	67	56	56	73	37	39	74	44	41	36	49	54	27	41	65

		Tiempos de Procesamiento														
k=		1			2			3			4			5		
j=		1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
1		78	79	117	107	88	82	78	104	112	88	107	76	124	104	117
2		111	76	88	107	93	120	89	111	96	107	98	78	114	105	75
3		78	101	112	121	77	115	111	87	102	77	87	118	80	81	117
4		93	120	120	89	114	90	84	120	96	118	117	77	83	107	109
5		78	81	116	108	11	76	106	107	83	123	103	113	85	98	124
6		115	118	91	109	115	75	117	111	97	83	104	113	76	112	113
7		88	106	86	85	101	123	106	111	117	120	89	76	106	99	88
8		80	104	118	91	94	100	103	121	77	77	108	81	104	97	103
9		115	101	81	121	87	117	120	82	75	123	104	120	110	108	117

Tabla 45. Tiempos de procesamiento y alistamiento de tareas para la instancia i9j3k5-2.

Tiempo de Alistamiento																																																							
k=	1									2																																													
j=	1									2									3																																				
h=	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9																									
1	53	38	28	47	52	33	41	61	50	39	59	36	42	26	26	26	54	26	49	66	53	52	25	39	26	46	64	51	46	41	59	45	35	58	43	36	37	41	68	43	52	38	74	52	56	67	28	31	74	58	51	74	48		
2	55	34	32	26	57	70	49	60	57	45	54	46	65	30	37	69	26	58	33	26	74	57	50	37	28	36	63	69	35	73	42	37	49	33	71	41	52	53	58	50	27	52	59	45	41	72	73	60	28	33	74	31	35	56	
3	52	40	53	55	35	34	68	54	73	50	65	28	26	54	45	71	38	46	35	37	43	37	60	74	26	31	34	65	51	72	52	66	74	69	58	51	69	26	62	56	54	69	72	65	65	58	31	29	63	51	39	49	45		
4	43	30	51	51	67	37	71	47	32	60	69	49	47	55	29	32	31	63	26	68	65	28	43	55	46	60	55	40	43	52	38	53	64	74	53	68	72	64	69	38	62	34	65	74	60	60	27	29	64	45	34	72	65	26	
5	37	26	25	33	27	58	26	39	73	72	59	44	49	39	25	73	72	55	38	71	49	38	45	70	34	34	40	50	28	52	33	33	47	47	51	53	32	39	38	70	68	64	32	66	30	54	25	47	54	38	29	55	72	32	
6	26	57	58	66	29	63	44	27	47	25	54	58	59	47	69	47	68	35	67	33	35	38	32	32	53	25	71	53	56	61	32	49	72	32	67	42	39	66	45	60	49	43	46	54	71	43	59	72	33	29	29	51	36	52	
7	60	32	45	67	48	25	35	44	45	71	52	73	54	72	70	38	63	39	37	73	47	67	40	59	32	39	26	28	62	61	31	53	36	63	40	29	25	65	62	71	45	52	70	43	34	61	62	59	44	52	48	29	61	72	
8	30	28	49	72	50	26	32	46	53	41	72	74	36	50	71	53	30	34	26	60	64	69	57	43	74	59	72	37	37	35	42	59	70	49	70	33	73	70	50	70	61	25	25	29	74	33	58	40	36	62	25	68	68	61	72
9	72	39	62	56	40	35	44	73	46	54	52	27	65	54	33	66	72	50	48	28	73	61	27	57	28	35	57	40	39	62	62	67	33	32	67	68	28	59	42	65	48	58	57	41	60	50	53	45	60	67	57	50	29	50	

Tiempo de Procesamiento																		
k=	1			2			3			4			5					
j=	1			2			3			1			2			3		
h=	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
1	104	97	93	118	108	108	101	107	86	106	125	86	103	123	103			
2	87	113	114	96	110	116	125	78	95	86	99	97	87	98	82			
3	114	114	109	99	100	123	112	102	115	78	75	88	89	84	78			
4	109	87	76	110	121	119	85	84	120	104	75	122	109	77	86			
5	115	81	109	118	92	87	98	120	78	96	85	87	96	121	106			
6	113	85	112	112	109	123	89	120	84	120	118	124	91	95	96			
7	102	101	115	96	123	93	98	102	85	87	90	105	95	118	116			
8	106	112	87	120	92	88	109	111	113	75	93	82	102	75	97			
9	115	95	79	88	79	111	84	123	96	103	98	86	93	106	77			

**Tabla 46.** Tiempos de procesamiento y alistamiento de tareas para la instancia i9j3k5-3.

Tiempos de Alistamiento																																																																																										
k=	1																														2																																																											
j=	1										2										3																																																																					
h=	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9																																																		
1	70	29	59	70	29	34	55	66	30	43	33	30	40	69	67	40	25	31	71	34	53	44	63	63	41	68	64	36	72	53	45	25	66	48	27	63	31	60	71	59	56	30	66	30	35	73	37	45	56	29	30	37	70	29																																				
2	59	63	51	67	53	56	30	73	36	73	65	49	29	57	67	73	73	55	43	74	36	27	35	74	49	32	41	63	70	64	29	38	39	32	55	49	72	42	33	53	32	71	27	48	54	62	70	64	44	53	54	71	47	54																																				
3	31	53	65	57	61	28	43	71	51	68	29	53	50	28	26	33	58	63	42	37	66	25	35	46	60	52	64	49	29	54	72	60	46	69	69	60	44	35	58	49	42	43	59	58	59	55	61	50	65	52	35	65	63	56																																				
4	55	50	65	70	28	71	35	33	32	29	64	47	68	57	35	53	39	32	33	55	54	61	54	26	55	60	37	59	29	34	26	71	51	25	30	34	69	64	27	68	29	38	40	71	60	69	64	69	65	50	29	50	53	43																																				
5	61	25	46	63	52	32	32	37	40	50	39	65	70	57	37	42	36	58	64	46	71	32	32	52	65	43	51	31	66	56	64	63	27	50	65	48	29	69	41	49	49	36	31	73	38	49	65	32	56	60	58	26	71	37																																				
6	29	44	31	35	72	73	32	64	41	60	53	26	42	45	35	55	59	34	40	40	71	46	34	51	42	67	73	64	73	54	70	29	37	57	45	36	45	40	69	71	33	55	59	73	41	26	30	27	38	66	56	55	66	32																																				
7	54	28	73	47	74	62	26	57	43	41	70	72	42	63	74	57	70	61	64	29	74	36	59	27	67	64	69	46	25	65	70	45	56	27	46	45	37	61	45	54	38	65	50	44	32	42	70	27	47	31	35	25	56	27																																				
8	42	63	57	38	46	37	41	36	38	37	48	51	53	37	38	49	61	37	49	32	28	68	56	65	44	50	59	28	34	41	33	27	63	69	40	46	44	40	33	45	32	63	39	61	38	31	46	55	57	69	54	57	34	35																																				
9	49	33	25	30	51	37	40	56	50	27	34	36	31	44	38	37	66	45	29	35	53	30	74	49	53	60	29	46	37	61	42	33	48	43	31	28	69	65	39	73	39	33	31	38	67	36	73	69	68	73	73	31	54	51																																				

k=	3																														4																													
j=	1										2										3																																							
h=	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9																				
1	72	64	45	72	34	50	59	56	30	55	54	70	25	71	32	55	56	70	58	61	74	71	31	73	41	33	71	46	39	26	39	36	55	36	33	28	50	54	48	33	30	33	72	32	32	33	32	69	33	51	56	26	47	26						
2	48	60	55	45	25	49	25	53	45	67	27	30	34	66	53	31	58	57	35	46	67	53	36	49	72	32	53	50	45	50	51	36	33	72	33	36	31	36	40	72	72	54	33	53	61	40	67	65	71	73	45	31	69	61						
3	57	74	55	25	54	39	70	51	28	43	35	70	50	38	37	52	47	34	35	55	59	26	33	43	73	66	71	63	63	42	34	42	61	27	55	59	33	28	47	56	66	70	52	58	35	64	63	31	43	45	33	69	45	54						
4	29	58	33	58	53	38	26	49	56	53	44	46	31	50	67	36	46	65	48	36	51	42	46	47	35	70	29	64	46	29	53	68	35	26	39	48	55	61	57	59	67	47	41	33	39	34	25	45	47	73	35	64	61	44						
5	42	41	39	58	68	44	30	28	38	49	49	41	35	43	55	56	35	61	62	54	66	64	50	45	33	36	72	54	34	29	65	35	57	59	34	29	36	54	38	40	27	27	43	40	40	69	39	69	73	32	26	48	65	67						
6	26	25	41	35	45	72	30	70	34	37	39	73	35	27	73	67	31	27	70	51	29	39	25	65	48	33	55	69	45	33	39	34	44	68	53	64	58	59	73	60	33	52	48	71	41	47	31	74	59	59	47	59	71	42						
7	61	64	68	43	72	59	68	60	56	30	32	29	50	51	40	52	62	55	71	72	38	62	63	34	73	64	64	67	26	57	28	38	35	41	74	26	41	70	53	34	41	31	47	34	69	56	68	49	71	31	69	44	58	26						
8	35	45	55	51	26	73	74	69	74	27	55	30	26	48	66	46	36	71	66	61	66	70	56	72	38	41	45	65	42	33	57	29	62	38	50	60	41	64	69	58	58	49	66	32	63	71	67	60	44	57	25	47	63	46						
9	72	31	37	65	42	58	63	37	44	70	47	63	39	30	64	45	35	49	41	55	62	59	58	61	67	63	54	72	40	40	66	40	36	64	50	43	49	62	45	69	28	54	52	64	58	38	35	32	47	65	43	53	67	55						

k=	5																													
j=	1										2										3									
h=	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
1	36	34	68	33	74	64	54	35	42	44	66	36	44	74	31	46	65	27	38	25	73	29	40	50	38	53	41			
2	48	30	44	74	64	50	72	57	38	36	52	74	49	56	31	34	70	59	61	43	63	67	53	52	25	43	59			
3	29	40	73	54	64	44	42	71	40	38	66	38	74	68	25	46	61	38	52	54	68	36	52	46	74	67	53			
4	39	47	35	35	34	41	53	27	55	63	46	35	44	54	37	39	48	57	72	41	66	49	41	26	66	60	74			
5	42	29	25	64	69	74	63	58	64	52	62	63	30	26	73	53	54	32	44	35	33	44	31	57	62	44	38			
6	48	38	59	30	57	29	57	67	73	28	64	46	48	39	72	38	66	32	31	43	62	37	43	26	53	60	73			
7	69	70	71	35	48	40	55	38	27	69	35	39	26	59	72	67	30	65	29	55	63	36	38	68	73	42	68			
8	68	61	41	69	51	65	48	68	25	57	25	43	25	32	42	59	26	49	40	43	41	26	51	66	63	30	44			
9	41	53	60	37	40	41	38	60	45	70	36	36	42	48	29	32	35	25	67	58	32	36	31	44	27	70	30			

A partir de la siguiente tabla se presentan los tiempos de alistamiento y procesamiento de trabajos utilizados para comparación entre los resultados arrojados por el ICA para el HFS con buffers de tamaño limitado y el ICA para HFS con buffers de tamaño ilimitado.

**Tabla 47.** Tiempos de procesamiento y alistamiento de tareas para la instancia i7j2k3 de Moratto y Pérez.

Tiempos de Alistamiento																																
k=	1														2																	
j=	1							2							1							2										
h=	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1	27	■	27	35	59	47	53	37	30	■	71	49	38	30	29	45	62	■	70	26	34	25	54	27	40	■	26	43	59	50	67	69
2	65	73	■	34	60	55	54	30	39	44	■	28	73	42	52	53	52	44	■	27	56	65	57	54	47	55	■	37	71	46	56	42
3	70	31	30	■	45	42	31	65	48	30	73	■	30	36	35	41	39	34	49	■	36	55	59	58	64	49	26	■	31	35	70	53
4	52	46	71	31	■	45	52	68	55	61	69	50	■	54	33	26	70	60	52	58	■	45	55	72	62	38	28	36	■	50	29	71
5	33	37	55	40	40	■	31	48	55	32	74	63	55	■	62	56	30	71	53	61	47	■	32	61	37	54	26	69	27	■	27	31
6	68	47	42	72	36	55	■	26	43	55	37	70	70	30	■	40	68	51	48	59	43	29	■	56	55	59	60	61	72	41	■	29
7	44	69	51	55	69	40	54	■	61	44	30	33	64	50	45	■	50	41	69	39	62	39	31	■	26	48	69	48	50	42	55	■

k=	3															
j=	1							2								
h=	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1	60	■	53	60	66	65	31	41	31	■	51	35	26	45	53	53
2	53	49	■	37	74	61	36	70	53	35	■	53	64	37	48	37
3	48	72	69	■	36	35	68	45	65	40	47	■	71	55	35	35
4	27	28	27	52	■	38	63	68	32	42	45	34	■	51	35	43
5	43	55	40	61	55	■	34	28	49	50	43	31	31	■	62	63
6	58	39	64	40	74	46	■	55	36	28	38	33	43	27	■	50
7	74	32	26	28	65	67	41	■	47	28	31	73	40	63	48	■

Tiempos de Procesamiento						
k=	1		2		3	
j=	1	2	1	2	1	2
h=						
1	112	79	76	89	108	88
2	90	108	80	95	89	78
3	90	118	98	80	80	97
4	118	103	75	97	82	114
5	112	79	79	103	82	112
6	87	102	117	83	112	111
7	107	118	98	76	119	102

**Tabla 48.** Tiempos de procesamiento y alistamiento de tareas para la instancia i7j3k3 de Moratto y Pérez.

Tiempos de Alistamiento																																																								
k=	1							2							3							1							2							3																				
j=	1							2							3							1							2							3																				
h=	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1	70	42	60	67	65	56	53	53	53	64	65	61	73	56	27	71	44	61	35	56	51	60	43	52	47	74	37	35	66	66	26	57	67	31	65	39	59	67	43	44	34	73														
2	58	71	65	71	47	48	64	66	40	74	52	60	32	60	27	66	26	71	25	30	36	26	40	28	54	32	35	66	64	56	57	70	55	26	26	57	44	54	59	33	39	28														
3	49	64	46	64	73	74	53	44	50	36	30	31	31	44	60	50	37	73	61	37	45	52	64	73	32	58	63	56	53	43	53	45	59	41	44	60	43	57	71	58	60	57														
4	27	28	56	46	37	72	57	42	47	27	31	69	73	58	44	53	37	30	29	67	72	38	60	69	27	52	30	65	42	65	58	47	45	54	58	36	30	36	53	36	35	69														
5	46	67	31	48	40	61	29	46	67	31	48	40	61	29	41	46	52	55	58	73	43	39	42	48	53	73	32	66	65	44	61	60	73	25	27	56	58	26	47	65	50	59														
6	27	29	55	44	32	41	61	51	41	68	51	69	49	39	73	61	30	31	56	63	54	70	41	28	32	49	44	64	46	38	38	29	26	39	73	40	63	26	56	70	35	50														
7	43	58	45	72	51	73	41	64	61	59	55	71	65	43	28	33	48	57	38	43	36	49	46	39	46	27	69	55	68	62	28	66	42	69	51	70	35	69	29	48	48	64														

k=	3																							
j=	1							2							3									
h=	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1	30	50	54	29	59	43	72	44	44	35	56	56	53	53	28	53	29	42	59	55	73			
2	29	64	74	48	72	62	53	31	48	36	30	36	60	41	56	37	44	36	71	47	70			
3	50	66	60	29	43	70	41	38	71	62	64	64	55	45	44	58	43	36	48	66	34			
4	68	53	48	57	26	42	45	58	60	65	44	27	34	66	58	58	30	44	57	27	51			
5	47	73	28	42	51	68	65	66	38	74	32	70	48	70	46	43	47	47	70	47	47			
6	30	44	37	44	37	65	29	73	62	43	56	28	63	52	73	64	54	29	39	74	67			
7	37	38	58	55	30	32	47	59	60	27	29	47	42	55	36	53	31	34	65	26	52			

Tiempos de Procesamiento									
k=	1			2			3		
j=	1	2	3	1	2	3	1	2	3
h=									
1	76	86	119	92	83	87	115	110	87
2	115	98	82	94	108	91	80	94	118
3	93	122	111	86	90	113	91	86	115
4	94	92	103	81	78	100	103	80	118
5	110	103	105	96	75	84	120	79	107
6	80	119	104	82	114	81	118	83	90
7	84	104	105	106	117	124	86	123	101

**Tabla 49.** Tiempos de procesamiento y alistamiento de tareas para la instancia i7j2k5 de Moratto y Pérez.

Tiempos de Alistamiento																												
k=	1							2																				
j=	1							2																				
h=	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7				
1	61	26	46	70	55	34	28	64	34	49	55	42	26	65	47	30	51	41	25	55	31	41	32	65	71	36	55	70
2	38	42	42	38	67	57	54	53	58	43	39	66	59	44	56	57	70	46	53	31	25	54	35	46	35	46	57	25
3	62	62	54	47	41	49	48	29	32	55	69	68	64	61	64	33	59	42	61	36	60	50	46	50	40	26	68	34
4	29	63	27	68	56	37	57	39	39	36	40	41	25	73	25	67	30	64	52	57	63	72	29	38	60	37	59	65
5	27	55	31	27	65	33	54	40	38	32	27	50	41	34	67	56	55	30	71	36	33	58	49	58	47	40	65	60
6	61	27	38	44	34	32	48	43	56	43	47	41	47	71	51	35	37	28	52	45	65	44	43	34	43	29	73	36
7	27	28	63	66	65	67	37	61	73	41	42	62	50	37	52	67	46	61	53	31	59	34	31	56	69	34	33	50

k=	3							4																				
j=	1							2																				
h=	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7				
1	54	70	27	36	45	61	37	73	56	46	50	50	52	34	31	44	35	35	36	65	30	56	42	74	36	33	47	39
2	29	54	48	42	34	28	50	73	41	44	64	39	40	71	43	40	58	30	48	55	64	55	36	36	44	58	60	33
3	43	57	59	32	45	37	32	57	28	57	37	30	72	67	40	63	59	56	72	56	55	69	67	70	59	59	70	67
4	61	39	45	34	63	36	73	68	64	70	34	61	58	45	32	61	65	26	61	61	56	70	52	55	68	67	69	33
5	59	49	66	64	41	49	40	72	66	66	58	36	44	58	34	57	48	32	47	50	60	72	36	41	57	44	35	42
6	61	42	72	63	39	57	59	72	71	53	74	73	34	43	64	56	52	59	70	32	73	68	44	71	70	68	43	72
7	63	35	38	41	47	44	48	47	43	49	49	53	25	62	69	37	70	61	42	47	40	72	45	27	57	42	26	29

k=	5															
j=	1							2								
h=	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1	59	27	40	37	36	59	51	49	53	46	48	38	30	32		
2	34	26	39	63	33	48	63	54	69	58	37	62	59	32		
3	52	68	66	43	69	73	57	35	27	30	58	32	46	30		
4	55	60	65	74	66	73	70	40	56	63	51	63	53	46		
5	49	55	51	38	45	69	69	69	31	36	30	67	73	36		
6	48	27	42	52	65	71	31	29	55	59	61	54	51	64		
7	69	63	72	70	45	59	33	72	28	65	57	64	33	50		

Tiempos de Procesamiento											
k=	1		2		3		4		5		
j=	1	2	1	2	1	2	1	2	1	2	
h=											
1	106	105	84	100	77	103	102	106	115	124	
2	105	121	77	108	106	116	99	79	96	103	
3	102	113	119	112	121	90	108	116	86	104	
4	77	83	107	122	100	92	97	100	87	125	
5	95	112	120	123	103	78	94	121	108	75	
6	86	105	81	94	78	96	119	81	105	98	
7	110	113	108	93	100	121	96	80	81	76	

**Tabla 50.** Tiempos de procesamiento y alistamiento de tareas para la instancia i7j3k5 de Moratto y Pérez.

Tiempos de Alistamiento																																																
k=	1														2																																	
j=	1							2							3							1							2						3													
h=	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1	52	68	27	71	37	43	57	26	31	47	31	45	42	61	62	44	42	71	54	35	52	39	54	32	42	54	70	33	40	36	40	32	33	38	53	41	67	38	71	40	42	42						
2	43	31	59	74	32	40	46	49	58	38	33	59	30	63	37	57	43	64	37	54	74	73	52	41	68	37	48	34	69	69	69	48	25	27	45	44	36	53	55	64	58	42						
3	34	49	29	43	25	57	61	38	47	60	58	53	34	58	52	68	31	50	72	70	57	26	35	38	48	46	42	26	52	41	39	63	54	71	60	25	68	62	71	69	53	39						
4	43	54	36	52	67	26	30	29	70	32	28	67	47	48	64	62	48	55	28	55	51	58	38	29	53	70	46	69	66	64	39	73	52	49	48	54	35	62	74	26	71	41						
5	48	40	56	40	38	63	72	32	49	41	69	60	34	71	64	57	61	54	32	39	36	64	56	65	67	61	58	54	54	46	60	45	40	69	30	32	27	44	51	73	72	39						
6	53	60	60	42	28	27	36	72	68	48	50	67	45	70	62	33	42	40	70	27	31	64	71	25	47	55	51	68	38	43	68	30	73	55	63	51	43	64	54	27	43	49						
7	42	47	66	53	50	34	66	38	71	32	67	31	69	56	51	70	63	32	34	34	60	72	48	66	54	71	48	33	51	55	40	63	73	61	60	28	67	32	61	51	69	68						

k=	3														4																																	
j=	1							2							3							1							2						3													
h=	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1	53	56	40	36	50	49	61	56	38	37	62	28	49	53	63	26	42	39	62	64	54	58	41	73	57	61	35	29	67	36	33	55	41	52	30	44	68	60	65	53	29	52						
2	55	46	26	26	49	68	71	54	64	50	54	34	71	69	46	31	35	44	70	62	43	68	47	74	41	71	42	63	51	70	31	47	53	59	30	30	27	70	30	66	48	61						
3	35	53	62	59	50	46	62	45	73	53	40	54	41	39	33	40	42	37	31	46	43	54	67	61	56	43	29	51	26	26	49	65	48	50	40	73	69	64	40	65	67	26						
4	31	70	44	57	37	27	46	67	64	26	46	64	37	57	51	40	56	28	67	64	56	68	62	54	73	38	61	48	49	29	58	66	58	45	31	59	40	69	33	57	73	71						
5	47	64	56	48	26	35	37	50	41	66	66	62	37	36	41	63	33	32	29	33	34	38	61	66	29	32	42	39	61	51	41	28	71	65	55	73	55	36	70	36	33	41						
6	29	50	32	59	43	31	28	34	64	67	59	39	27	34	51	58	72	32	49	35	32	47	63	57	70	73	72	27	63	28	46	50	43	36	41	49	61	43	33	26	41	42	59					
7	68	69	42	27	41	65	52	35	65	69	73	31	28	59	59	40	44	53	66	63	68	39	56	41	36	60	28	39	44	43	64	46	60	48	51	65	37	33	57	40	39	41						

k=	5																							
j=	1							2							3									
h=	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1	30	61	56	30	36	42	70	34	72	37	35	74	42	37	54	61	48	39	52	34	31			
2	35	35	34	59	27	59	72	39	33	63	32	25	63	36	51	67	29	40	48	33	65			
3	40	36	33	68	63	30	67	49	47	68	36	42	53	34	50	46	72	37	63	60	45			
4	34	34	26	63	55	68	37	67	64	36	25	57	26	25	61	49	41	38	43	54	49			
5	39	44	69	63	33	45	74	56	64	28	48	28	28	33	47	58	31	72	71	31	31			
6	39	39	71	48	58	39	33	53	62	42	30	55	50	73	27	27	39	36	63	39	48			
7	70	56	72	63	44	34	60	42	26	57	66	63	29	30	64	70	34	50	29	27	32			

Tiempos de Procesamiento															
k=	1			2			3			4			5		
j=	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
h=	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
1	113	107	121	117	104	124	77	121	104	88	111	117	111	107	92
2	85	75	77	124	84	103	86	122	93	116	78	98	99	80	115
3	105	85	97	92	86	121	125	118	95	115	93	112	83	81	98
4	119	84	114	94	91	119	113	91	86	99	120	83	96	99	86
5	124	103	82	77	104	115	84	82	81	119	120	95	120	104	108
6	102	80	122	121	112	110	77	119	112	96	88	117	108	124	80
7	79	90	93	83	113	114	105	113	80	116	81	84	102	106	113

**Tabla 51.** Tiempos de procesamiento y alistamiento de tareas para la instancia i9j2k3 de Moratto y Pérez.

Tiempos de Alistamiento																					
k=	1									2											
j=	1									2											
h=	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	
1	60	43	59	70	56	27	58	42	53	36	51	57	58	47	32	27	53	46	43	29	52
2	35	60	69	29	63	30	45	39	32	51	59	74	58	59	48	73	74	49	63	32	71
3	71	57	50	69	62	41	29	52	65	47	54	64	48	59	46	27	51	57	44	49	41
4	50	66	55	43	65	66	55	71	44	36	40	37	69	44	52	67	35	61	51	68	63
5	28	45	39	33	43	64	64	42	38	66	35	42	31	57	36	66	42	64	43	30	33
6	59	74	63	55	43	29	72	32	51	59	40	69	36	70	68	50	47	57	70	60	42
7	51	49	36	43	61	43	64	69	32	36	49	33	42	70	57	31	33	28	51	40	50
8	29	39	32	33	26	37	55	55	58	61	36	48	45	28	73	57	64	35	52	49	59
9	49	37	61	70	39	34	57	67	56	64	46	27	57	48	55	67	33	71	74	47	33

k=	3									
j=	1									
h=	0	1	2	3	4	5	6	7	8	9
1	71	54	26	61	38	25	39	60	57	70
2	62	36	38	42	27	57	55	58	59	71
3	39	35	43	29	62	31	71	49	61	48
4	33	52	56	66	25	48	58	68	49	57
5	70	48	30	33	44	71	34	50	53	34
6	65	66	61	54	69	73	43	62	71	69
7	32	70	30	34	72	49	65	51	46	68
8	65	62	39	56	71	30	60	47	53	52
9	61	66	39	71	34	38	55	56	68	39

Tiempos de Procesamiento						
k=	1		2		3	
j=	1	2	1	2	1	2
h=						
1	78	84	80	92	84	99
2	77	100	107	110	124	122
3	121	84	88	80	83	81
4	95	76	81	105	88	124
5	94	86	108	93	76	123
6	100	83	82	100	122	85
7	116	85	121	110	92	77
8	75	82	118	75	111	116
9	76	88	96	102	94	122

**Tabla 52.** Tiempos de procesamiento y alistamiento de tareas para la instancia i9j3k3 de Moratto y Pérez.

Tiempos de Alistamiento																																																																																																			
k=	1																		2																		3																																																														
j=	1						2						3						1						2						3																																																																				
h=	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9																																																	
1	40	42	58	30	38	38	65	42	57	39	52	34	40	60	68	28	55	73	48	43	53	33	70	33	61	71	39	56	35	53	68	68	57	66	57	33	72	40	72	69	57	32	29	56	72	48	65	35	73	30	44	58	71	40																																													
2	30	43	40	65	57	43	39	65	56	62	61	43	67	74	34	40	41	50	63	44	36	32	57	42	64	39	67	70	74	35	69	35	50	36	48	68	35	52	55	27	49	53	60	46	34	32	29	39	28	58	33	65	48	64																																													
3	73	46	50	34	72	49	30	48	59	45	71	67	29	45	40	35	36	54	26	72	44	28	72	64	73	72	58	30	67	25	47	30	49	71	50	73	59	26	33	65	54	66	50	60	57	53	40	38	61	64	31	36	29	32																																													
4	51	44	32	74	54	33	26	57	29	47	37	38	48	49	27	39	74	74	53	61	71	68	36	65	34	39	62	39	29	62	30	61	56	40	66	66	30	51	35	42	65	66	42	45	44	28	28	66	56	70	62	64	46	71																																													
5	55	57	40	51	57	45	53	27	28	49	48	56	69	53	37	64	65	42	73	29	62	74	67	30	39	37	57	43	30	70	54	34	56	35	31	33	30	50	33	29	73	52	69	58	59	43	26	58	67	61	36	52	31	40																																													
6	53	30	37	50	31	38	62	40	39	62	34	55	62	53	30	51	40	63	71	72	61	60	34	59	66	49	28	51	73	28	27	49	33	64	46	44	51	34	71	57	34	44	30	27	35	58	54	73	70	52	38	39	37	70																																													
7	49	43	41	59	32	40	31	26	53	47	73	43	74	26	52	71	65	42	49	46	54	61	73	67	39	45	70	35	35	51	36	33	55	66	54	44	70	47	54	73	71	43	30	50	67	63	69	65	52	44	40	73	72	54																																													
8	35	43	45	53	64	39	54	48	25	70	47	61	28	69	45	35	33	39	41	53	47	66	41	55	36	62	51	38	46	58	46	33	55	42	53	49	51	53	56	48	42	60	38	64	53	43	69	53	30	49	65	66	41	57																																													
9	72	60	61	32	39	64	70	26	46	27	34	63	28	73	73	73	66	48	31	47	52	44	39	48	69	64	42	56	40	71	62	56	38	65	48	38	32	54	41	72	32	34	53	49	64	41	26	42	68	30	50	50	31	71																																													

k=	3																													
j=	1						2						3																	
h=	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
1	26	28	39	55	74	32	69	61	45	34	33	41	31	74	69	35	54	28	31	39	49	71	34	69	66	34	33			
2	43	35	70	60	58	73	43	33	74	59	74	65	62	71	73	45	63	71	30	51	34	61	56	30	52	30	50			
3	67	36	61	55	51	31	51	37	68	35	65	71	27	47	57	49	32	49	68	73	40	63	54	57	50	35	33			
4	63	67	40	46	73	37	72	67	70	66	71	64	28	28	69	32	51	56	58	55	29	66	45	40	50	73	32			
5	56	28	74	49	37	43	53	55	47	33	38	68	54	31	69	71	29	31	63	69	32	41	53	41	55	47	52			
6	35	65	33	41	69	41	31	38	42	63	53	36	29	28	44	45	48	45	45	62	32	27	51	57	35	55	56			
7	32	56	47	57	27	73	70	51	27	38	28	58	55	43	64	57	37	35	65	43	62	34	40	56	37	72	60			
8	38	46	58	46	33	55	42	53	49	63	56	63	49	68	56	71	71	57	69	39	60	57	57	50	66	56	66			
9	30	63	26	31	62	30	26	49	38	42	46	26	60	50	44	63	54	44	31	37	38	64	39	37	37	39	35			

Tiempos de Procesamiento									
k=	1			2			3		
j=	1	2	3	1	2	3	1	2	3
1	115	99	81	90	97	111	104	90	115
2	121	93	89	92	116	101	110	103	87
3	107	89	84	112	101	117	85	119	109
4	90	93	111	100	80	99	117	109	88
5	91	83	80	90	75	111	109	98	85
6	111	90	112	77	105	81	82	105	87
7	89	97	105	93	101	123	94	119	115
8	110	80	100	79	96	105	95	76	115
9	114	122	118	114	102	108	92	105	93

**Tabla 53.** Tiempos de procesamiento y alistamiento de tareas para la instancia i9j2k5 de Moratto y Pérez.

Tiempos de Alistamiento																																				
k=	1										2																									
j=	1					2					1				2																					
h=	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9						
1	60	41	57	47	43	66	36	33	60	33	42	49	31	46	36	44	26	52	51	71	70	47	35	51	74	26	56	29	74	72	45	44	42	66	66	56
2	32	40	28	39	48	38	48	48	48	60	35	25	32	32	53	39	46	51	58	36	58	63	58	33	73	57	53	31	63	35	25	42	42	40	59	27
3	28	32	43	37	59	50	35	34	28	27	65	65	35	41	32	58	50	53	57	61	35	42	51	35	62	49	67	28	56	71	64	62	66	59	43	65
4	40	41	29	44	29	29	57	26	44	31	66	31	28	54	48	43	41	40	31	30	32	28	38	49	45	45	51	64	69	51	47	29	30	41	72	61
5	70	29	58	55	27	61	57	45	31	46	67	31	67	29	69	59	26	41	47	52	33	28	42	34	41	30	41	58	57	40	60	40	29	51	49	53
6	45	68	28	32	70	70	34	70	67	30	73	60	54	70	32	53	69	50	44	52	59	74	53	51	30	33	27	28	71	46	41	44	32	48	38	73
7	45	41	54	71	53	63	73	55	65	62	70	57	50	54	35	59	45	48	30	64	73	46	41	56	45	39	29	28	66	74	68	65	30	38	38	46
8	69	59	74	26	70	60	37	50	33	28	34	53	32	39	66	40	58	29	27	30	54	35	50	54	48	58	33	36	26	45	73	45	71	70	28	
9	72	52	56	43	37	63	33	70	44	35	63	44	38	62	55	59	48	50	30	73	42	47	40	44	28	67	70	28	53	66	59	65	28	45	54	30

Tiempos de Alistamiento																																								
k=	3										4																													
j=	1					2					1					2																								
h=	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
1	38	39	35	54	49	53	60	43	62	67	67	39	65	27	55	45	51	72	34	74	67	26	70	74	30	68	37	48	59	55	48	36	49	28	51	70				
2	34	35	42	27	67	59	32	54	42	26	66	31	34	55	64	51	41	39	65	39	65	44	33	41	49	58	34	34	69	67	45	54	44	63	65	60				
3	49	71	61	67	68	54	28	32	30	55	45	43	69	47	57	27	61	56	32	55	37	51	53	66	54	64	48	54	57	35	62	34	25	68	61	30				
4	44	70	45	41	56	42	42	45	44	51	53	74	40	55	30	48	46	61	48	33	39	47	52	30	35	38	25	60	31	26	45	67	73	74	30	28				
5	45	51	45	57	52	27	57	45	37	36	60	66	35	34	62	30	49	71	64	53	50	43	49	49	50	60	31	53	58	72	68	47	57	44	29	60				
6	40	70	49	58	29	68	29	47	38	50	47	34	25	61	55	55	63	48	62	26	27	50	71	27	40	51	52	72	38	67	35	47	69	27	66	68				
7	31	55	35	41	56	74	38	69	47	51	25	66	40	55	67	46	34	42	63	30	49	31	53	61	45	45	58	32	57	49	49	59	31	57	42	43				
8	64	31	44	74	37	31	42	56	62	64	28	51	28	52	63	69	42	71	65	32	31	38	40	50	34	26	42	67	32	51	36	65	29	27	73	49				
9	56	48	66	61	53	41	51	63	28	29	50	54	29	68	32	32	26	25	66	26	49	31	35	37	28	48	60	32	48	56	42	29	47	38	71	66				

Tiempos de Alistamiento																																					
k=	5																																				
j=	1					2																															
h=	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9																	
1	28	61	44	29	57	57	29	72	65	69	52	41	48	74	53	45	45	55																			
2	72	56	58	52	44	40	26	51	38	41	74	37	37	48	26	44	27	39																			
3	37	65	26	51	56	46	37	49	61	74	60	45	49	34	41	47	48	74																			
4	30	72	42	72	56	74	60	64	26	51	42	39	66	57	38	61	56	52																			
5	42	32	39	67	39	69	72	71	64	59	31	39	40	68	28	67	47	36																			
6	31	72	65	37	51	32	50	70	44	36	64	54	41	67	45	50	59	31																			
7	50	57	67	45	57	40	35	65	55	68	26	32	56	55	49	48	66	46																			
8	28	48	55	73	70	57	29	52	43	40	61	42	49	39	52	37	38	64																			
9	32	37	28	26	56	44	37	67	61	35	36	51	51	46	44	37	70	53																			

Tiempos de Procesamiento										
k=	1		2		3		4		5	
j=	1	2	1	2	1	2	1	2	1	2
h=	1	2	1	2	1	2	1	2	1	2
1	123	80	111	104	77	102	102	105	90	104
2	120	76	100	116	104	90	118	82	103	92
3	82	110	99	102	101	105	88	77	120	104
4	87	104	80	88	88	88	101	116	125	104
5	92	106	85	75	123	107	104	90	96	115
6	101	106	106	81	117	109	114	119	111	78
7	114	101	115	82	112	81	121	104	75	113
8	79	77	99	98	94	88	102	80	112	82
9	88	94	108	122	90	108	94	112	113	124



## Anexo C. Pseudocódigo de programación en lenguaje *Matlab* de la metaheurística basada en el algoritmo competitivo imperialista.

Se detallan a continuación los códigos de la función principal y funciones suplementarias de la metaheurística basada en el algoritmo competitivo imperialista las cuales se usaron para dar solución al problema del *flow shop* híbrido con máquinas paralelas no relacionadas, tiempos de alistamiento dependientes de la secuencia y *buffers* de tamaño limitado.

Varios de los códigos y funciones utilizados para la metaheurística están basados en el código del algoritmo competitivo imperialista desarrollado también en *Matlab* y los cuales pertenecen al grupo de investigación OPALO<sup>124</sup> de la Universidad Industrial de Santander.

- Función de interfaz gráfica para la entrada de datos y parámetros.

```
function varargout = main_imperio(varargin)
% MAIN_IMPERIO MATLAB code for main_imperio.fig
%   MAIN_IMPERIO, by itself, creates a new MAIN_IMPERIO or raises the
existing
%   singleton*.
%
%   H = MAIN_IMPERIO returns the handle to a new MAIN_IMPERIO or the
handle to
%   the existing singleton*.
%
%   MAIN_IMPERIO('CALLBACK', hObject,eventData,handles,...) calls the
local
%   function named CALLBACK in MAIN_IMPERIO.M with the given input
arguments.
%
%   MAIN_IMPERIO('Property','Value',...) creates a new MAIN_IMPERIO or
raises the
%   existing singleton*. Starting from the left, property value pairs
are
%   applied to the GUI before main_imperio_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property
application
%   stop. All inputs are passed to main_imperio_OpeningFcn via
varargin.
%
```

---

<sup>124</sup> Grupo de Investigación en sistemas Productivos, Administrativos y Logísticos (OPALO). Escuela de Estudios Industriales y Empresariales. Universidad Industrial de Santander. Director: Carlos Eduardo Díaz Bohórquez (cediazbo@uis.edu.co).

```

%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only
one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help main_imperio

% Last Modified by GUIDE v2.5 19-Dec-2016 08:56:01

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @main_imperio_OpeningFcn, ...
                  'gui_OutputFcn',  @main_imperio_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before main_imperio is made visible.
function main_imperio_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to main_imperio (see VARARGIN)
addpath('support')
addpath('images')
addpath('resultados')
set(handles.Pais_mejor,'Data',cell(2,31));
set(handles.radiobutton2,'Value',1);
clear global costos
axes(handles.axes1)
matlabImage = imread('images\logo_uis.png');
image(matlabImage)
axis off
axis image

axes(handles.axes2)
matlabImage = imread('images\logo_opalo.jpg');
image(matlabImage)
axis off

```

```

axis image

axes(handles.axes3)
matlabImage = imread('images\logo_escuela.jpg');
image(matlabImage)
axis off
axis image
% Choose default command line output for main_imperio
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes main_imperio wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = main_imperio_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function cantEtap_Callback(hObject, eventdata, handles)
% hObject handle to cantEtap (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of cantEtap as text
% str2double(get(hObject,'String')) returns contents of cantEtap
as a double

% --- Executes during object creation, after setting all properties.
function cantEtap_CreateFcn(hObject, eventdata, handles)
% hObject handle to cantEtap (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

```

```

function cantTrab_Callback(hObject, eventdata, handles)
% hObject    handle to cantTrab (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of cantTrab as text
%        str2double(get(hObject,'String')) returns contents of cantTrab
as a double

% --- Executes during object creation, after setting all properties.
function cantTrab_CreateFcn(hObject, eventdata, handles)
% hObject    handle to cantTrab (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function cantPais_Callback(hObject, eventdata, handles)
% hObject    handle to cantPais (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of cantPais as text
%        str2double(get(hObject,'String')) returns contents of cantPais
as a double

% --- Executes during object creation, after setting all properties.
function cantPais_CreateFcn(hObject, eventdata, handles)
% hObject    handle to cantPais (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function cantMaqu_Callback(hObject, eventdata, handles)
% hObject    handle to cantMaqu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of cantMaqu as text
%        str2double(get(hObject,'String')) returns contents of cantMaqu
as a double

% --- Executes during object creation, after setting all properties.
function cantMaqu_CreateFcn(hObject, eventdata, handles)
% hObject    handle to cantMaqu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function porcen_imperialistas_Callback(hObject, eventdata, handles)
% hObject    handle to porcen_imperialistas (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of porcen_imperialistas
as text
%        str2double(get(hObject,'String')) returns contents of
porcen_imperialistas as a double

% --- Executes during object creation, after setting all properties.
function porcen_imperialistas_CreateFcn(hObject, eventdata, handles)
% hObject    handle to porcen_imperialistas (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

end

```
function par_chi_Callback(hObject, eventdata, handles)
% hObject    handle to par_chi (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of par_chi as text
%        str2double(get(hObject,'String')) returns contents of par_chi as
a double
```

```
% --- Executes during object creation, after setting all properties.
function par_chi_CreateFcn(hObject, eventdata, handles)
% hObject    handle to par_chi (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function par_compartir_Callback(hObject, eventdata, handles)
% hObject    handle to par_compartir (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of par_compartir as text
%        str2double(get(hObject,'String')) returns contents of
par_compartir as a double
```

```
% --- Executes during object creation, after setting all properties.
function par_compartir_CreateFcn(hObject, eventdata, handles)
% hObject    handle to par_compartir (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

end

```
function par_revolucion_Callback(hObject, eventdata, handles)
% hObject    handle to par_revolucion (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of par_revolucion as text
%        str2double(get(hObject,'String')) returns contents of
par_revolucion as a double
```

```
% --- Executes during object creation, after setting all properties.
function par_revolucion_CreateFcn(hObject, eventdata, handles)
% hObject    handle to par_revolucion (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function dir_costos_Callback(hObject, eventdata, handles)
% hObject    handle to dir_costos (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of dir_costos as text
%        str2double(get(hObject,'String')) returns contents of dir_costos
as a double
```

```
% --- Executes during object creation, after setting all properties.
function dir_costos_CreateFcn(hObject, eventdata, handles)
% hObject    handle to dir_costos (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
end
```

```
% --- Executes on button press in cargar_costos.
function cargar_costos_Callback(hObject, eventdata, handles)
% hObject    handle to cargar_costos (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.figure1, 'pointer', 'watch')
drawnow;
[FileName Path] = uigetfile({'*.xlsx'}, 'Escoger');
if isequal(FileName,0)
    return;
else
    set(handles.dir_costos, 'String', strcat(Path, FileName));
    %set(handles.button_Confirmar_fcd, 'Enable', 'on');
end
global costos
Dir=get(handles.dir_costos, 'String');
costos = xlsread(Dir, 'Hojal');
set(handles.cargar_costos, 'String', 'Cargado')
set(handles.figure1, 'pointer', 'arrow')
```

```
function edit13_Callback(hObject, eventdata, handles)
% hObject    handle to edit13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of edit13 as text
%        str2double(get(hObject, 'String')) returns contents of edit13 as
a double
```

```
% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
```

```
function edit15_Callback(hObject, eventdata, handles)
```

```

% hObject    handle to edit15 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit15 as text
%         str2double(get(hObject,'String')) returns contents of edit15 as
a double

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit15 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit_Iter_Callback(hObject, eventdata, handles)
% hObject    handle to edit_Iter (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_Iter as text
%         str2double(get(hObject,'String')) returns contents of edit_Iter
as a double

% --- Executes during object creation, after setting all properties.
function edit_Iter_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_Iter (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in radiobutton1.
function radiobutton1_Callback(hObject, eventdata, handles)

```

```

% hObject    handle to radiobutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of radiobutton1
global criterio
criterio = 'Iteraciones';
set(handles.radiobutton2,'Value', 0);
set(handles.edit_Iter,'Enable','on');
% --- Executes on button press in radiobutton2.
function radiobutton2_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of radiobutton2
global criterio
criterio = 'Default';
set(handles.radiobutton1,'Value', 0);
set(handles.edit_Iter,'Enable','off');

% --- Executes on button press in Ejecutar.
function Ejecutar_Callback(hObject, eventdata, handles)
% hObject    handle to Ejecutar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

set(handles.figure1, 'pointer', 'watch')
drawnow;

global cantMaqu
global cantPais
global cantTrab
global cantEtap
global porcen_imperialistas
global par_chi
global par_compartir % porcentaje de colonias que comparten informacion
global par_revolucion
global cos_bueno
global solucion
global costos
global tiempo_al
global Iter
global criterio
global iteraciones_luego
global capBuffer
global solucion_FIFO

cantMaqu=str2double(get(handles.cantMaqu,'String'));
cantPais=str2double(get(handles.cantPais,'String'));
cantTrab=str2double(get(handles.cantTrab,'String'));

```

```

cantEtap = str2double(get(handles.cantEtap, 'String'));
porcen_imperialistas=str2double(get(handles.porcen_imperialistas, 'String'
));
par_chi=str2double(get(handles.par_chi, 'String'));
par_compartir=str2double(get(handles.par_compartir, 'String'));
par_revolucion=str2double(get(handles.par_revolucion, 'String'));
iteraciones_luego=str2double(get(handles.edit_Iter, 'String'));
criterio=str2double(get(handles.edit_Iter, 'String'));
capBuffer=str2double(get(handles.capBuffer, 'String'));

if(size(costos,1)~= (cantTrab^2+cantTrab)*cantMaqu ||
size(costos,2)~=cantEtap)
h = msgbox('ERROR: Las dimensiones de la matriz de costos no son
consistentes con los parámetros', '¡ERROR!');
else
tic
[cos_bueno , solucion, Iter,
solucion_FIFO]=imperio(iteraciones_luego,criterio, cantMaqu, cantPais,
cantTrab, cantEtap, porcen_imperialistas, par_chi , par_compartir,
par_revolucion,costos,capBuffer);
tiempo_al = toc;
set(handles.Pais_mejor, 'Data', solucion);
set(handles.editTime, 'String', tiempo_al);
set(handles.editMakespan, 'String', cos_bueno);
set(handles.editItera, 'String', Iter);
set(handles.solucion_FIFO, 'String', solucion_FIFO);
end
% your computation
set(handles.figure1, 'pointer', 'arrow')

% --- If Enable == 'on', executes on mouse press in 5 pixel border.
% --- Otherwise, executes on mouse press in 5 pixel border or over
Ejecutar.
function Ejecutar_ButtonDownFcn(hObject, eventdata, handles)
% hObject      handle to Ejecutar (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% --- Executes on key press with focus on Ejecutar and none of its
controls.
function Ejecutar_KeyPressFcn(hObject, eventdata, handles)
% hObject      handle to Ejecutar (see GCBO)
% eventdata    structure with the following fields (see UICONTROL)
%   Key: name of the key that was pressed, in lower case
%   Character: character interpretation of the key(s) that was pressed
%   Modifier: name(s) of the modifier key(s) (i.e., control, shift)
pressed
% handles      structure with handles and user data (see GUIDATA)

function editTime_Callback(hObject, eventdata, handles)

```

```

% hObject    handle to editTime (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of editTime as text
%         str2double(get(hObject,'String')) returns contents of editTime
as a double

% --- Executes during object creation, after setting all properties.
function editTime_CreateFcn(hObject, eventdata, handles)
% hObject    handle to editTime (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function editMakespan_Callback(hObject, eventdata, handles)
% hObject    handle to editMakespan (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of editMakespan as text
%         str2double(get(hObject,'String')) returns contents of
editMakespan as a double

% --- Executes during object creation, after setting all properties.
function editMakespan_CreateFcn(hObject, eventdata, handles)
% hObject    handle to editMakespan (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function editItera_Callback(hObject, eventdata, handles)

```

```

% hObject    handle to editItera (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of editItera as text
%         str2double(get(hObject,'String')) returns contents of editItera
as a double

% --- Executes during object creation, after setting all properties.
function editItera_CreateFcn(hObject, eventdata, handles)
% hObject    handle to editItera (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
set(handles.cantMaqu,'String','2');
set(handles.cantPais,'String','250');
set(handles.cantTrab,'String','5');
set(handles.cantEtap,'String','3');
set(handles.capBuffer,'String','1');
set(handles.porcen_imperialistas,'String','0.3');
set(handles.par_chi,'String','0.31395');
set(handles.par_compartir,'String','0.5');
set(handles.par_revolucion,'String','0.3');
set(handles.dir_costos,'String','');
set(handles.cargar_costos,'String','Cargar');
set(handles.edit_Iter,'String','50');
set(handles.edit_Iter,'enable','off');
set(handles.radiobutton2,'Value',1);
set(handles.radiobutton1,'Value',0);
set(handles.editTime,'String','');
set(handles.editMakespan,'String','');
set(handles.editItera,'String','');
set(handles.Pais_mejor,'Data',cell(2,31));
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
set(handles.figure1, 'pointer', 'watch')
drawnow;
global solucion
global nombre
nombre = get(handles.nombre, 'String');
%xlswrite(['\resultados\', nombre, '.xlsx'],solucion)
xlswrite([nombre, '.xlsx'],solucion)
set(handles.figure1, 'pointer', 'arrow')

function nombre_Callback(hObject, eventdata, handles)
% hObject handle to nombre (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of nombre as text
% str2double(get(hObject,'String')) returns contents of nombre as
a double

% --- Executes during object creation, after setting all properties.
function nombre_CreateFcn(hObject, eventdata, handles)
% hObject handle to nombre (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function capBuffer_Callback(hObject, eventdata, handles)
% hObject handle to capBuffer (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of capBuffer as text
% str2double(get(hObject,'String')) returns contents of capBuffer
as a double

% --- Executes during object creation, after setting all properties.
function capBuffer_CreateFcn(hObject, eventdata, handles)
% hObject handle to capBuffer (see GCBO)

```

```
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

- **Función principal de la metaheurística**

```
function [cos_bueno , solucion, Iter,
solucion_FIFO]=imperio(iteraciones_luego, criterio, cantMaqu, cantPais,
cantTrab, cantEtap, porcen_imperialistas, par_chi , par_compartir,
par_revolucion, costos,capBuffer)
Iter=0;
% IMPERIO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% **PASO 1 **
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
[costos_sepa]=separa_costos(costos,cantEtap, cantTrab,cantMaqu);
clear costos
%% generacion de paises
[Paises]=genera_paises(cantPais,cantMaqu, cantTrab,
cantEtap,capBuffer,costos_sepa);
%load('test_paises.mat');
%Paises= test_paises;
%% calcula el makespan de cada pais
makespan =
calcula_makespan(Paises,cantPais,cantMaqu,cantTrab,cantEtap,costos_sepa,c
apBuffer);
%% identificacion de los imperialistas
solucion_FIFO = min(makespan);
[val, pos_pa]=sort(makespan);
val = val';
pos_pa = pos_pa';
cantImper=round(porcen_imperialistas*cantPais);

%% calcula costos y poder normalizados
[pod_nor] = costo_normalizado(cantImper,val);
clear val
%% calcula cantidad de colonias
cant_Colo=cantidad_colonias(cantImper,cantPais,pod_nor);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% **PASO 2 **
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

%% asignación de colonias
Imper= asignacion_colonias(cantImper,pos_pa,cant_Colo);
%load('test_imper.mat')
%Imper=test_Imper;

%% asimilacion
posi_imperialistas=pos_pa(1:cantImper);
while (size(Imper,1)>1)
Iter=Iter+1;
[Países, makespan]= asimilacion(cantImper, cantMaqu, cantTrab,
cantEtap,posi_imperialistas,cant_Colo,Imper,Países,costos_sepa,makespan,c
apBuffer);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PASO 3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Compartir informacion entre colonias
[makespan , Países]=compartir_colonias(costos_sepa, makespan, Países,
par_compartir, cantImper, cantEtap, cantMaqu, cantTrab, cant_Colo,
Imper,capBuffer);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PASO 4
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% revolucion
[makespan , Países]=revolucion(costos_sepa, cantEtap, makespan, Países,
cantImper, cant_Colo, par_revolucion, Imper,cantMaqu,
cantTrab,capBuffer);

%% recalcu lo imperialista
posi_imperialistas=zeros(cantImper,1);
for k=1:cantImper
[val , pos]=min(makespan(Imper{k}));
aux_temp=Imper{k}(1,1);
Imper{k}(1,1)=Imper{k}(pos,1);
posi_imperialistas(k,1)=Imper{k}(1,1);
Imper{k}(pos,1)=aux_temp;
end
clear aux_temp
clear val
%% recalcu lo costo imperio
costo_Imper=zeros(cantImper,1);

```

```

for k=1:cantImper

costo_Imper(k,1)=makespan(Imper{k}(1,1))+par_chi*mean(makespan(Imper{k}(2
:end)));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PASO 5 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% guerra entre imperios
[Imper, cant_Colo, cantImper] = guerra_imperios(costo_Imper, Imper,
cant_Colo);
posi_imperialistas=zeros(cantImper,1);
for k=1:cantImper
    posi_imperialistas(k,1)=Imper{k}(1,1);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end
clear costo_Imper
clear k
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SEGUNDA PARTE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[cos_bueno , pais_buen]=min(makespan);
solucion=Paises(pais_buen,:);
switch (criterio)
    case 'Iteraciones'
        cont_iter=0;
        while (cont_iter <= iteraciones_luego)
            cont_iter=cont_iter+1;
            % asimilacion
            [Paises, makespan]= asimilacion(cantImper, cantMaqu,
cantTrab,
cantEtap,posi_imperialistas,cant_Colo,Imper,Paises, costos_sepa,makespan,c
apBuffer);
            % Compartir informacion entre colonias
            [makespan , Paises]=compartir_colonias(costos_sepa, makespan,
Paises, par_compartir, cantImper, cantEtap, cantMaqu, cantTrab,
cant_Colo, Imper,capBuffer);
            % revolucion
            [makespan , Paises]=revolucion(costos_sepa, cantEtap,
makespan, Paises, cantImper, cant_Colo, par_revolucion, Imper,cantMaqu,
cantTrab,capBuffer);
            % recalcu lo imperialista
            posi_imperialistas=zeros(cantImper,1);
            for k=1:cantImper
                [val , pos]=min(makespan(Imper{k}));
                aux_temp=Imper{k}(1,1);
                Imper{k}(1,1)=Imper{k}(pos,1);
                posi_imperialistas(k,1)=Imper{k}(1,1);
                Imper{k}(pos,1)=aux_temp;
            end

            clear val
            clear aux_temp

```

```

        end
        case 'Default'
            cambio=1;
            while(cambio==1)
                % asimilacion
                [Paises, makespan]= asimilacion(cantImper, cantMaqu,
cantTrab,
cantEtap, posi_imperialistas, cant_Colo, Imper, Paises, costos_sepa, makespan, c
apBuffer);
                % Compartir informacion entre colonias
                [makespan , Paises]=compartir_colonias(costos_sepa, makespan,
Paises, par_compartir, cantImper, cantEtap, cantMaqu, cantTrab,
cant_Colo, Imper, capBuffer);
                % revolucion
                [makespan , Paises]=revolucion(costos_sepa, cantEtap,
makespan, Paises, cantImper, cant_Colo, par_revolucion, Imper, cantMaqu,
cantTrab, capBuffer);
                % recalculo imperialista
                posi_imperialistas=zeros(cantImper,1);
                for k=1:cantImper
                    [val , pos]=min(makespan(Imper{k}));
                    if(pos==1)
                        cambio=0;
                    end
                    aux_temp=Imper{k}(1,1);
                    Imper{k}(1,1)=Imper{k}(pos,1);
                    posi_imperialistas(k,1)=Imper{k}(1,1);
                    Imper{k}(pos,1)=aux_temp;
                end
                clear val
                clear aux_temp
            end

end

%cos_bueno=makespan(Imper{1}(1,1));
solucion=Paises(Imper{1}(1,1),:);

solucion(2,:)=makespan_individual(1,Paises(Imper{1}(1,1),:),cantMaqu,cant
Trab,cantEtap,costos_sepa,capBuffer);
cos_bueno=max(solucion(2,:));
end

```

- Función de complemento a la función principal para los ajustes de la metaheurística al problema con *buffers* limitados.

```

clear all
close all
clc

% IMPERIO
tic

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% **PASO 1 **
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%% solicita parametros
%maxMaqu=2; minMaqu=2; % limites para las maquinas
%maxTrab=9; minTrab=3; % limites para los trabajos
%maxPais=20; minPais=10000; %limites para los paises
%maxEtap=7; minEtap=2; %limites para las etapas
%maxCost=175; minCost=90; %limites para el costo de procesos
porcen_imperialistas=0.25; % cantidad de imperialistas iniciales
par_chi= 0.31395; % peso del costo de las colonias
par_compartir=0.3; % porcentaje de colonias que comparten informacion
par_revolucion = 0.5; % porcentaje de colonias que se revolucionan
%% genera aleatoriamente las cantidades a manejar
%[cantMaqu, cantTrab, cantPais,
cantEtap]=genera_cantidades(maxMaqu,minMaqu,maxTrab,minTrab, maxPais,
minPais, maxEtap, minEtap);
%de prueba como el ejemplo
cantMaqu=3;
cantPais=1000;
cantTrab=5;
cantEtap=3;

%% generacion de matriz de costos
%costos = floor((maxCost-
minCost+1)*rand(cantMaqu*(cantTrab^2+cantTrab),cantEtap))+ minCost;
load('test_costos1.mat');
costos=test_costos;
clear test_costos
[costos_sepa]=separa_costos(costos,cantEtap, cantTrab,cantMaqu);
clear costos

%% generacion de paises
[Paises]=genera_paises(cantPais,cantMaqu, cantTrab, cantEtap);
%load('test_paises.mat');
%Paises= test_paises;
%% calcula el makespan de cada pais
makespan =
calcula_makespan(Paises,cantPais,cantMaqu,cantTrab,cantEtap,costos_sepa);
%% identificacion de los imperialistas
[val, pos_pa]=sort(makespan);
cantImper=round(porcen_imperialistas*cantPais);

%% calcula costos y poder normalizados
[cos_nor , pod_nor] = costo_normalizado(cantImper,val);
clear val
%% calcula cantidad de colonias
cant_Colo=cantidad_colonias(cantImper,cantPais,pod_nor);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% **PASO 2 **
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

%% asignación de colonias
Imper= asignacion_colonias(cantImper,pos_pa,cant_Colo);
%load('test_imper.mat')
%Imper=test_Imper;

%% calculo de costo total del imperio
% costo_Imper=zeros(cantImper,1);
% for k=1:cantImper
%
costo_Imper(k,1)=makespan(Imper{k}(1,1))+par_chi*mean(makespan(Imper{k}(2
:end)));
% end

%% asimilacion
Iter=0;

while (size(Imper,1)>1)

[Paises, makespan]= asimilacion(cantImper, cantMaqu, cantTrab,
cantEtap,pos_pa,cant_Colo,Imper,Paises, costos_sepa,makespan);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PASO 3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Compartir informacion entre colonias
[makespan , Paises]=compartir_colonias(costos_sepa, makespan, Paises,
par_compartir, cantImper, cantEtap, cantMaqu, cantTrab, cant_Colo,
Imper);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PASO 4
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% revolucion
[makespan , Paises]=revolucion(costos_sepa, cantEtap, makespan, Paises,
cantImper, cant_Colo, par_revolucion, Imper,cantMaqu, cantTrab);

%% recalculation imperialista
for k=1:cantImper
[val , pos]=min(makespan(Imper{k}));

```

```

    aux_temp=Imper{k}(1,1);
    Imper{k}(1,1)=Imper{k}(pos,1);
    Imper{k}(pos,1)=aux_temp;
end

%% recalcu lo costo imperio
costo_Imper=zeros(cantImper,1);
for k=1:cantImper

costo_Imper(k,1)=makespan(Imper{k}(1,1))+par_chi*mean(makespan(Imper{k}(2
:end)));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PASO 5 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% guerra entre imperios
[val, pos] = max(costo_Imper);
diffcost=max(costo_Imper)-costo_Imper;
number_C= (diffcost/(sum(diffcost)))*size(Imper{pos},1);
[val1 , pos1]=sort(number_C,'descend');
vall=vall(1:end-1);
pos1=pos1(1:end-1);
vall=max(round(vall),1);
if(sum(vall) < size(Imper{pos},1))
    vall(1,1)=vall(1,1)+size(Imper{pos},1)-sum(vall);
end
% asignacion de colonias
pos_asig = randperm(size(Imper{pos},1));
k=1;
asignados=0;
while (asignados<size(Imper{pos},1))
    cant_a_asig=min(size(Imper{pos},1)-asignados,vall(k));

Imper{pos1(k)}(end+1:end+cant_a_asig)=Imper{pos}(pos_asig(k:k+cant_a_asig
-1));
    cant_Colo(pos1(k))=cant_Colo(pos1(k))+cant_a_asig;
    asignados=asignados+cant_a_asig;
    k=k+1;
end
Imper(pos,:)=[];
cant_Colo(pos)=0;
cant_Colo=cant_Colo(cant_Colo>0);
cantImper=size(Imper,1);
Iter =Iter+1;
end
toc
[cos_bueno , pais_buen]=min(makespan)
Paises(pais_buen,:)
Iter

```

- Funciones de apoyo a la función principal

- Función separación de costos.

Separa los costos de la matriz de entrada con los datos de tiempos de alistamiento y procesamiento para las diferentes máquinas y etapas.

```
function [costos_sepa]=separa_costos(costos,cantEtap, cantTrab,cantMaqu)
costos_sepa=zeros(cantTrab^2+cantTrab,cantEtap,cantMaqu);
for k=1:cantMaqu
    costos_sepa(:, :, k)=costos(k:cantMaqu:end, :);
end
end
```

- Función genera países.

Permite generar la población inicial de la metaheurística, para el caso de este trabajo se genera utilizando el método *FIFO* por secuencia.

```
function [Paises]=genera_paises(cantPais,cantMaqu, cantTrab,
cantEtap,capBuffer,costos_sepa)

%EN PRIMERA INSTANCIA SE GENERAN 5000 PAÍSES PARA COGER DESPUÉS
SELECCIONAR
%LOS MEJORES Y HACER EL ICA CON ELLOS

cantPais_inicial = 5000;

contpais = 1; %Contador que sirve para crear la cantidad de países
necesarios
while contpais ~= cantPais_inicial+1
Paises_iniciales(contpais,2:cantMaqu+cantTrab)=randperm(cantMaqu+cantTrab
-1); %Asigna una secuencia aleatoria para la primera etapa
posiciones(1)=1;
posiciones(2:cantMaqu)=find(Paises_iniciales(contpais, :))>=cantTrab+1);
    if (posiciones(2)==2 || posiciones(cantMaqu)==cantMaqu+cantTrab ||
posiciones(cantMaqu)-posiciones(2)==1)
        contpais = contpais;
    else
        for contetap = 2:cantEtap
            contmaqu = 1;
            while contmaqu ~= cantMaqu+1 %Este while es únicamente para
determinar la matriz de orden en la etapa anterior
                if contmaqu == cantMaqu
                    i=1;
                    for
k=posiciones(contmaqu)+1:(cantMaqu+cantTrab)*(contetap-1)
                        orden(contmaqu,i)=Paises_iniciales(contpais,k);
                        i=i+1;
                    end
                else
```

```

        i=1;
        for k=posiciones (contmaqu)+1:posiciones (contmaqu+1)-1
            orden (contmaqu,i)=Paises_iniciales (contpais,k);
            i=i+1;
        end
    end
    contmaqu = contmaqu+1;
end

Paises_iniciales (contpais, (cantTrab+cantMaqu) * (contetap-
1)+1:(cantTrab+cantMaqu) * contetap) = zeros (1,cantTrab+cantMaqu);

%Generamos de manera aleatoria las posiciones de los cambios de
%máquina en la siguiente etapa, garantizando que no se encuentren
%ni en la primera ni última casilla, así como que tampoco queden
%unidos

posiciones (1)=(cantTrab+cantMaqu) * (contetap-1)+1;
posiciones (2:cantMaqu) = randperm (cantTrab-1,cantMaqu-
1)+(cantTrab+2*cantMaqu-(cantMaqu-2))+(cantTrab+cantMaqu) * (contetap-2);
while abs (posiciones (cantMaqu)- posiciones (2)) == 1
    posiciones (2:cantMaqu) = randperm (cantTrab,cantMaqu-
1)+(cantTrab+2*cantMaqu-1)+(cantTrab+cantMaqu) * (contetap-2);
end
posiciones=sort (posiciones);

%Ahora se deben asignar los puntos de cambio de máquina al vector
%del país

i=1;
for k=1:cantMaqu-1
    Paises_iniciales (contpais,posiciones (k+1)) = cantTrab+i;
    i=i+1;
end

%Se llena la segunda etapa con una estrategia FIFO

[fil col]=size (orden);

%Se calcula la capacidad de cada una de las máquinas en la nueva
%etapa

for k=1:cantMaqu
    if k==cantMaqu
        capMaqu (k) = (cantTrab+cantMaqu) * contetap-posiciones (k);
    else
        capMaqu (k) = posiciones (k+1)-posiciones (k)-1;
    end
end

utilMaqu = zeros (1,cantMaqu);

```

```

dispMaqu = capMaqu - utilMaqu;
for j=1:col
    asig = randperm(cantMaqu);
    ordenposi = find(orden(:,j));
    m = zeros(size(ordenposi,1));
    for t=1:size(ordenposi,1)
        k=1;
        f = 0;
        while orden(ordenposi(t),j) ~= 0
            if dispMaqu(asig(k))>0
                c = 0;
                f = f+1;
                for i=1:cantMaqu
                    b = size(find(m(:)==asig(i)),1);
                    if b == 1
                        c=c+1;
                    end
                end
                b = size(find(m(:)==asig(k)),1);
                disp = find(dispMaqu);

                if b==0 || size(disp,2)<=1 || (c)=(cantMaqu-1) &&
f>=size(disp,2))
                    Países_iniciales(contpays,posiciones(asig(k))+utilMaqu(asig(k))+1) =
orden(ordenposi(t),j);
                    utilMaqu(asig(k)) = utilMaqu(asig(k))+1;
                    orden(ordenposi(t),j) = 0;
                    m(t)=asig(k);
                    dispMaqu = capMaqu - utilMaqu;
                    %k = 0;
                    %f = 0;
                    end
                end
                k=k+1;
            end
        end
    end
    contpays = contpays+1;
end
end

%DE AQUÍ EN ADELANTE SE HACE UN PRIMER CÁLCULO DEL MAKESPAN DE LOS 5000
%PAÍSES PARA EMPEZAR EL ICA CON LA CANTIDAD DESEADA Y CON LOS MEJORES
%GENERADOS A PARTIR DEL FIFO

for k=1:cantPais_inicial
    makespan1(k) = 0;

```

```

%Se garantiza que el país que se va a analizar no contenga los
cambios
%de máquina ni al principio ni al final de cada etapa, así como que
%tampoco estén pegados, si es así se considera que ese es un mal país
y
%automáticamente se le castiga con un makespan muy alto.

for contetap=1:cantEtap

    posiciones(contetap,1) = (cantTrab+cantMaqu)*(contetap-1)+1;
    posiciones(contetap,2:cantMaqu) = (contetap-
1)*(cantMaqu+cantTrab)+find(Paises_iniciales(k,(cantTrab+cantMaqu)*(con
tap-1)+1:(cantTrab+cantMaqu)*(contetap))>cantTrab);

    for j=2:cantMaqu
        if posiciones(contetap,j)==posiciones(contetap,1)+1 ||
posiciones(contetap,j)==(cantTrab+cantMaqu)*(contetap) ||
abs(posiciones(contetap,j)-posiciones(contetap,2))== 1
            makespan1(k) = 10000;
            break;
        end
    end

%Se calcula la capacidad de cada máquina en cada etapa, esto es
%necesario para poder recorrer el país de la manera deseada más
%adelante.

if makespan1(k) == 10000
    break;
else
    for contMaqu=1:cantMaqu
        if contMaqu==cantMaqu
            capMaqu(contMaqu,contetap) =
(cantTrab+cantMaqu)*contetap-posiciones(contetap,contMaqu);
        else
            capMaqu(contMaqu,contetap) =
posiciones(contetap,contMaqu+1)-posiciones(contetap,contMaqu)-1;
        end
    end

if contetap == 1
    sumador = 0;
    cont_etapa=0;
    cont_maquina=1;
    for s=1:cantTrab+cantMaqu;
        opti=Paises_iniciales(k,s);
        switch(opti)
            case 0 %% etapa nueva
                cont_etapa=cont_etapa+1;
                cont_maquina=1;
                sumador = 0;

```

```

                                case num2cell(cantTrab+1:cantTrab+cantMaqu-1) %% hubo
cambio de maquina
                                cont_maquina=cont_maquina+1;
                                sumador = 0;
                                case num2cell(1:cantTrab)
                                    if Paises_iniciales(k,s-1) > cantTrab
                                        anterior = 0;
                                        sumador =
sumador+costos_sepa(1+anterior+(cantTrab+1)*(Paises_iniciales(k,s)-
1),cont_etapa,cont_maquina);
                                        else
sumador+costos_sepa(1+Paises_iniciales(k,s-
1)+(cantTrab+1)*(Paises_iniciales(k,s)-1),cont_etapa,cont_maquina);
                                        end
                                end
                                makespan(1,s)=sumador;
                            end
                        end

                    end

                if makespan1(k) == 0

                    makespan(1,s+1:cantEtap*(cantTrab+cantMaqu)) = zeros(1,(cantEtap-
1)*(cantTrab+cantMaqu));
                    buffers = zeros(cantTrab,4,cantEtap-1);

                    for s=1:cantEtap*(cantTrab+cantMaqu);
                        opti=Paises_iniciales(k,s);
                        switch(opti)
                            case 0 %% etapa nueva
                                cont_maquina=1;
                                maquinas(s) = 0;
                                case num2cell(cantTrab+1:cantTrab+cantMaqu-1) %% hubo
cambio de maquina
                                    cont_maquina=cont_maquina+1;
                                    maquinas(s) = 0;
                                    case num2cell(1:cantTrab)
                                        maquinas(s) = cont_maquina;

                                end
                            end
                        end

                    for contetap=2:cantEtap
                        bloqueos = zeros(cantTrab,4);
                        contTrab = 0; %Contador de la cantidad de trabajos que entran
al buffer, se reinicia en cada etapa
                        bloqTrab = 0; %Contador de la cantidad de trabajos que se
bloquean, se reinicia en cada etapa
                        for j=1:max(capMaqu(:,contetap-1))

```

```

        compara = zeros(cantMaqu,4);
        for contMaqu=1:cantMaqu
            if j<=capMaqu(contMaqu,contetap-1)
                compara(contMaqu,1) =
Paises_iniciales(k,posiciones(contetap-1,contMaqu)+j); %Trabajo a
comparar
                    compara(contMaqu,2) =
makespan(posiciones(contetap-1,contMaqu)+j); %Makespan del trabajo a
comparar
                    compara(contMaqu,3) = contMaqu; %Máquina del
trabajo a comparar
                    compara(contMaqu,4) = (contetap-
2)*(cantMaqu+cantTrab)+find(Paises_iniciales(k,(cantTrab+cantMaqu)*(conte
tap-2)+1:(cantTrab+cantMaqu)*(contetap-1))==compara(contMaqu,1));
%Posición del trabajo a comparar
            end
        end
        for contMaqu=1:cantMaqu
            [minimo,pos] = min(compara(:,2)); %Determina el
mínimo makespan de los trabajos a comparar
            while minimo == 0 %Puede que no haya salido un
trabajo de una máquina, pero este valor no sería válido
                compara(pos,2) = 5000; %Para evitar los ceros, a
los que ya se han calculado les ponemos un valor grande
                [minimo,pos] = min(compara(:,2)); %Se calcula el
nuevo mínimo, a fin de ir asignando de manera paulatina los trabajos a
comparar
            end
            if minimo < 5000 %Si el mínimo da 5000, es porque ya
todos los trabajos fueron asignados
                trabajo = compara(pos,1);
                posi = (contetap-
1)*(cantMaqu+cantTrab)+find(Paises_iniciales(k,(cantTrab+cantMaqu)*(conte
tap-1)+1:(cantTrab+cantMaqu)*(contetap))==trabajo); %Posición del trabajo
a asignar en la nueva etapa
                anterior = Paises_iniciales(k,posi-1); %Trabajo
anterior al que se debe asignar en la nueva etapa
                if anterior > cantTrab || anterior == 0 %Si
anterior es un cambio de etapa o cambio de máquina o su makespan ya está
calculado se puede asignar el trabajo a la nueva etapa
                    anterior = 0; %Se pone cero porque en caso de
que sea un cambio de máquina la matriz de costos busca el valor de 0 al
trabajo asignado
                    makespan(posi) =
makespan(compara(pos,4))+costos_sepa(1+anterior+(cantTrab+1)*(Paises_inic
iales(k,posi)-1),contetap,maquinas(posi));
                    compara(pos,2) = 5000; %Asignamos 5000 al
trabajo ya asignado para que se pase al siguiente
                    for c=1:contTrab
                        if buffers(c,1,contetap-1) > 0 &&
buffers(c,4,contetap-1) == 0 && makespan(buffers(c,2,contetap-1)-1) > 0
                            buffers(c,4,contetap-1) =
makespan(buffers(c,2,contetap-1)-1); %Salida del trabajo del buffer

```

```

                                previo =
Paises_iniciales(k,buffers(c,2,contetap-1)-1);
                                makespan(buffers(c,2,contetap-1)) =
buffers(c,4,contetap-
1)+costos_sepa(1+previo+(cantTrab+1)*(Paises_iniciales(k,buffers(c,2,cont
etap-1))-1),contetap,maquinas(buffers(c,2,contetap-1)));
                                end
                                end
                                else
                                if makespan(posi-1) > 0 &&
makespan(compara(pos,4)) >= makespan(posi-1)
                                makespan(posi) =
makespan(compara(pos,4))+costos_sepa(1+anterior+(cantTrab+1)*(Paises_inic
iales(k,posi)-1),contetap,maquinas(posi));
                                compara(pos,2) = 5000;
                                for c=1:contTrab
                                if buffers(c,1,contetap-1) > 0 &&
buffers(c,4,contetap-1) == 0 && makespan(buffers(c,2,contetap-1)-1) > 0
                                buffers(c,4,contetap-1) =
makespan(buffers(c,2,contetap-1)-1); %Salida del trabajo del buffer
                                previo =
Paises_iniciales(k,buffers(c,2,contetap-1)-1);
                                makespan(buffers(c,2,contetap-1))
= buffers(c,4,contetap-
1)+costos_sepa(1+previo+(cantTrab+1)*(Paises_iniciales(k,buffers(c,2,cont
etap-1))-1),contetap,maquinas(buffers(c,2,contetap-1)));
                                end
                                end
                                else
                                if contTrab == 0 %Si es el primer trabajo
que entra al buffer no habrá ningún problema
                                contTrab = contTrab + 1; %Cantidad de
trabajos que entran al buffer respectivo
                                buffers(contTrab,1,contetap-1) =
trabajo; %Trabajo que entra al buffer
                                buffers(contTrab,2,contetap-1) =
posi; %Posición del trabajo que entra al buffer
                                buffers(contTrab,3,contetap-1) =
makespan(compara(pos,4)); %Llegada del trabajo al buffer
                                if makespan(posi-1) > 0
                                buffers(contTrab,4,contetap-1) =
makespan(posi-1); %Salida del trabajo del buffer
                                makespan(posi) = makespan(posi-
1)+costos_sepa(1+anterior+(cantTrab+1)*(Paises_iniciales(k,posi)-
1),contetap,maquinas(posi));
                                compara(pos,2) = 5000;
                                for c=1:contTrab
                                if buffers(c,1,contetap-1) >
0 && buffers(c,4,contetap-1) == 0 && makespan(buffers(c,2,contetap-1)-1)
> 0
                                buffers(c,4,contetap-1) =
makespan(buffers(c,2,contetap-1)-1); %Salida del trabajo del buffer
                                previo =
Paises_iniciales(k,buffers(c,2,contetap-1)-1);

```

```

makespan (buffers (c,2,contetap-1)) = buffers (c,4,contetap-
1)+costos_sepa (1+previo+(cantTrab+1)*(Paises_iniciales (k,buffers (c,2,cont
etap-1))-1),contetap,maquinas (buffers (c,2,contetap-1)));
        end
    end
    end
else
    utilBuffer = 0;
    t = 1;
    for c=1:contTrab
        if (makespan (compara (pos,4)) <
buffers (c,4,contetap-1)) || (buffers (c,4,contetap-1) == 0 &&
buffers (c,1,contetap-1) > 0)
            utilBuffer = utilBuffer+1;
        end
        if buffers (c,4,contetap-1) > 0
            salida (t) =
buffers (c,4,contetap-1);
        else
            salida (t) = 0;
        end
        t = t+1;
    end
    if utilBuffer < capBuffer &&
size (find (buffers (:,1,contetap-1)==trabajo),1) == 0
        contTrab = contTrab + 1;
        %Cantidad de trabajos que entran al buffer respectivo
        buffers (contTrab,1,contetap-
1) = trabajo; %Trabajo que entra al buffer
        buffers (contTrab,2,contetap-
1) = posi; %Posición del trabajo que entra al buffer (en la nueva etapa)
        buffers (contTrab,3,contetap-
1) = max (min (salida),makespan (compara (pos,4))); %Llegada del trabajo al
buffer
        if makespan (posi-1) > 0
            buffers (contTrab,4,contetap-1) = makespan (posi-1); %Salida del trabajo
del buffer
            makespan (posi) =
makespan (posi-
1)+costos_sepa (1+anterior+(cantTrab+1)*(Paises_iniciales (k,posi)-
1),contetap,maquinas (posi));
            compara (pos,2) = 5000;
            for c=1:contTrab
                if
buffers (c,1,contetap-1) > 0 && buffers (c,4,contetap-1) == 0 &&
makespan (buffers (c,2,contetap-1)-1) > 0
                    buffers (c,4,contetap-1) = makespan (buffers (c,2,contetap-1)-1); %Salida
del trabajo del buffer
                    previo =
Paises_iniciales (k,buffers (c,2,contetap-1)-1);

```

```

makespan (buffers (c, 2, contetap-1)) = buffers (c, 4, contetap-
1)+costos_sepa (1+previo+(cantTrab+1)*(Paises_iniciales (k, buffers (c, 2, cont
etap-1))-1), contetap, maquinas (buffers (c, 2, contetap-1)));
end
end
end
else
bloqTrab = bloqTrab + 1;
bloqueos (bloqTrab, 1) =
trabajo; %Trabajo que se bloquea
bloqueos (bloqTrab, 2) =
(contetap-
2)*(cantMaqu+cantTrab)+find (Paises_iniciales (k, (cantTrab+cantMaqu)*(conte
tap-2)+1:(cantTrab+cantMaqu)*(contetap-1)) == trabajo); %Posición del
trabajo que se bloquea (en la etapa anterior)
bloqueos (bloqTrab, 3) =
makespan (compara (pos, 4)); %Momento en el que el trabajo se bloquea
if makespan (posi-1) > 0 &&
makespan (posi-1) <= buffers (contTrab, 4, contetap-1)
bloqueos (bloqTrab, 4) =
makespan (posi-1); %Momento en el que el trabajo se desbloquea
dif = makespan (posi-1) -
makespan (compara (pos, 4));
c = 0;
i =
Paises_iniciales (k, bloqueos (bloqTrab, 2));
%El siguiente while es
para
%actualizar el makespan
en
%la etapa anterior debido
%al bloqueo.
while i ~= 0 && i <=
cantTrab
if c == 0 || contetap
== 2
makespan (bloqueos (bloqTrab, 2)+c) = makespan (bloqueos (bloqTrab, 2)+c)+dif;
else
dif =
makespan (bloqueos (bloqTrab, 2)+c)-makespan (bloqueos (bloqTrab, 2)+c-1);
if dif < 0
makespan (bloqueos (bloqTrab, 2)+c) = makespan (bloqueos (bloqTrab, 2)+c) +
dif*(-1);
end
end
c = c+1;
i =
Paises_iniciales (k, bloqueos (bloqTrab, 2)+c);
end
makespan (posi-
makespan (posi) =

```

```

1)+costos_sepa(1+anterior+(cantTrab+1)*(Paises_iniciales(k,posi)-
1),contetap,maquinas(posi));
compara(pos,2) = 5000;
for c=1:contTrab
    if
buffers(c,1,contetap-1) > 0 && buffers(c,4,contetap-1) == 0 &&
makespan(buffers(c,2,contetap-1)-1) > 0

buffers(c,4,contetap-1) = makespan(buffers(c,2,contetap-1)-1); %Salida
del trabajo del buffer
Paises_iniciales(k,buffers(c,2,contetap-1)-1);
previo =
makespan(buffers(c,2,contetap-1)) = buffers(c,4,contetap-
1)+costos_sepa(1+previo+(cantTrab+1)*(Paises_iniciales(k,buffers(c,2,cont
etap-1))-1),contetap,maquinas(buffers(c,2,contetap-1)));
end
end
else
    if makespan(posi-1) > 0
        contTrab = contTrab +
1;

buffers(contTrab,1,contetap-1) = trabajo; %Trabajo que entra al buffer
después de un bloqueo

buffers(contTrab,2,contetap-1) = posi; %Posición del trabajo que entra al
buffer

buffers(contTrab,3,contetap-1) = buffers(contTrab-1,4,contetap-1);
%Llegada del trabajo al buffer
dif =
buffers(contTrab,3,contetap-1) - makespan(bloqueos(bloqTrab,2));
c = 0;
i =
Paises_iniciales(k,bloqueos(bloqTrab,2));
%El siguiente while
es para
%actualizar el
makespan en
%la etapa anterior
debido
%al bloqueo.
while i ~= 0 && i <=
cantTrab
    if c == 0 || contetap
== 2
makespan(bloqueos(bloqTrab,2)+c) = makespan(bloqueos(bloqTrab,2)+c)+dif;
else
    dif =
makespan(bloqueos(bloqTrab,2)+c)-makespan(bloqueos(bloqTrab,2)+c-1);
    if dif < 0

```



```

for k=1:cantPais
    [minmakespan mejorpais] = min(makespan1);
    Paises(k,:) = Paises_iniciales(mejorpais,:);
    makespan1(mejorpais) = 10000;
end

end

```

- Funciones de *makespan*.

Son dos funciones, la primera sirve para llamar el cálculo del *makespan* individual de cada uno de los países y la segunda, es la encargada de calcularlo teniendo en cuenta estancias en el *buffer*, bloqueos y demás fenómenos inherentes al problema planteado en esta investigación.

```

function
[makespan1]=calcula_makespan(Paises,cantPais,cantMaqu,cantTrab,cantEtap,c
ostos_sepa,capBuffer)

for k=1:cantPais
    makespan1(k) =
max(makespan_individual(k,Paises,cantMaqu,cantTrab,cantEtap,ostos_sepa,c
apBuffer));
end
end

```

```

function
[makespan]=makespan_individual(k,Paises,cantMaqu,cantTrab,cantEtap,ostos
_sepa,capBuffer)

    makespan(1) = 0;

    %Se garantiza que el país que se va a analizar no contenga los
cambios
    %de máquina ni al principio ni al final de cada etapa, así como que
    %tampoco estén pegados, si es así se considera que ese es un mal país
y
    %automáticamente se le castiga con un makespan muy alto.

    for contetap=1:cantEtap

        posiciones(contetap,1) = (cantTrab+cantMaqu)*(contetap-1)+1;
        posiciones(contetap,2:cantMaqu) = (contetap-
1)*(cantMaqu+cantTrab)+find(Paises(k,(cantTrab+cantMaqu)*(contetap-
1)+1:(cantTrab+cantMaqu)*(contetap))>cantTrab);

        for j=2:cantMaqu
            if posiciones(contetap,j)==posiciones(contetap,1)+1 ||
posiciones(contetap,j)==(cantTrab+cantMaqu)*(contetap) ||
abs(posiciones(contetap,j)-posiciones(contetap,2))== 1

```

```

        makespan(1) = 10000;
        break;
    end
end

%Se calcula la capacidad de cada máquina en cada etapa, esto es
%necesario para poder recorrer el país de la manera deseada más
%adelante.

if makespan(1) == 10000
    break;
else
    for contMaqu=1:cantMaqu
        if contMaqu==cantMaqu
            capMaqu(contMaqu, contetap) =
(cantTrab+cantMaqu)*contetap-posiciones(contetap, contMaqu);
        else
            capMaqu(contMaqu, contetap) =
posiciones(contetap, contMaqu+1)-posiciones(contetap, contMaqu)-1;
        end
    end
end

if contetap == 1
    sumador = 0;
    cont_etapa=0;
    cont_maquina=1;
    for s=1:cantTrab+cantMaqu;
        opti=Paises(k, s);
        switch(opti)
            case 0 %% etapa nueva
                cont_etapa=cont_etapa+1;
                cont_maquina=1;
                sumador = 0;
            case num2cell(cantTrab+1:cantTrab+cantMaqu-1) %% hubo
cambio de maquina
                cont_maquina=cont_maquina+1;
                sumador = 0;
            case num2cell(1:cantTrab)
                if Paises(k, s-1) > cantTrab
                    anterior = 0;
                    sumador =
sumador+costos_sepa(1+anterior+(cantTrab+1)*(Paises(k, s)-
1), cont_etapa, cont_maquina);
                else
                    sumador = sumador+costos_sepa(1+Paises(k, s-
1)+(cantTrab+1)*(Paises(k, s)-1), cont_etapa, cont_maquina);
                end
            end
        end
        makespan(1, s)=sumador;
    end
end
end

```

```

end

if makespan(1) == 0

    makespan(1,s+1:cantEtap*(cantTrab+cantMaqu)) = zeros(1,(cantEtap-
1)*(cantTrab+cantMaqu));
    buffers = zeros(cantTrab,4,cantEtap-1);

    for s=1:cantEtap*(cantTrab+cantMaqu);
        opti=Paises(k,s);
        switch(opti)
            case 0 %% etapa nueva
                cont_maquina=1;
                maquinas(s) = 0;
            case num2cell(cantTrab+1:cantTrab+cantMaqu-1) %% hubo
cambio de maquina
                cont_maquina=cont_maquina+1;
                maquinas(s) = 0;
            case num2cell(1:cantTrab)
                maquinas(s) = cont_maquina;

        end
    end

    for contetap=2:cantEtap
        bloqueos = zeros(cantTrab,4);
        contTrab = 0; %Contador de la cantidad de trabajos que entran
al buffer, se reinicia en cada etapa
        bloqTrab = 0; %Contador de la cantidad de trabajos que se
bloquean, se reinicia en cada etapa
        for j=1:max(capMaqu(:,contetap-1))
            compara = zeros(cantMaqu,4);
            for contMaqu=1:cantMaqu
                if j<=capMaqu(contMaqu,contetap-1)
                    compara(contMaqu,1) =
Paises(k,posiciones(contetap-1,contMaqu)+j); %Trabajo a comparar
                    compara(contMaqu,2) =
makespan(posiciones(contetap-1,contMaqu)+j); %makespan del trabajo a
comparar
                    compara(contMaqu,3) = contMaqu; %Máquina del
trabajo a comparar
                    compara(contMaqu,4) = (contetap-
2)*(cantMaqu+cantTrab)+find(Paises(k,(cantTrab+cantMaqu)*(contetap-
2)+1:(cantTrab+cantMaqu)*(contetap-1))==compara(contMaqu,1)); %Posición
del trabajo a comparar
                end
            end
            for contMaqu=1:cantMaqu
                [minimo,pos] = min(compara(:,2)); %Determina el
mínimo makespan de los trabajos a comparar

```

```

        while minimo == 0 %Puede que no haya salido un
trabajo de una máquina, pero este valor no sería válido
            compara(pos,2) = 5000; %Para evitar los ceros, a
los que ya se han calculado les ponemos un valor grande
            [minimo,pos] = min(compara(:,2)); %Se calcula el
nuevo mínimo, a fin de ir asignando de manera paulatina los trabajos a
comaparar
        end
        if minimo < 5000 %Si el mínimo da 5000, es porque ya
todos los trabajos fueron asignados
            trabajo = compara(pos,1);
            posi = (contetap-
1)*(cantMaqu+cantTrab)+find(Paises(k, (cantTrab+cantMaqu)*(contetap-
1)+1:(cantTrab+cantMaqu)*(contetap))==trabajo); %Posición del trabajo a
asignar en la nueva etapa
            anterior = Paises(k,posi-1); %Trabajo anterior al
que se debe asignar en la nueva etapa
            if anterior > cantTrab || anterior == 0 %Si
anterior es un cambio de etapa o cambio de máquina o su makespan ya está
calculado se puede asignar el trabajo a la nueva etapa
                anterior = 0; %Se pone cero porque en caso de
que sea un cambio de máquina la matriz de costos busca el valor de 0 al
trabajo asignado
                makespan(posi) =
makespan(compara(pos,4))+costos_sepa(1+anterior+(cantTrab+1)*(Paises(k,po
si)-1),contetap,maquinas(posi));
                compara(pos,2) = 5000; %Asignamos 5000 al
trabajo ya asignado para que se pase al siguiente
                for c=1:contTrab
                    if buffers(c,1,contetap-1) > 0 &&
buffers(c,4,contetap-1) == 0 && makespan(buffers(c,2,contetap-1)-1) > 0
                        buffers(c,4,contetap-1) =
makespan(buffers(c,2,contetap-1)-1); %Salida del trabajo del buffer
                        previo =
Paises(k,buffers(c,2,contetap-1)-1);
                        makespan(buffers(c,2,contetap-1)) =
buffers(c,4,contetap-
1)+costos_sepa(1+previo+(cantTrab+1)*(Paises(k,buffers(c,2,contetap-1))-
1),contetap,maquinas(buffers(c,2,contetap-1)));
                    end
                end
            else
                if makespan(posi-1) > 0 &&
makespan(compara(pos,4)) >= makespan(posi-1)
                    makespan(posi) =
makespan(compara(pos,4))+costos_sepa(1+anterior+(cantTrab+1)*(Paises(k,po
si)-1),contetap,maquinas(posi));
                    compara(pos,2) = 5000;
                    for c=1:contTrab
                        if buffers(c,1,contetap-1) > 0 &&
buffers(c,4,contetap-1) == 0 && makespan(buffers(c,2,contetap-1)-1) > 0
                            buffers(c,4,contetap-1) =
makespan(buffers(c,2,contetap-1)-1); %Salida del trabajo del buffer

```

```

                                previo =
Paises(k,buffers(c,2,contetap-1)-1);                                makespan(buffers(c,2,contetap-1))
= buffers(c,4,contetap-1)+costos_sepa(1+previo+(cantTrab+1)*(Paises(k,buffers(c,2,contetap-1))-1),contetap,maquinas(buffers(c,2,contetap-1)));
                                end
                                end
                                else
                                if contTrab == 0 %Si es el primer trabajo
que entra al buffer no habrá ningún problema
                                contTrab = contTrab + 1; %Cantidad de
trabajos que entran al buffer respectivo
                                buffers(contTrab,1,contetap-1) =
trabajo; %Trabajo que entra al buffer
                                buffers(contTrab,2,contetap-1) =
posi; %Posición del trabajo que entra al buffer
                                buffers(contTrab,3,contetap-1) =
makespan(compara(pos,4)); %Llegada del trabajo al buffer
                                if makespan(posi-1) > 0
                                buffers(contTrab,4,contetap-1) =
makespan(posi-1); %Salida del trabajo del buffer
                                makespan(posi) = makespan(posi-
1)+costos_sepa(1+anterior+(cantTrab+1)*(Paises(k,posi)-
1),contetap,maquinas(posi));
                                compara(pos,2) = 5000;
                                for c=1:contTrab
                                if buffers(c,1,contetap-1) >
0 && buffers(c,4,contetap-1) == 0 && makespan(buffers(c,2,contetap-1)-1)
> 0
                                buffers(c,4,contetap-1) =
makespan(buffers(c,2,contetap-1)-1); %Salida del trabajo del buffer
                                previo =
Paises(k,buffers(c,2,contetap-1)-1);
                                makespan(buffers(c,2,contetap-1)) = buffers(c,4,contetap-
1)+costos_sepa(1+previo+(cantTrab+1)*(Paises(k,buffers(c,2,contetap-1))-
1),contetap,maquinas(buffers(c,2,contetap-1)));
                                end
                                end
                                end
                                else
                                utilBuffer = 0;
                                t = 1;
                                for c=1:contTrab
                                if (makespan(compara(pos,4)) <
buffers(c,4,contetap-1)) || (buffers(c,4,contetap-1) == 0 &&
buffers(c,1,contetap-1) > 0)
                                utilBuffer = utilBuffer+1;
                                end
                                if buffers(c,4,contetap-1) > 0
                                salida(t) =
buffers(c,4,contetap-1);
                                else

```

```

        salida(t) = 0;
    end
    t = t+1;
end
    if utilBuffer < capBuffer &&
size(find(buffer(:,1,contetap-1)==trabajo),1) == 0
        contTrab = contTrab + 1;
%Cantidad de trabajos que entran al buffer respectivo
        buffers(contTrab,1,contetap-
1) = trabajo; %Trabajo que entra al buffer
        buffers(contTrab,2,contetap-
1) = posi; %Posición del trabajo que entra al buffer (en la nueva etapa)
        buffers(contTrab,3,contetap-
1) = max(min(salida),makespan(compara(pos,4))); %Llegada del trabajo al
buffer
        if makespan(posi-1) > 0

buffers(contTrab,4,contetap-1) = makespan(posi-1); %Salida del trabajo
del buffer

        makespan(posi) =
makespan(posi-1)+costos_sepa(1+anterior+(cantTrab+1)*(Paises(k,posi)-
1),contetap,maquinas(posi));

        compara(pos,2) = 5000;
        for c=1:contTrab
            if
buffers(c,1,contetap-1) > 0 && buffers(c,4,contetap-1) == 0 &&
makespan(buffers(c,2,contetap-1)-1) > 0

buffers(c,4,contetap-1) = makespan(buffers(c,2,contetap-1)-1); %Salida
del trabajo del buffer

                previo =
Paises(k,buffers(c,2,contetap-1)-1);

makespan(buffers(c,2,contetap-1)) = buffers(c,4,contetap-
1)+costos_sepa(1+previo+(cantTrab+1)*(Paises(k,buffers(c,2,contetap-1))-
1),contetap,maquinas(buffers(c,2,contetap-1)));
            end
        end
    end
else
        bloqTrab = bloqTrab + 1;
        bloqueos(bloqTrab,1) =

trabajo; %Trabajo que se bloquea

        bloqueos(bloqTrab,2) =
(contetap-
2)*(cantMaqu+cantTrab)+find(Paises(k,(cantTrab+cantMaqu)*(contetap-
2)+1:(cantTrab+cantMaqu)*(contetap-1))==trabajo); %Posición del trabajo
que se bloquea (en la etapa anterior)

        bloqueos(bloqTrab,3) =
makespan(compara(pos,4)); %Momento en el que el trabajo se bloquea
        if makespan(posi-1) > 0 &&
makespan(posi-1) <= buffers(contTrab,4,contetap-1)
            bloqueos(bloqTrab,4) =
makespan(posi-1); %Momento en el que el trabajo se desbloquea
        end
    end
end

```



```

buffers(contTrab,1,contetap-1) = trabajo; %Trabajo que entra al buffer
después de un bloqueo

buffers(contTrab,2,contetap-1) = posi; %Posición del trabajo que entra al
buffer

buffers(contTrab,3,contetap-1) = buffers(contTrab-1,4,contetap-1);
%Llegada del trabajo al buffer
                                dif =
buffers(contTrab,3,contetap-1) - makespan(bloqueos(bloqTrab,2));
                                c = 0;
                                i =
Paises(k,bloqueos(bloqTrab,2));
                                %El siguiente while
es para
                                %actualizar el
makespan en
                                %la etapa anterior
debido
                                %al bloqueo.
                                while i ~= 0 && i <=
cantTrab
                                if c == 0 || contetap
== 2
                                makespan(bloqueos(bloqTrab,2)+c) = makespan(bloqueos(bloqTrab,2)+c)+dif;
                                else
                                dif =
makespan(bloqueos(bloqTrab,2)+c)-makespan(bloqueos(bloqTrab,2)+c-1);
                                if dif < 0
makespan(bloqueos(bloqTrab,2)+c) = makespan(bloqueos(bloqTrab,2)+c) +
dif*(-1);
                                end
                                end
                                c = c+1;
                                i =
Paises(k,bloqueos(bloqTrab,2)+c);
                                end

buffers(contTrab,4,contetap-1) = makespan(posi-1); %Salida del trabajo
del buffer
                                makespan(posi) =
makespan(posi-1)+costos_sepa(1+anterior+(cantTrab+1)*(Paises(k,posi)-
1),contetap,maquinas(posi));
                                compara(pos,2) =
5000;
                                for c=1:contTrab
                                if
buffers(c,1,contetap-1) > 0 && buffers(c,4,contetap-1) == 0 &&
makespan(buffers(c,2,contetap-1)-1) > 0

```



segunda, llamada “asignar colonias” determina exactamente y con base en “cantidad colonias” cuáles países van para cada imperio.

```
function [cant_Colo]=cantidad_colonias(cantImper,cantPais,pod_nor)
cant_Colo=ones(cantImper,1);
sum_aux=sum(cant_Colo);
sum_cant_Colo=sum(cant_Colo);
k=1;
while(sum_cant_Colo <= (cantPais-cantImper) && k <= cantImper)
    cant_Colo(k)=cant_Colo(k)+ min(round((cantPais-cantImper-
sum_aux)*pod_nor(k,1)), (cantPais-cantImper-sum_cant_Colo));
    sum_cant_Colo=sum_cant_Colo+cant_Colo(k)-1;
    k=k+1;
end
clear sum_aux
clear sum_cant_Colo
end
```

```
function [Imper]= asignacion_colonias(cantImper,pos_pa,cant_Colo)
Imper=cell(cantImper,1);
asig_aux=pos_pa(cantImper+1:end,1);
a_aux=randperm(length(asig_aux));
asig_aux=asig_aux(a_aux);
Imper=cell(cantImper,1);
ind_aux=1;
for k=1:cantImper
    Imper{k}=[pos_pa(k,1) ; asig_aux(ind_aux:ind_aux+cant_Colo(k,1)-1)];
    ind_aux=ind_aux+cant_Colo(k);
end
clear asig_aux
clear a_aux
end
```

- Función de asimilación del imperialista por parte de las colonias

Permite iniciar el proceso de asimilación en cada uno de los imperios.

```
function [Paises , makespan] = asimilacion(cantImper, cantMaqu, cantTrab,
cantEtap,pos_pa,cant_Colo,Imper,Paises,costos_sepa,makespan, capBuffer)
mascaras_binarias=round(rand(cantImper, ((cantMaqu+cantTrab)*cantEtap)));
copias=Paises(pos_pa(1:cantImper),:).*mascaras_binarias;
for k=1:cantImper
    colo_asimi=(copias(k,:) '*ones(1,cant_Colo(k,1))')';
    pos=find(copias(k,)==0);
    mis_paises=Paises(Imper{k},:);
    for h=1:cant_Colo(k,1)
        for et=1:cantEtap
```

```

        mi_pos=pos(pos<=et*(cantMaqu+cantTrab)); % posiciones que hace
falta completas
        mi_pos=mi_pos(mi_pos >(et-1)*(cantMaqu+cantTrab));
        mi_pais=mis_paises(h+1,:);
        tengo= find(colo_asimi(h,((et-
1)*(cantMaqu+cantTrab)+1:(et)*(cantMaqu+cantTrab)))>0);
        tengo1=colo_asimi(h,tengo+(et-1)*(cantMaqu+cantTrab)); %% los
valores que ya tengo
        ind_mis_pos=2;
        ss=2;
        while (ss<=cantMaqu+cantTrab && ind_mis_pos <=size(mi_pos,2))
            if(find(tengo1==mi_pais(ss+(et-1)*(cantMaqu+cantTrab))))
                ss=ss+1;
            else
                colo_asimi(h,mi_pos(ind_mis_pos))=mi_pais(ss +(et-
1)*(cantMaqu+cantTrab));
                ind_mis_pos=ind_mis_pos+1;
                ss=ss+1;
            end
        end
        end
        end
        end
        Países(Imper{k}(2:end,1),:)=colo_asimi;
        % actualiza costos de asimilados

        for m=2:cant_Colo(k)+1

makespan(Imper{k}(m,1),1)=max(makespan_individual(Imper{k}(m,1),Países,ca
ntMaqu,cantTrab,cantEtap,costos_sepa,capBuffer));
        end

end
clear pos ; clear mis_paises ; clear mi_pos ; clear mi_pais; clear
ind_mis_pos;
clear ss ; clear colo_asimi ; clear mascarar_binarias ; clear copias ;
end

```

- **Función compartir información entre colonias**

Dependiendo del parámetro de entrada, esta función permite determinar cuáles colonias dentro de cada imperio compartirán información entre ellas e inicia el proceso para que se realice el intercambio de datos.

```

function [makespan , Países]=compartir_colonias(costos_sepa, makespan,
Países, par_compartir, cantImper, cantEtap, cantMaqu, cantTrab,
cant_Colo, Imper,capBuffer)
for k=1:cantImper

```

```

    if(cant_Colo(k,1)>=2) % verifica que haya al menos dos colonias para
    compartir info
        cant_comp=max(2,2*round(cant_Colo(k,1)*par_compartir/2));
        for m=1:cant_comp/2
            pos=randperm(size(Imper{k},1)-1)+1;
            pos_coll=pos(1,1);
            pos_col2=pos(1,2);
            colo_aux1=zeros(size(Paises(1,:)));
            colo_aux2=zeros(size(Paises(1,:)));
            for s=1:cantEtap
                % elige el punto para compartir
                pun_com=round((cantMaqu+cantTrab-1)*rand(1,1)+1)+(s-
1)*(cantMaqu+cantTrab);
                colo_aux1((s-
1)*(cantMaqu+cantTrab)+1:pun_com)=Paises(Imper{k}(pos_coll,1),(s-
1)*(cantMaqu+cantTrab)+1:pun_com);
                colo_aux2((s-
1)*(cantMaqu+cantTrab)+1:pun_com)=Paises(Imper{k}(pos_col2,1),(s-
1)*(cantMaqu+cantTrab)+1:pun_com);
                ind_pos=2+(s-1)*(cantMaqu+cantTrab);
                ind_col=pun_com+1;
                %COLONIA 1
                while (ind_col<=s*(cantMaqu+cantTrab))
                    if(find(colo_aux1((s-
1)*(cantMaqu+cantTrab)+1:pun_com))==Paises(Imper{k}(pos_col2,1),ind_pos))
                        ind_pos=ind_pos+1;
                    else
                        colo_aux1(1,ind_col)=Paises(Imper{k}(pos_col2,1),ind_pos);
                        ind_pos=ind_pos+1;
                        ind_col=ind_col+1;
                    end
                end
                %COLONIA 2
                ind_pos=2+(s-1)*(cantMaqu+cantTrab);
                ind_col=pun_com+1;
                while (ind_col<=s*(cantMaqu+cantTrab))
                    if(find(colo_aux2((s-
1)*(cantMaqu+cantTrab)+1:pun_com))==Paises(Imper{k}(pos_coll,1),ind_pos))
                        ind_pos=ind_pos+1;
                    else
                        colo_aux2(1,ind_col)=Paises(Imper{k}(pos_coll,1),ind_pos);
                        ind_pos=ind_pos+1;
                        ind_col=ind_col+1;
                    end
                end
            end
            % calcula makespan y elige los dos menores
            aux_cos=[ makespan(Imper{k}(pos_coll,1)) ;
makespan(Imper{k}(pos_col2,1)) ;
max(makespan_individual(1,colo_aux1,cantMaqu,cantTrab,cantEtap,costos_sep
a,capBuffer));
max(makespan_individual(1,colo_aux2,cantMaqu,cantTrab,cantEtap,costos_sep
a,capBuffer))];
            [val_1 , pos_1]=sort(aux_cos);

```

```

    opti=pos_1(1,1);
    opti2=pos_1(2,1);
    tempo_re=Paises(Imper{k}(pos_coll,1),:);
    switch(opti)
        case 2
            Paises(Imper{k}(pos_coll,1),:)=Paises(Imper{k}(pos_col2,1),:);
            makespan(Imper{k}(pos_coll,1),1)=val_1(1,1);
            break;
        case 3
            Paises(Imper{k}(pos_coll,1),:)=colo_aux1;
            makespan(Imper{k}(pos_coll,1),1)=val_1(1,1);
            break;
        case 4
            Paises(Imper{k}(pos_coll,1),:)=colo_aux2;
            makespan(Imper{k}(pos_coll,1),1)=val_1(1,1);
            break;
    end

    switch(opti2)
        case 1
            Paises(Imper{k}(pos_col2,1),:)=tempo_re;
            makespan(Imper{k}(pos_col2,1),1)=val_1(2,1);
            break;
        case 3
            Paises(Imper{k}(pos_col2,1),:)=colo_aux1;
            makespan(Imper{k}(pos_col2,1),1)=val_1(2,1);
            break;
        case 4
            Paises(Imper{k}(pos_col2,1),:)=colo_aux2;
            makespan(Imper{k}(pos_col2,1),1)=val_1(2,1);
    end
end
end
end
end

```

- Función de revolución dentro de las colonias.

Con base en el parámetro de entrada, determina cuáles colonias se van a revolucionar en cada imperio y luego aplica los operadores genéticos para realizar ese procedimiento.

```

function [makespan, Paises]= revolucion(costos_sepa, cantEtap, makespan,
Paises, cantImper, cant_Colo, par_revolucion, Imper,cantMaqu,
cantTrab,capBuffer)
for k=1:cantImper
    cant_revo=max(1,round(cant_Colo(k,1)*par_revolucion)); %% define
    cuantas colonias se van a revolucionar de dicho imperio

```

```

pos = (randperm(cant_Colo(k,1))+1);
pos= pos(1,1:cant_revo);
for s=1:cant_revo % recorre las colonias que se revolucionaran
    mi_colonia=Paises(Imper{k}(pos(1,s)),:);
    for et=1:cantEtap
        pos1=randperm(cantMaqu+cantTrab-1)+1+(et-
1)*(cantMaqu+cantTrab);
        pos1=pos1(1:2);
        aux_tem=mi_colonia(1,pos1(1,1));
        mi_colonia(1,pos1(1,1))=mi_colonia(1,pos1(1,2));
        mi_colonia(1,pos1(1,2))=aux_tem;
    end
    Paises(Imper{k}(pos(1,s)),:)=mi_colonia;

makespan(Imper{k}(pos(1,s)),1)=max(makespan_individual(Imper{k}(pos(1,s))
,Paises,cantMaqu,cantTrab,cantEtap,costos_sepa,capBuffer));
end
end
end
end

```

- **Función de guerra entre imperios**

Desarrolla la guerra entre los imperios, donde los más fuertes tomarán las colonias del más débil hasta que un imperio tenga todas las colonias y terminen las iteraciones del algoritmo.

```

function [Imper, cant_Colo, cantImper] = guerra_imperios(costo_Imper,
Imper, cant_Colo)
[val, pos] = max(costo_Imper);
diffcost=max(costo_Imper)-costo_Imper;
number_C= (diffcost/(sum(diffcost)))*size(Imper{pos},1);
[vall , pos1]=sort(number_C,'descend');
vall=vall(1:end-1);
pos1=pos1(1:end-1);
vall=max(round(vall),1);
if(sum(vall) < size(Imper{pos},1))
    vall(1,1)=vall(1,1)+size(Imper{pos},1)-sum(vall);
end
% asignacion de colonias
pos_asig = randperm(size(Imper{pos},1));
k=1;
asignados=0;
while (asignados<size(Imper{pos},1))
    cant_a_asig=min(size(Imper{pos},1)-asignados,vall(k));

Imper{pos1(k)}(end+1:end+cant_a_asig)=Imper{pos}(pos_asig(k:k+cant_a_asig
-1));
    cant_Colo(pos1(k))=cant_Colo(pos1(k))+cant_a_asig;
    asignados=asignados+cant_a_asig;
    k=k+1;
end

```

```
end
Imper(pos,:)=[];
cant_Colo(pos)=0;
cant_Colo=cant_Colo(cant_Colo>0);
cantImper=size(Imper,1);
end
```

**Anexo E.** Análisis comparativo entre los resultados obtenidos por el *ICA* para el *HFS* con *buffers* limitados y el *ICA* para el *HFS* con *buffers* ilimitados.

A continuación, se busca realizar un comparativo entre el desempeño de la metaheurística basada en el algoritmo competitivo imperialista para el caso con tamaño de *buffers* limitados y la metaheurística basada en el algoritmo competitivo imperialista para tamaño de *buffers* ilimitados.

El proceso comparativo se realiza mediante una prueba de hipótesis de medias entre el *makespan* de las secuencias generadas por la metaheurística basada en el *ICA* para el *HFS* con *buffers* limitados y el *makespan* de las secuencias obtenidas mediante la misma metaheurística, pero para el *HFS* con *buffers* ilimitados.

Los datos del *makespan* de las secuencias con *buffers* limitados se basan en la combinación de niveles de los parámetros que obtuvo el mejor resultado promedio para cada una de las 20 instancias planteadas durante la validación de la metaheurística del presente trabajo (ver sección 6.1) y se presentan en detalle en el anexo D. Por otra parte, los resultados del *makespan* de las secuencias con *buffers* ilimitados son tomados de la validación de la metaheurística del trabajo de grado presentando por Moratto y Pérez<sup>125</sup> y se utilizan también los resultados con mejor promedio para las combinaciones de tratamientos en las instancias a comparar (para más detalle de los resultados, ver anexo D).

Tanto el presente proyecto como el de Moratto y Pérez<sup>126</sup> buscan dar solución al problema el *HFS* con máquinas paralelas no relacionadas y tiempos de alistamiento dependientes de la secuencia mediante una metaheurística basada en la modificación del algoritmo competitivo imperialista planteada por Rabiee et

---

<sup>125</sup> MORATTO CHIMENTY, ESELENDIS; PÉREZ FIGUEREDO, LUIS EDUARDO. Op. Cit. p. 107-121.

<sup>126</sup> Ibid.

al.<sup>127</sup> Ambas metaheurísticas utilizan los niveles de parámetros presentados en la tabla 6 del presente trabajo y también ambas son codificadas en *Matlab* versión R2012a (ver sección 6.1 y anexo C).

Los tiempos de alistamiento y procesamiento de los trabajos para ambos elementos comparativos de la prueba de hipótesis de medias son los utilizados por Moratto y Pérez<sup>128</sup> que se basan en el artículo de Miranda Lugo et al.<sup>129</sup> y siguen una distribución uniforme entre 75 y 125 para los tiempos de procesamiento y para los tiempos de alistamiento entre 25 y 74. Se presentan en detalle en el anexo B.

Por último, ambos componentes de la prueba de hipótesis fueron corridos en una computadora con procesador *Intel Core i5-4570*, 3.2 GHz con una memoria *RAM* de 8 *Gigabytes* y un sistema operativo de 64 *bits*.

Para hacer la prueba de hipótesis de medias se utiliza el *software* estadístico *Minitab 17*, primero se comprueba si ambas poblaciones tienen varianzas iguales mediante una prueba de hipótesis para la varianza (prueba de 2 varianzas en *Minitab 17*) a un nivel de confianza del 95% para todas las instancias, los resultados de esta prueba se mencionan para cada instancia evaluada. Las hipótesis de dichas pruebas se presentan a continuación:

$$H_0: \sigma_{BL}^2 / \sigma_{BI}^2 = 1$$

$$H_1: \sigma_{BL}^2 / \sigma_{BI}^2 \neq 1$$

Donde  $\sigma_{BL}^2$  es la varianza poblacional del *makespan* obtenido de las respuestas mediante la metaheurística basada en el *ICA* para el problema con *buffers* limitados, mientras  $\sigma_{BI}^2$  es la varianza poblacional del *makespan* obtenido de las

---

<sup>127</sup> RABIEE, M. et al. Op cit, p. 1229-1245.

<sup>128</sup> MORATTO CHIMENTY, ELENDIS; PÉREZ FIGUEREDO, LUIS EDUARDO. Op. Cit. p. 63.

<sup>129</sup> MIRANDA LUGO, P.L. et al. Op. Cit. p.154-165.

soluciones mediante la metaheurística basada en el *ICA* para el problema con *buffers* ilimitados.

Luego de evaluar si las varianzas son iguales, se realiza la prueba de hipótesis de media de la población, para muestras pequeñas ( $n < 30$ ) y varianzas poblacionales desconocidas mediante una prueba *t* de dos muestras (llamada así en *Minitab 17*) con un nivel de confianza del 95% para todas las instancias. Para cada una se enseña el diagrama de caja obtenido y para todas se plantearon las siguientes hipótesis:

$$H_0: \mu_{BL} \geq \mu_{BI}$$

$$H_1: \mu_{BL} < \mu_{BI}$$

Donde  $\mu_{BL}$  es la media poblacional del *makespan* obtenido de las respuestas mediante la metaheurística basada en el *ICA* para el problema con *buffers* limitados, mientras  $\mu_{BI}$  es la media poblacional del *makespan* obtenida de las respuestas mediante la metaheurística basada en el *ICA* para el problema con *buffers* ilimitados.

A continuación, se presenta el detalle de los datos utilizados para las pruebas junto a los resultados obtenidos en las mismas para cada una de las 20 instancias planteadas para el problema del *HFS* con *buffers* de tamaño limitado.

**Tabla 55.** Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia *i7j2k3-1*.

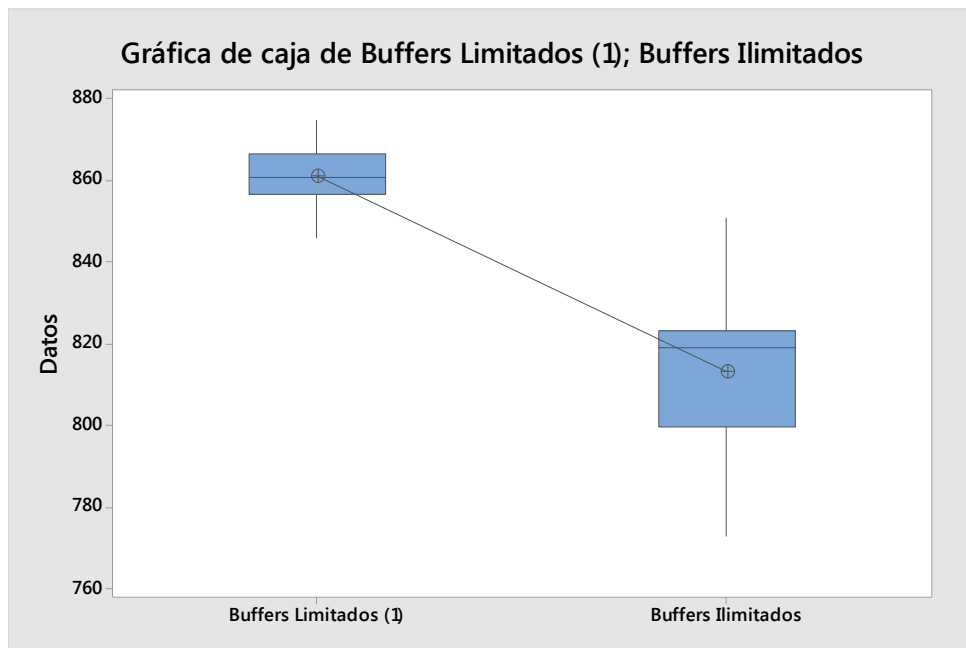
#	CP	%Imp	% $\xi$	%Cr	%Rev	MO [ut]	Med [ut]	SD [ut]
<b>Buffers Limitados (29)</b>	100	0,1	0,5	0,6	0,4	846	861,1	7,752
<b>Buffers Ilimitados (20)</b>	250	0,25	0,5	0,4	0,2	773	813,3	20,833

Para la instancia *i7j2k3-1* (la cual se compara para la prueba de hipótesis de medias con los resultados de la instancia *i7j2k3* de Moratto y Pérez) se obtuvo que

como Valor-P=0,045  $\leq$  0,05 existe evidencia suficiente para rechazar la hipótesis nula por lo tanto se asumen varianzas diferentes.

Los resultados obtenidos para la prueba de hipótesis de medias para la instancia i7j2k3-1 fueron Valor T = 6,80, Valor P = 1,000, GL = 11, por lo tanto, como Valor-P = 1,000  $>$  0,05 no existe evidencia suficiente para rechazar la hipótesis nula y teniendo en cuenta también el diagrama de cajas se demuestra que la metaheurística basada en el *ICA* para el *HFS* con *buffers* limitados tamaño uno obtiene secuencias con *makespan* superiores a los de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado. El incremento se puede deber tanto al desempeño del *ICA* como al impacto de la reducción en el tamaño del *buffer*.

**Figura 14.** Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia i7j2k3-1.



Para el caso de la instancia i7j2k3-2 como hay dos combinaciones de niveles de tratamientos con los mejores resultados se escoge la que mejor respuesta

observada presente (antes era la de mejor promedio). En caso de persistir el empate, se escoge la que menor desviación estándar presente.

**Tabla 56.** Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia i7j2k3-2.

#	CP	%Imp	% ξ	%Cr	%Rev	MO [ut]	Med [ut]	SD [ut]
<b>Buffers Limitados (14)</b>	250	0,1	0,3	0,6	0,4	839	848,7	8,314
<b>Buffers Ilimitados (20)</b>	250	0,25	0,5	0,4	0,2	773	813,3	20,833

Como Valor-P=0,060 > 0,05 no existe evidencia suficiente para rechazar la hipótesis nula por lo tanto se asumen varianzas iguales.

Para el caso de la prueba de hipótesis de medias de la instancia i7j2k3-2 (la cual se compara con los resultados de la instancia i7j2k3 de Moratto y Pérez) los resultados fueron Valor T = 4,99, Valor P = 1,000, GL = 18, por lo tanto, como Valor-P = 1,000 > 0,05 no existe evidencia suficiente para rechazar la hipótesis nula y se comprueba basándose también en el diagrama de cajas que la metaheurística basada en el *ICA* para el *HFS* con *buffers* limitados tamaño dos obtiene secuencias con *makespan* superiores a los de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado.

El incremento se puede deber tanto al desempeño del *ICA* como al impacto de la reducción en el tamaño del *buffer*. El diagrama de cajas de la presente instancia se presenta en la figura 15 de más adelante.

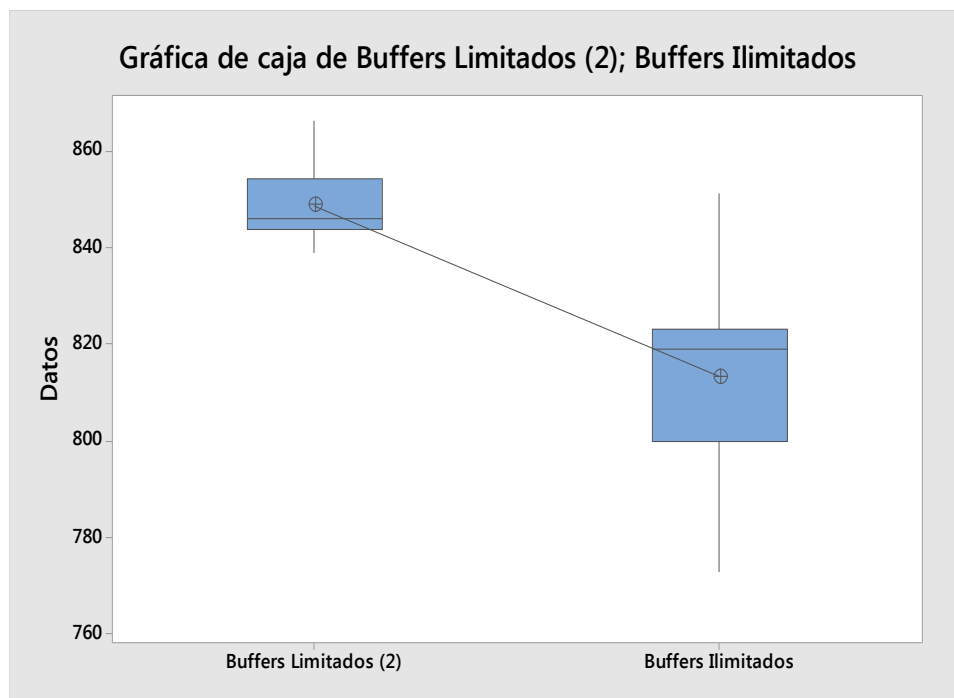
**Tabla 57.** Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia i7j3k3-1.

#	CP	%Imp	% ξ	%Cr	%Rev	MO [ut]	Med [ut]	SD [ut]
<b>Buffers Limitados (25)</b>	100	0,1	0,5	0,4	0,4	705	728	12,02
<b>Buffers Ilimitados (20)</b>	250	0,25	0,5	0,4	0,2	625	669,2	23,43

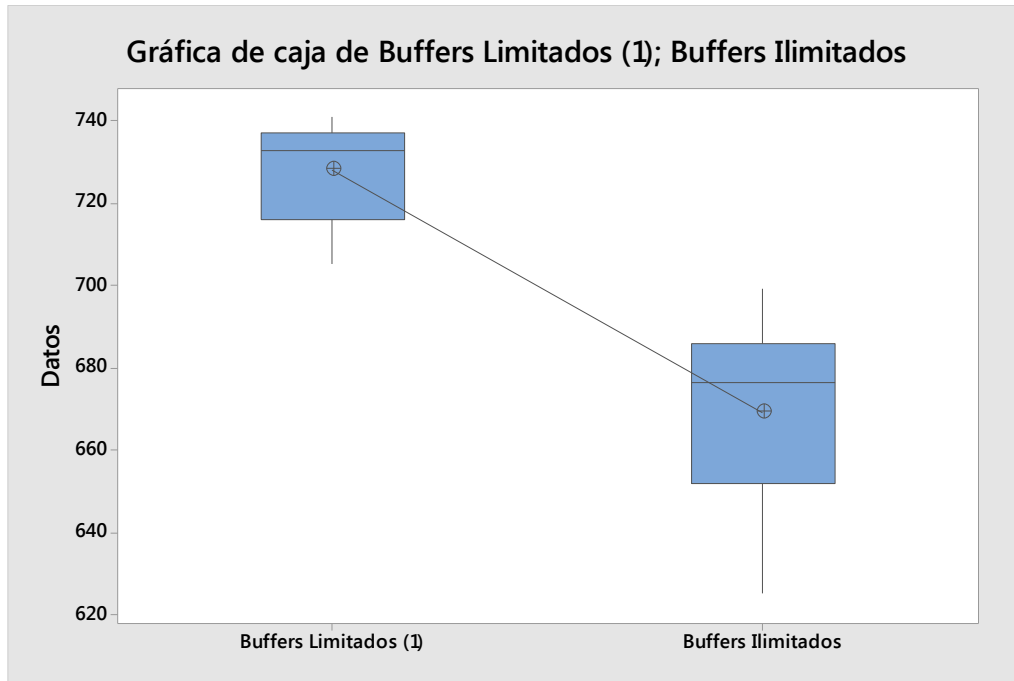
Comenzando con la instancia i7j3k3-1 (la cual se compara para la prueba de hipótesis de medias con las soluciones de la instancia i7j3k3 de Moratto y Pérez) se obtuvo que como Valor-P=0,081 > 0,05 no existe evidencia suficiente para rechazar la hipótesis nula por lo tanto se asumen varianzas iguales.

Mientras tanto, los resultados obtenidos para la prueba de hipótesis de medias para la instancia i7j3k3-1 fueron Valor T = 7,06, Valor P = 1,000, GL = 18, por lo tanto, como Valor-P = 1,000 > 0,05 y teniendo también en cuenta el diagrama de cajas de la figura 16, no existe evidencia suficiente para rechazar la hipótesis nula y en consecuencia demuestra que la metaheurística basada en el ICA para el HFS con *buffers* limitados tamaño uno obtiene secuencias con *makespan* superiores a los de la metaheurística basada en el ICA para el HFS con *buffers* de tamaño ilimitado. El incremento se puede deber tanto al desempeño del ICA como al impacto de la reducción en el tamaño del *buffer*.

**Figura 15.** Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia i7j2k3-2.



**Figura 16.** Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia i7j3k3-1.



**Tabla 58.** Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia i7j3k3-2.

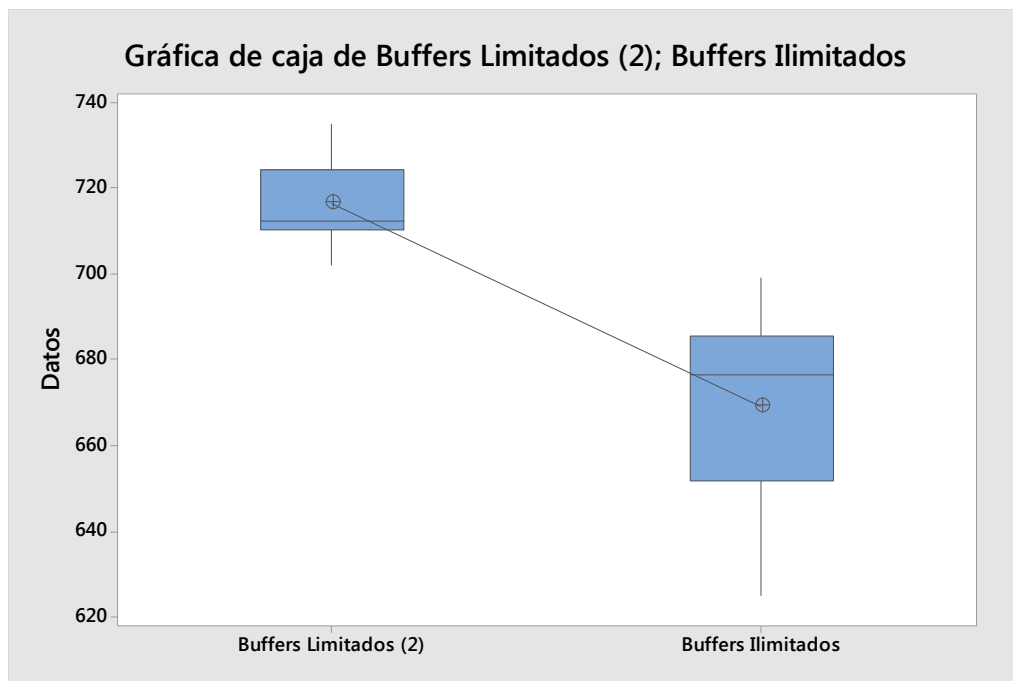
#	CP	%Imp	% ξ	%Cr	%Rev	MO [ut]	Med [ut]	SD [ut]
<b>Buffers limitados (5)</b>	100	0,1	0,3	0,6	0,2	702	716,5	9,8798
<b>Buffers ilimitados (20)</b>	250	0,25	0,5	0,4	0,2	625	669,2	23,43

En la instancia i7j3k3-2 se obtuvo que como Valor-P=0,031  $\leq$  0,05 existe evidencia suficiente para rechazar la hipótesis nula por lo tanto se asumen varianzas diferentes.

Para la prueba de hipótesis de medias de la instancia i7j3k3-2 (la cual se compara con las soluciones de la instancia la i7j3k3 de Moratto y Pérez) los resultados fueron Valor T = 5,88, Valor P = 1,000, GL = 12, por lo tanto, como Valor-P = 1,000  $>$  0,05 no existe evidencia suficiente para rechazar la hipótesis nula y

tomando en cuenta el diagrama de cajas, se demuestra que la metaheurística basada en el *ICA* para el *HFS* con *buffers* limitados tamaño dos obtiene secuencias con *makespan* superiores a los de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado. El incremento se puede deber tanto al desempeño del *ICA* como al impacto de la reducción en el tamaño del *buffer*.

**Figura 17.** Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia *i7j3k3-2*.



**Tabla 59.** Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia *i7j3k3-3*.

#	CP	%Imp	% $\xi$	%Cr	%Rev	MO [ut]	Med [ut]	SD [ut]
<b>Buffers limitados (29)</b>	100	0,1	0,5	0,6	0,4	688	703,7	10,034
<b>Buffers ilimitados (20)</b>	250	0,25	0,5	0,4	0,2	625	669,2	23,43

Para la instancia i7j3k3-3 se obtuvo que como Valor-P=0,033  $\leq$  0,05 existe evidencia suficiente para rechazar la hipótesis nula por lo tanto se asumen varianzas diferentes.

Con la instancia i7j3k3-3 (la cual se compara con las respuestas de la instancia i7j3k3 de Moratto y Pérez) los resultados de la prueba de hipótesis de medias fueron los siguientes, Valor T = 4,28, Valor P = 0,999, GL = 12. Como Valor-P = 0,999  $>$  0,05 no existe evidencia suficiente para rechazar la hipótesis nula y así se comprueba, observando también el diagrama de cajas, que la metaheurística basada en el ICA para el HFS con buffers limitados tamaño tres obtiene secuencias con *makespan* generalmente superiores a los de la metaheurística basada en el ICA para el HFS con buffers de tamaño ilimitado.

El incremento se puede deber tanto al desempeño del ICA como al impacto de la reducción en el tamaño del *buffer*. El diagrama de cajas se presenta en la figura 18 que se encuentra más adelante.

**Tabla 60.** Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con buffers limitados y buffers ilimitados para la instancia i7j2k5-1.

#	CP	%Imp	% $\xi$	%Cr	%Rev	MO [ut]	Med [ut]	SD [ut]
<b>Buffers limitados (1)</b>	100	0,1	0,3	0,4	0,2	1244	1277,1	19,9
<b>Buffers ilimitados (28)</b>	250	0,25	0,5	0,4	0,4	1172	1263,1	57,0

Con la instancia i7j2k5-1 (la cual se compara para la prueba de hipótesis de medias con las respuestas de la instancia i7j2k5 de Moratto y Pérez) se obtuvo que como Valor-P=0,036  $\leq$  0,05 existe evidencia suficiente para rechazar la hipótesis nula por lo tanto se asumen varianzas diferentes.

Para la prueba de hipótesis de medias se obtuvo lo siguiente, Valor T = 0,73, Valor P = 0,761, GL = 11. Como Valor-P = 0,761  $>$  0,05 no existe evidencia suficiente para rechazar la hipótesis nula y así se comprueba, también con base en el diagrama de cajas de la figura 19, que la metaheurística basada en el ICA para el

*HFS* con *buffers* tamaño uno obtiene secuencias con *makespan* prácticamente iguales a los de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado.

Por lo tanto, para la presente instancia el utilizar la metaheurística basada en el *ICA* con *buffers* limitados de tamaño uno o la metaheurística basada en el *ICA* con *buffers* ilimitados podría asegurar resultados de la misma calidad en el *makespan* de las secuencias.

**Tabla 61.** Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia i7j2k5-2.

#	CP	%Imp	% $\xi$	%Cr	%Rev	MO [ut]	Med [ut]	SD [ut]
<b>Buffers limitados (25)</b>	100	0,1	0,5	0,4	0,4	1244	1269,6	15,378
<b>Buffers ilimitados (28)</b>	250	0,25	0,5	0,4	0,4	1172	1263,1	57,0

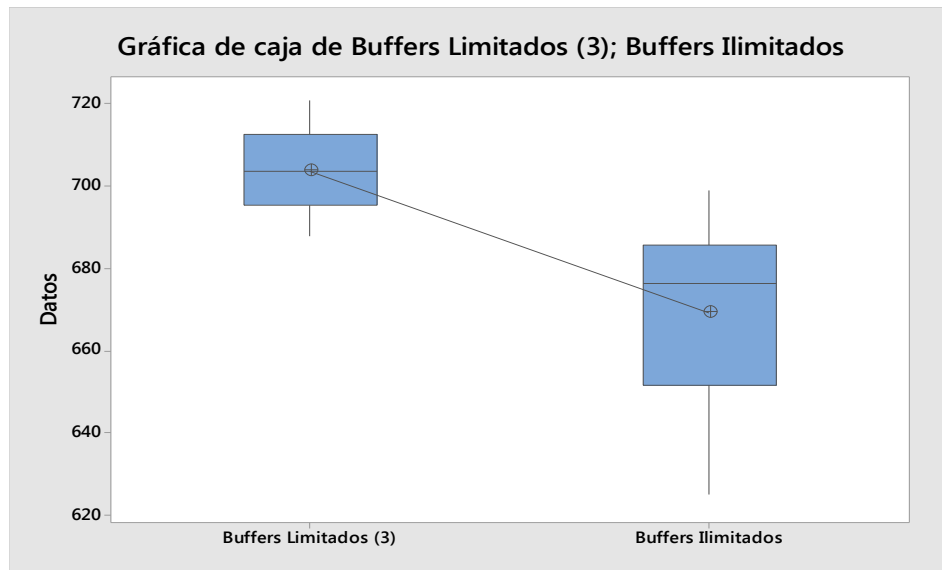
Para la prueba de varianzas de la presente instancia se obtuvo que como Valor-P = 0,012  $\leq$  0,05 existe evidencia suficiente para rechazar la hipótesis nula por lo tanto se asumen varianzas diferentes.

Con el caso de la prueba de hipótesis de medias los resultados de la instancia actual (la cual se compara con las soluciones de la instancia i7j2k5 de Moratto y Pérez) fueron los siguientes, Valor T = 0,35, Valor P = 0,663, GL = 10. Como Valor-P = 0,663 > 0,05 no existe evidencia suficiente para rechazar la hipótesis nula y así se comprueba observando también el diagrama de cajas y parecido como sucede en la instancia anterior que la metaheurística basada en el *ICA* para el *HFS* con *buffers* limitados de tamaño dos obtiene secuencias con *makespan* de la misma calidad a los de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado.

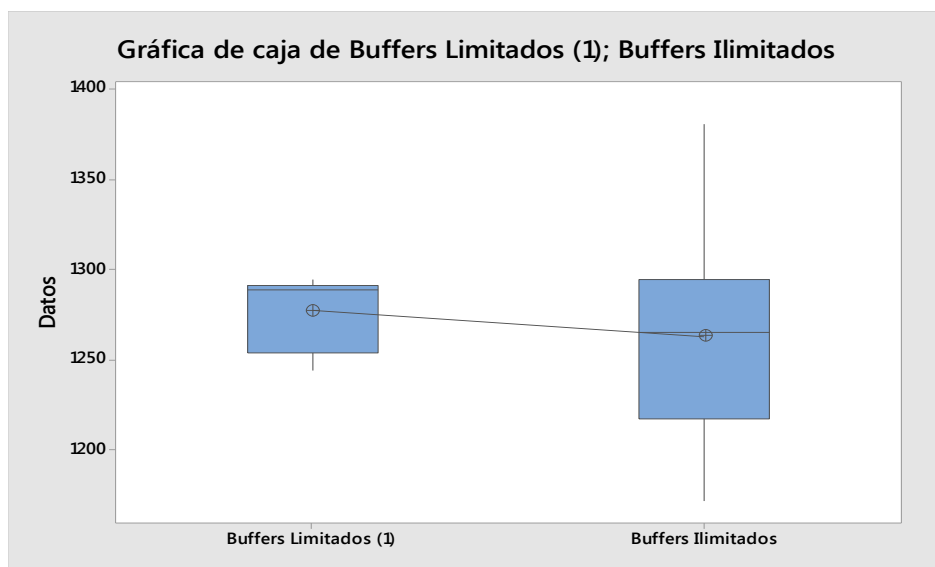
Así, para esta instancia el utilizar la metaheurística basada en el *ICA* con *buffers* tamaño dos o la metaheurística basada en el *ICA* con *buffers* ilimitados podría

asegurar resultados idénticos en cuestión del *makespan*. El diagrama de cajas de la instancia i7j2k5-2 se presenta en la figura 20 de más adelante.

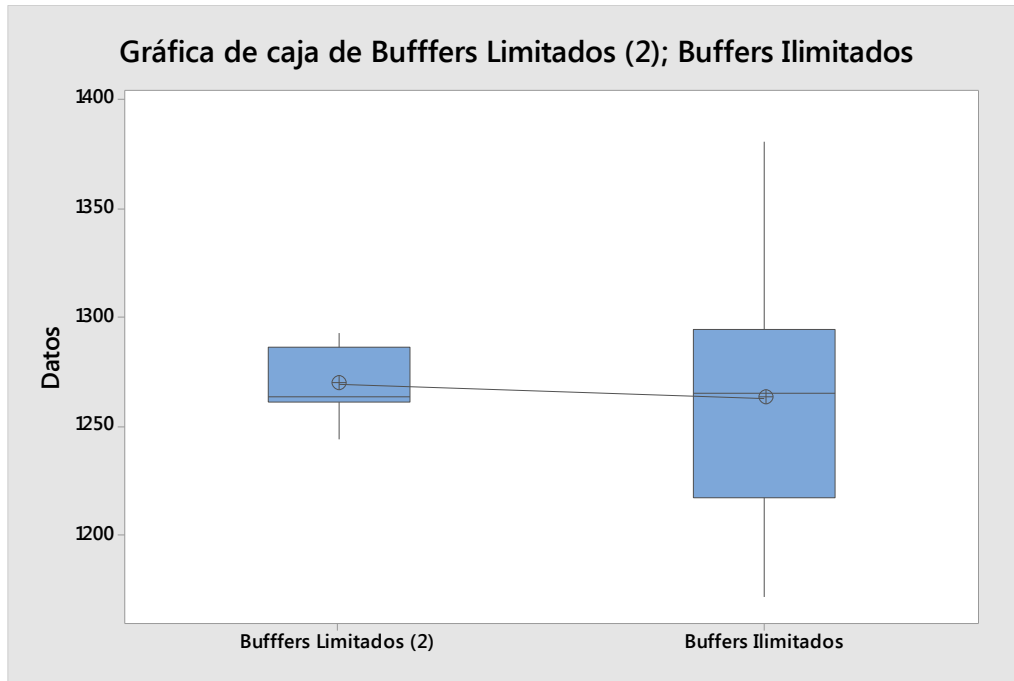
**Figura 18.** Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia i7j3k3-3.



**Figura 19.** Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia i7j2k5-1.



**Figura 20.** Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia i7j2k5-2.



**Tabla 62.** Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia i7j3k5-1.

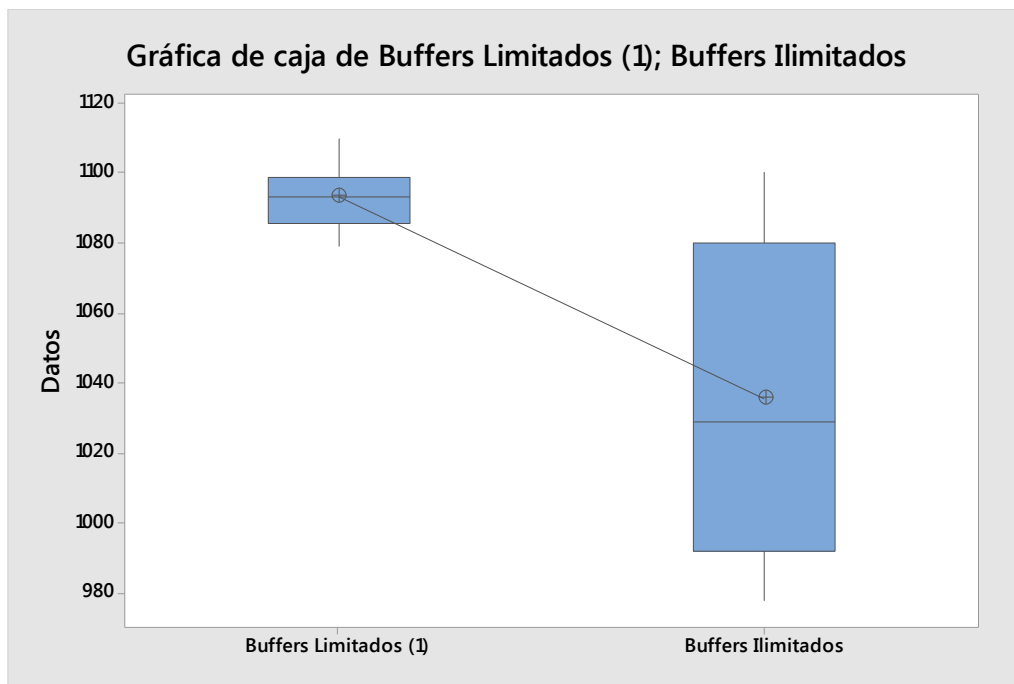
#	CP	%Imp	% ξ	%Cr	%Rev	MO [ut]	Med [ut]	SD [ut]
<b>Buffers limitados (21)</b>	100	0,1	0,5	0,6	0,2	1079	1093,2	10,358
<b>Buffers ilimitados (24)</b>	250	0,25	0,5	0,6	0,2	978	1035,5	46,7

Continuando con la instancia i7j3k5-1 (la cual se compara para la prueba de hipótesis de medias con las respuestas de la instancia i7j3k5 de Moratto y Pérez) se obtuvo que como Valor-P=0,000 ≤ 0,05 existe evidencia suficiente para rechazar la hipótesis nula por lo tanto se asumen varianzas diferentes.

Para la presente instancia los resultados de la prueba de hipótesis de medias fueron Valor T = 3,81, Valor P = 0,998, GL = 9, por lo tanto, como Valor-P = 0,998 > 0,05 no existe evidencia suficiente para rechazar la hipótesis nula y así se

demuestra tomando en cuenta también el diagrama de cajas, que la metaheurística basada en el *ICA* para el *HFS* con *buffers* limitados tamaño uno obtiene secuencias con *makespan* generalmente superiores a los de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado. El incremento se puede deber tanto al desempeño del *ICA* como al impacto de la reducción en el tamaño del *buffer*.

**Figura 21.** Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia *i7j3k5-1*.



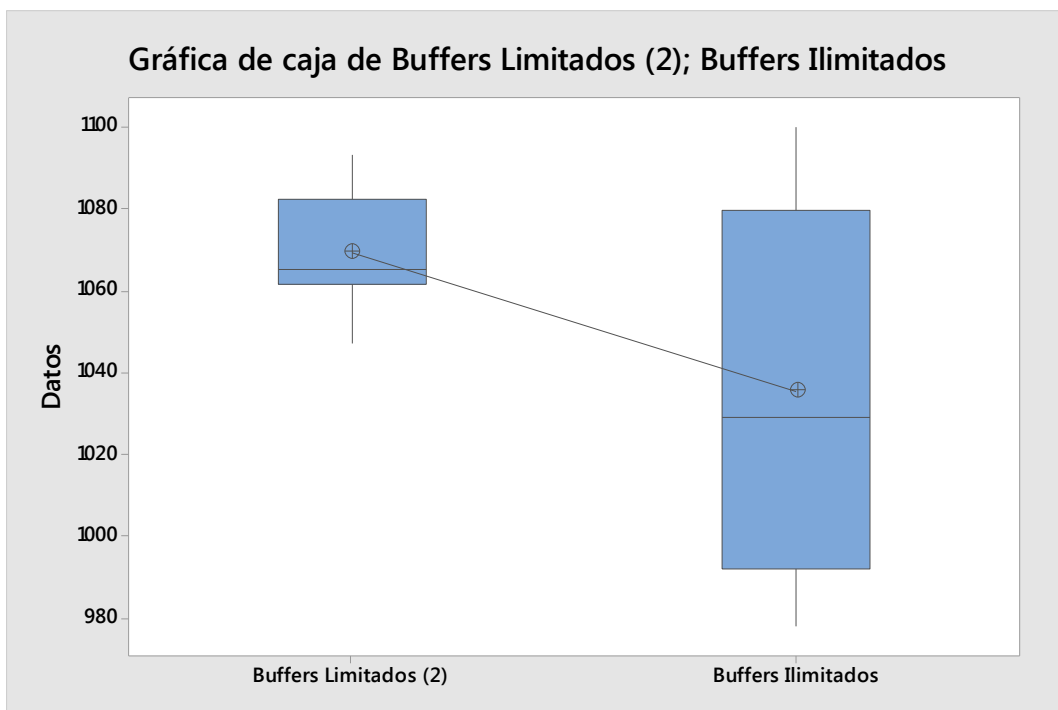
**Tabla 63.** Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia *i7j3k5-2*.

#	CP	%Imp	% $\xi$	%Cr	%Rev	MO [ut]	Med [ut]	SD [ut]
<b>Buffers limitados (13)</b>	100	0,1	0,3	0,6	0,4	1047	1069,3	14,213
<b>Buffers ilimitados (24)</b>	250	0,25	0,5	0,6	0,2	978	1035,5	46,7

En la instancia i7j3k5-2 (la cual se compara para la prueba de hipótesis de medias con las respuestas de la instancia i7j3k5 de Moratto y Pérez) se obtuvo que como Valor-P=0,000  $\leq$  0,05 existe evidencia suficiente para rechazar la hipótesis nula y así se asumen varianzas diferentes.

Para la prueba de hipótesis de medias fueron los siguientes, Valor T = 2,19, Valor P = 0,973, GL = 10. Como Valor-P=0,973  $>$  0,05 no existe evidencia suficiente para rechazar la hipótesis nula y así se comprueba tomando en cuenta también el diagrama de cajas que la metaheurística basada en el ICA para el HFS con buffers limitados tamaño dos obtiene secuencias con *makespan* algo superiores a los de la metaheurística basada en el ICA para el HFS con buffers de tamaño ilimitado. El incremento se puede deber tanto al desempeño del ICA como al impacto de la reducción en el tamaño del *buffer*.

**Figura 22.** Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con buffers limitados y buffers ilimitados para la instancia i7j3k5-2.



**Tabla 64.** Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia i7j3k5-3.

#	CP	%Imp	% $\xi$	%Cr	%Rev	MO [ut]	Med [ut]	SD [ut]
<b>Buffers limitados (23)</b>	100	0,25	0,5	0,6	0,2	1044	1062,8	16,247
<b>Buffers ilimitados (24)</b>	250	0,25	0,5	0,6	0,2	978	1035,5	46,7

Iniciando con la instancia i7j3k5-3 (la cual se compara para la prueba de hipótesis de medias con las soluciones de la instancia i7j3k5 de Moratto y Pérez) se obtuvo que como Valor-P=0,000  $\leq$  0,05 existe evidencia suficiente para rechazar la hipótesis nula y así se asumen varianzas diferentes.

Mientras tanto, para el caso de la prueba de hipótesis de medias, los resultados fueron Valor T = 1,75, Valor P = 0,946, GL = 11. Como Valor-P=0,946 > 0,05 no existe evidencia suficiente para rechazar la hipótesis nula y así se comprueba tomando en cuenta también el diagrama de cajas que la metaheurística basada en el *ICA* para el *HFS* con *buffers* limitados de tamaño tres obtiene secuencias con *makespan* un poco superiores a los de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado.

El incremento se puede deber tanto al desempeño del *ICA* como al impacto de la reducción en el tamaño del *buffer*. El diagrama de cajas de la presente instancia se presenta en la figura 23 más adelante.

**Tabla 65.** Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia i9j2k3-1.

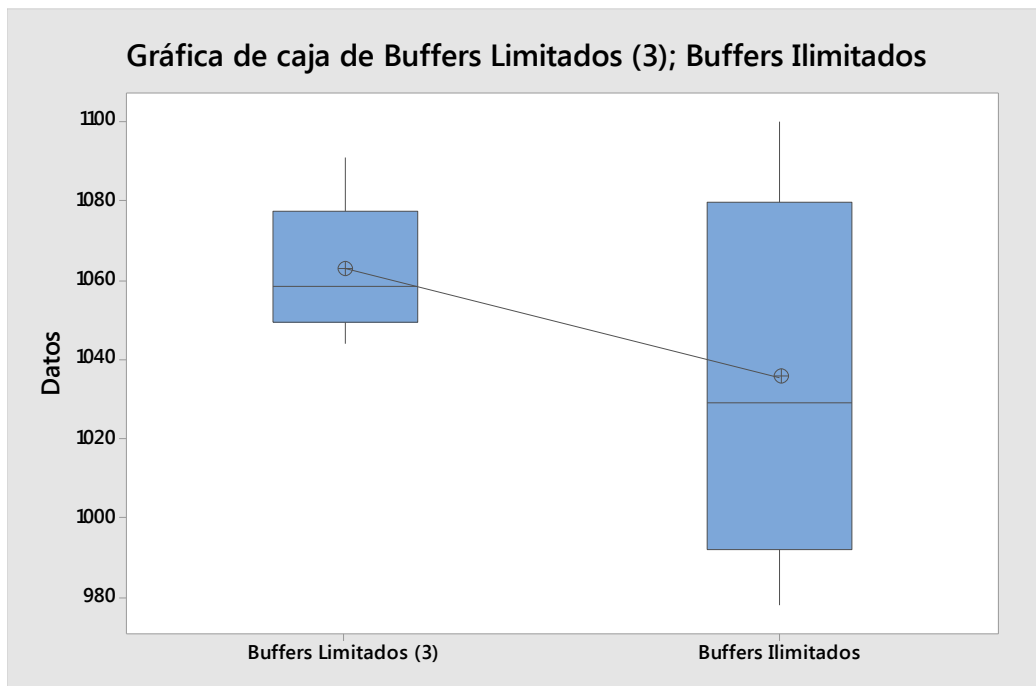
#	CP	%Imp	% $\xi$	%Cr	%Rev	MO [ut]	Med [ut]	SD [ut]
<b>Buffers limitados (17)</b>	100	0,1	0,5	0,4	0,2	1004	1026	11,284
<b>Buffers ilimitados (16)</b>	250	0,25	0,3	0,6	0,4	939	985,2	39,5693

Continuado con la instancia i9j2k3-1 se obtuvo que como Valor-P=0,000  $\leq$  0,05 existe evidencia suficiente para rechazar la hipótesis nula por lo tanto se asumen varianzas diferentes.

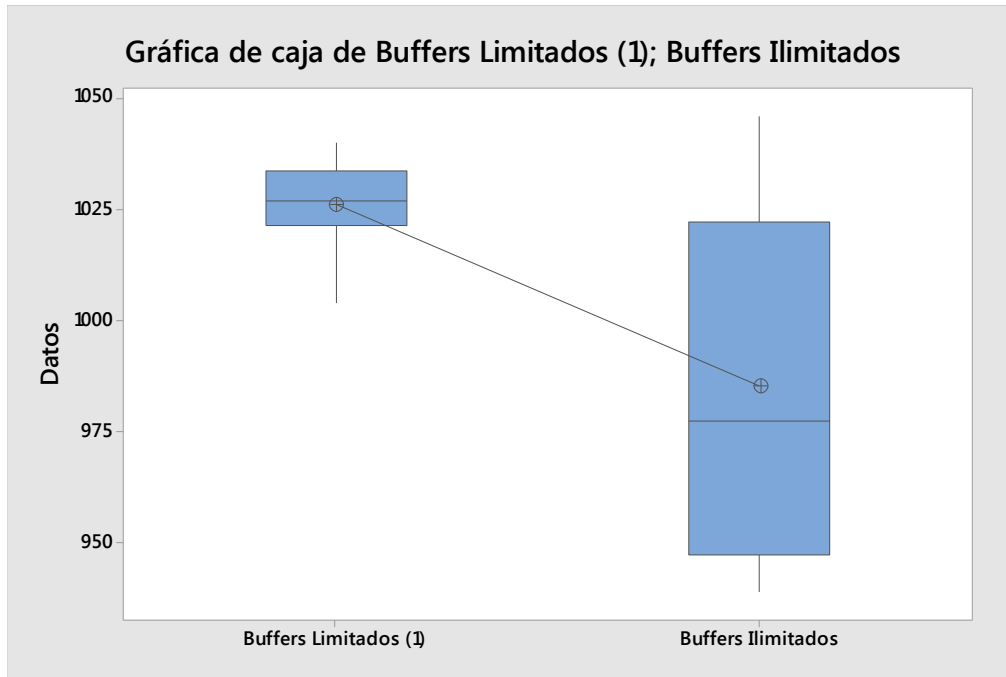
Para el caso de la prueba de hipótesis de medias de la instancia i9j2k3-1 (la cual se compara con la instancia i9j2k3 de Moratto y Pérez) los resultados fueron Valor  $T = 3,41$ , Valor  $P = 0,995$ ,  $GL = 10$ , por lo tanto, como  $\text{Valor-P} = 0,995 > 0,05$  no existe evidencia suficiente para rechazar la hipótesis nula, y así se demuestra, basándose también en el diagrama de cajas, que la metaheurística basada en el *ICA* para el *HFS* con *buffers* limitados tamaño uno obtiene secuencias con *makespan* generalmente superiores a los de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado.

El incremento se puede deber tanto al desempeño del *ICA* como al impacto de la reducción en el tamaño del *buffer*. El diagrama de cajas se presenta en la figura 24 más adelante.

**Figura 23.** Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia i7j3k5-3.



**Figura 24.** Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia i9j2k3-1.



**Tabla 66.** Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia i9j2k3-2.

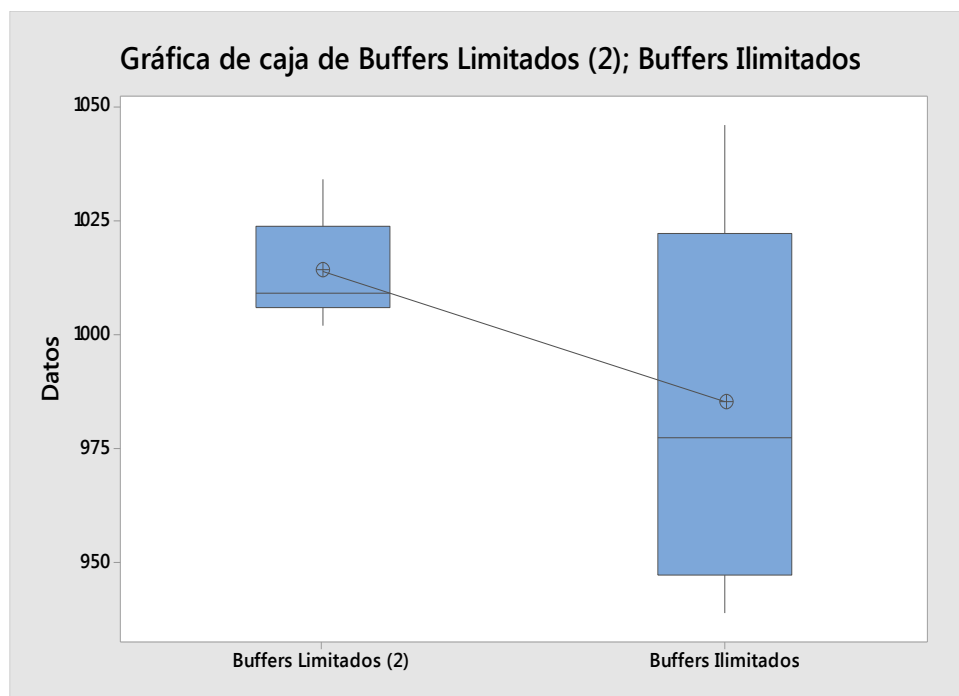
#	CP	%Imp	% $\xi$	%Cr	%Rev	MO [ut]	Med [ut]	SD [ut]
<b>Buffers limitados (25)</b>	100	0,1	0,5	0,4	0,4	1002	1014	11,136
<b>Buffers Ilimitados (16)</b>	250	0,25	0,3	0,6	0,4	939	985,2	39,5693

Para la instancia i9j2k3-2 (la cual se compara para la prueba de medias con las respuestas de la instancia i9j2k3 de Moratto y Pérez) en el caso de las varianzas se obtuvo que como Valor-P=0,000  $\leq$  0,05 existe evidencia suficiente para rechazar la hipótesis nula por lo tanto se asumen varianzas diferentes.

Mientras tanto, los resultados obtenidos para la prueba de hipótesis de medias para la instancia i9j2k3-2 fueron Valor T = 2,22, Valor P = 0,974, GL = 10, por lo tanto, como Valor-P=0,974  $>$  0,05 no existe evidencia suficiente para rechazar la

hipótesis nula y así se comprueba, observando también el diagrama de caja que la metaheurística basada en el *ICA* para el *HFS* con *buffers* limitados tamaño dos obtiene secuencias con *makespan* algo superiores y para algunos pocos casos iguales a los de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado. El incremento se puede deber tanto al desempeño del *ICA* como al impacto de la reducción en el tamaño del *buffer*.

**Figura 25.** Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia i9j2k3-2.



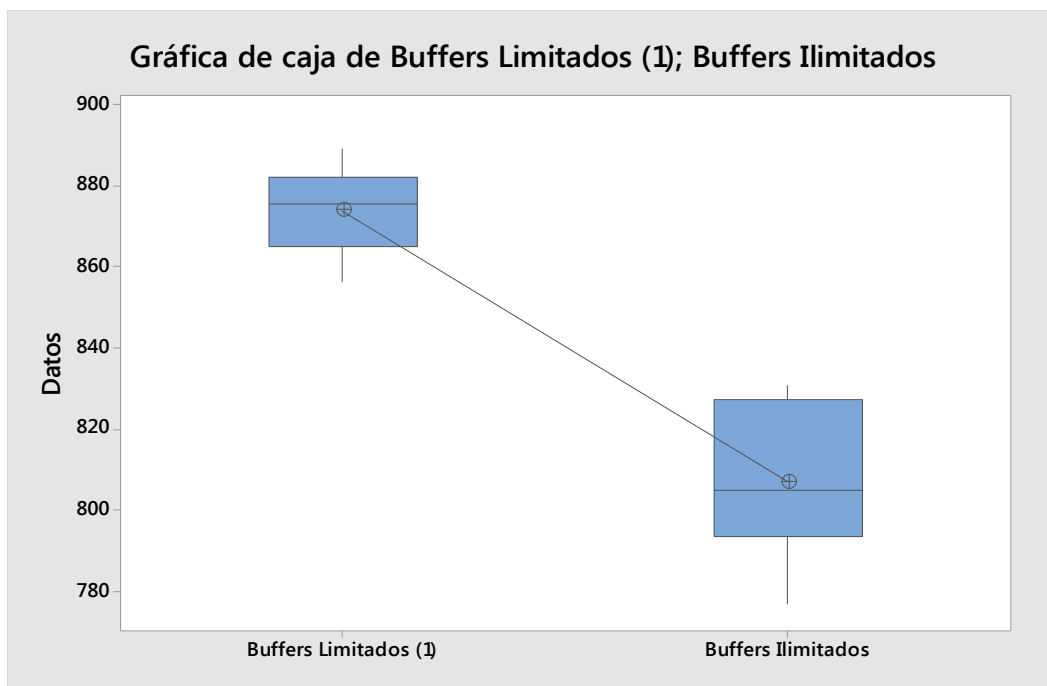
**Tabla 67.** Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia i9j3k3-1.

#	CP	%Imp	% $\xi$	%Cr	%Rev	MO [ut]	Med [ut]	SD [ut]
<b>Buffers limitados (29)</b>	100	0,1	0,5	0,6	0,4	856	873,8	10,80
<b>Buffers ilimitados (12)</b>	250	0,25	0,3	0,4	0,4	777	807,1	17,32

En el caso presentado para la instancia i9j3k3-1 para las varianzas se obtuvo que como Valor-P=0,144 > 0,05 no existe evidencia suficiente para rechazar la hipótesis nula por lo tanto se asumen varianzas iguales.

Para la prueba de hipótesis de medias de la instancia i9j3k3-1 (la cual se compara con las respuestas de la instancia i9j3k3 de Moratto y Pérez) los resultados fueron Valor T = 10,34, Valor P = 1,000, GL = 18, por lo tanto, como Valor-P = 1,000 > 0,05 no existe evidencia suficiente para rechazar la hipótesis nula y tomando en cuenta el diagrama de cajas, que la metaheurística basada en el ICA para el HFS con buffers limitados tamaño uno obtiene secuencias con *makespan* superiores a los de la metaheurística basada en el ICA para el HFS con buffers de tamaño ilimitado. El incremento se puede deber tanto al desempeño del ICA como al impacto de la reducción en el tamaño del *buffer*.

**Figura 26.** Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con buffers limitados y buffers ilimitados para la instancia i9j3k3-1.



**Tabla 68.** Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia i9j3k3-2.

#	CP	%Imp	% $\xi$	%Cr	%Rev	MO [ut]	Med [ut]	SD [ut]
<b>Buffers limitados (13)</b>	100	0,1	0,3	0,6	0,4	851	864	7,77
<b>Buffers ilimitados (12)</b>	250	0,25	0,3	0,4	0,4	777	807,1	17,32

Para la instancia i9j3k3-2 (la cual se compara para la prueba de hipótesis de medias con las soluciones de la instancia i9j3k3 de Moratto y Pérez) en el caso de las varianzas se obtuvo que como Valor-P=0,028  $\leq$  0,05 existe evidencia suficiente para rechazar la hipótesis nula por lo tanto se asumen varianzas diferentes.

Con la prueba de hipótesis de medias de la instancia i9j3k3-2 los resultados fueron Valor T = 9,48, Valor P = 1,000, GL = 12, por lo tanto, como Valor-P = 1,000 > 0,05 no existe evidencia suficiente para rechazar la hipótesis nula y tomando en cuenta el diagrama de cajas, se demuestra que la metaheurística basada en el *ICA* para el *HFS* con *buffers* limitados tamaño dos obtiene secuencias con *makespan* superiores a los de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado.

El incremento se puede deber tanto al desempeño del *ICA* como al impacto de la reducción en el tamaño del *buffer*. La figura 27 de más adelante presenta el diagrama de cajas de la presente instancia.

**Tabla 69.** Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia i9j3k3-3.

#	CP	%Imp	%x	%Cr	%Rev	MO [ut]	Med [ut]	SD [ut]
<b>Buffers limitados (5)</b>	100	0,1	0,3	0,6	0,2	832	841,4	7,79
<b>Buffers ilimitados (12)</b>	250	0,25	0,3	0,4	0,4	777	807,1	17,32

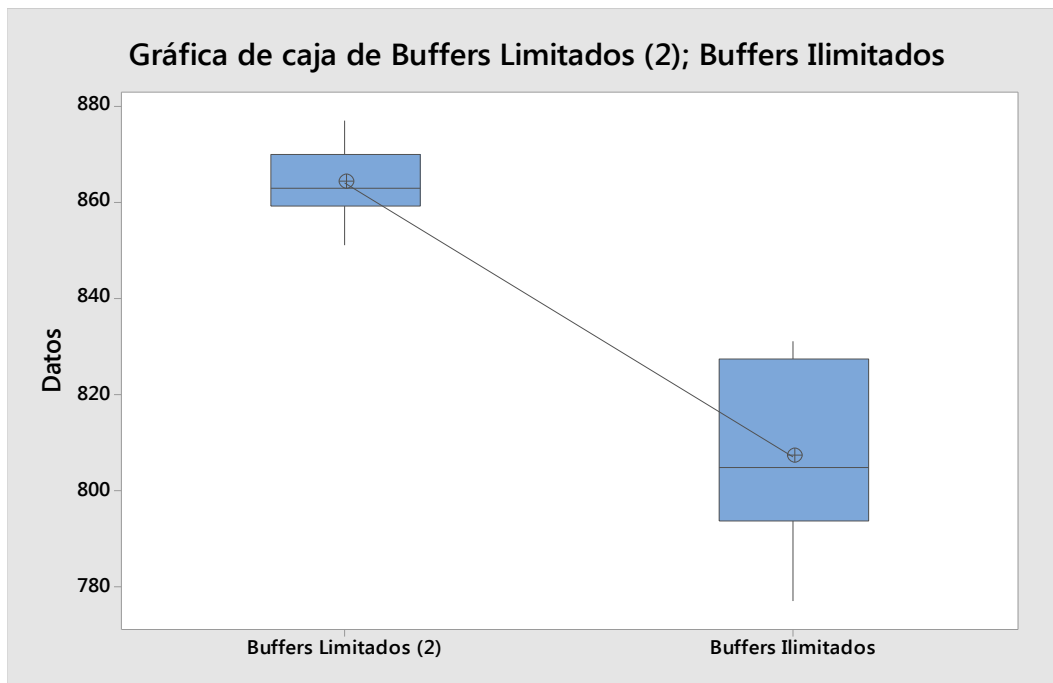
Continuando con el caso de la instancia i9j3k3-3 (que se compara en la prueba de medias con las respuestas de la instancia i9j3k3 de Moratto y Pérez) para las

varianzas se obtuvo que como Valor-P=0,028  $\leq$  0,05 existe evidencia suficiente para rechazar la hipótesis nula por lo tanto se asumen varianzas diferentes.

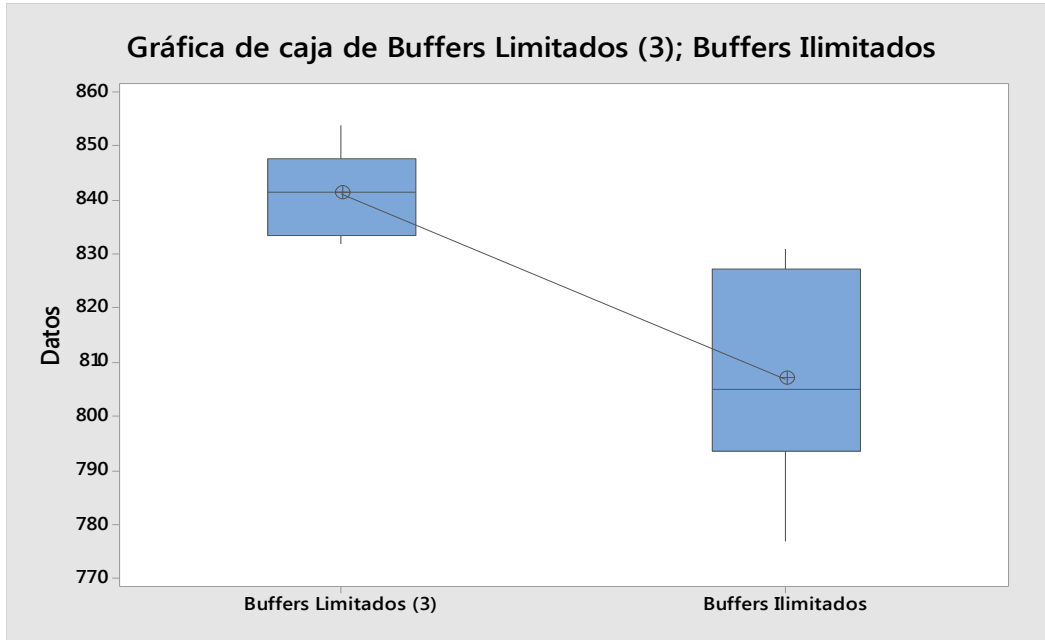
Para la prueba de hipótesis de medias los resultados fueron Valor T = 5,71, Valor P = 1,000, GL = 12, por lo tanto, como Valor-P = 1,000  $>$  0,05 no existe evidencia suficiente para rechazar la hipótesis nula y tomando en cuenta el diagrama de cajas, se demuestra que la metaheurística basada en el ICA para el HFS con buffers tamaño tres obtiene secuencias con *makespan* superiores a los de la metaheurística basada en el ICA para el HFS con buffers de tamaño ilimitado.

El incremento se puede deber tanto al desempeño del ICA como al impacto de la reducción en el tamaño del *buffer*. El diagrama de cajas de la instancia i9j3k3-3 se enseña más adelante mediante figura número 28.

**Figura 27.** Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con buffers limitados y buffers ilimitados para la instancia i9j3k3-2.



**Figura 28.** Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia i9j3k3-3.



**Tabla 70.** Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia i9j2k5-1.

#	CP	%Imp	% $\xi$	%Cr	%Rev	MO [ut]	Med [ut]	SD [ut]
<b>Buffers limitados (17)</b>	100	0,1	0,5	0,4	0,2	1468	1484,8	9,0774
<b>Buffers ilimitados (24)</b>	250	0,25	0,5	0,6	0,2	1460	1496,7	29,68

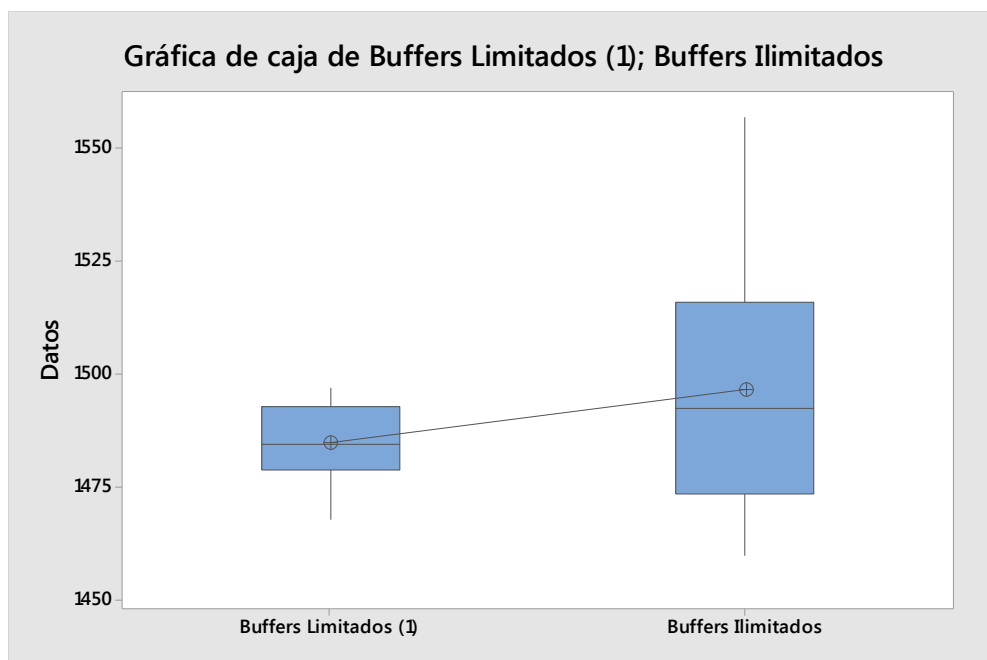
Comenzando con la instancia i9j2k5-1 (la cual se compara para la prueba de hipótesis de medias con las respuestas de la instancia i9j2k5 de Moratto y Pérez) se obtuvo que como Valor-P=0,025  $\leq$  0,05 existe evidencia suficiente para rechazar la hipótesis nula por lo tanto se asumen varianzas diferentes.

Para la prueba de hipótesis de medias de la presente instancia los resultados fueron Valor T = -1,21, Valor P = 0,127, GL = 10, por lo tanto, como Valor-P = 0,127 > 0,05 no existe evidencia suficiente para rechazar la hipótesis nula y tomando en cuenta el diagrama de cajas, se demuestra que la metaheurística

basada en el *ICA* para el *HFS* con *buffers* limitados tamaño uno obtiene secuencias con *makespan* prácticamente iguales a los de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado.

Por lo tanto, para la presente instancia el utilizar la metaheurística basada en el *ICA* con *buffers* limitados de tamaño uno o la metaheurística basada en el *ICA* con *buffers* ilimitados podría asegurar resultados de la misma calidad en el *makespan* de las secuencias.

**Figura 29.** Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia i9j2k5-1.



**Tabla 71.** Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia i9j2k5-2.

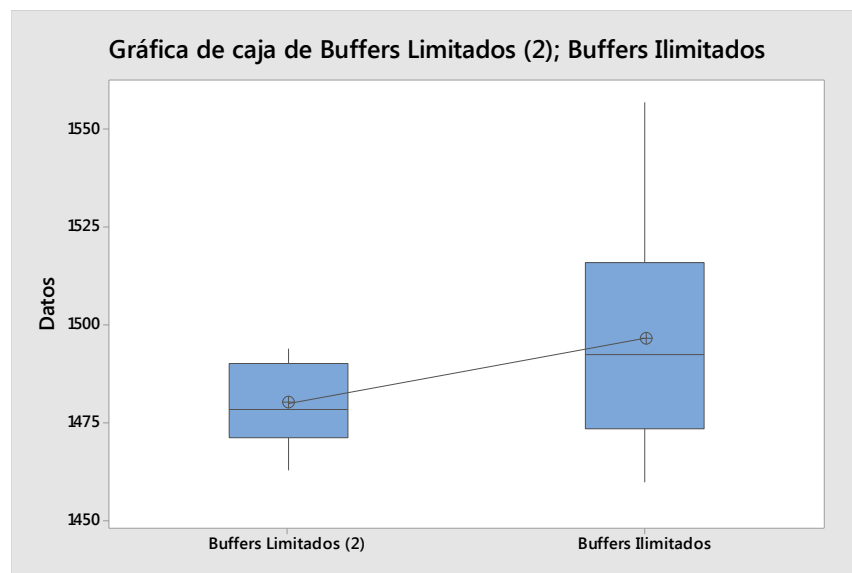
#	CP	%Imp	% $\xi$	%Cr	%Rev	MO [ut]	Med [ut]	SD [ut]
<b>Buffers limitados (29)</b>	100	0,1	0,5	0,6	0,4	1463	1480,1	10,376
<b>Buffers Ilimitados (24)</b>	250	0,25	0,5	0,6	0,2	1460	1496,7	29,68

Para la prueba de varianzas de la presente instancia se obtuvo que como Valor- $P=0,041 \leq 0,05$  existe evidencia suficiente para rechazar la hipótesis nula por lo tanto se asumen varianzas diferentes.

Con el caso de la prueba de hipótesis de medias los resultados de la instancia actual (la cual se compara con las soluciones de la instancia i9j2k5 de Moratto y Pérez) fueron los siguientes, Valor  $T = -1,67$ , Valor  $P = 0,062$ ,  $GL = 11$ . Como Valor- $P = 0,062 > 0,05$  no existe evidencia suficiente para rechazar la hipótesis nula y así se comprueba, observando también el diagrama de cajas y parecido como sucede en la instancia anterior que la metaheurística basada en el *ICA* para el *HFS* con *buffers* limitados tamaño dos obtiene secuencias con *makespan* de idéntica calidad a los de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado.

Así, para esta instancia el utilizar la metaheurística basada en el *ICA* con *buffers* limitados de tamaño dos o la metaheurística basada en el *ICA* con *buffers* ilimitados podría asegurar resultados iguales en cuestión del *makespan*.

**Figura 30.** Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia i9j2k5-2.



**Tabla 72.** Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia i9j3k5-1.

#	CP	%Imp	% ξ	%Cr	%Rev	MO [ut]	Med [ut]	SD [ut]
<b>Buffers limitados (3)</b>	100	0,25	0,3	0,4	0,2	1278	1294,7	11,166
<b>Buffers ilimitados (8)</b>	250	0,25	0,3	0,6	0,2	1175	1263,3	48,76

Para la instancia i9j3k5-1 los resultados de la prueba de hipótesis de varianzas dieron que como Valor-P=0,001 ≤ 0,05 existe evidencia suficiente para rechazar la hipótesis nula por lo tanto se asumen varianzas diferentes.

Para la prueba de hipótesis de medias (en la cual se compara la presente instancia con las respuestas de la instancia i9j3k5 de Moratto y Pérez) los resultados fueron Valor T = 1,99, Valor P = 0,961, GL = 9, por lo tanto, como Valor-P = 0,961 > 0,05 no existe evidencia suficiente para rechazar la hipótesis nula y tomando en cuenta el diagrama de cajas, se demuestra que la metaheurística basada en el *ICA* para el *HFS* con *buffers* limitados tamaño uno obtiene secuencias con *makespan* un poco superiores a los de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado.

El incremento se puede deber tanto al desempeño del *ICA* como al impacto de la reducción en el tamaño del *buffer*. El diagrama de cajas para la actual instancia se enseña más adelante en la figura 31.

**Tabla 73.** Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia i9j3k5-2.

#	CP	%Imp	% ξ	%Cr	%Rev	MO [ut]	Med [ut]	SD [ut]
<b>Buffers limitados (13)</b>	100	0,1	0,3	0,6	0,4	1260	1272,8	7,9415
<b>Buffers ilimitados (8)</b>	250	0,25	0,3	0,6	0,2	1175	1263,3	48,76

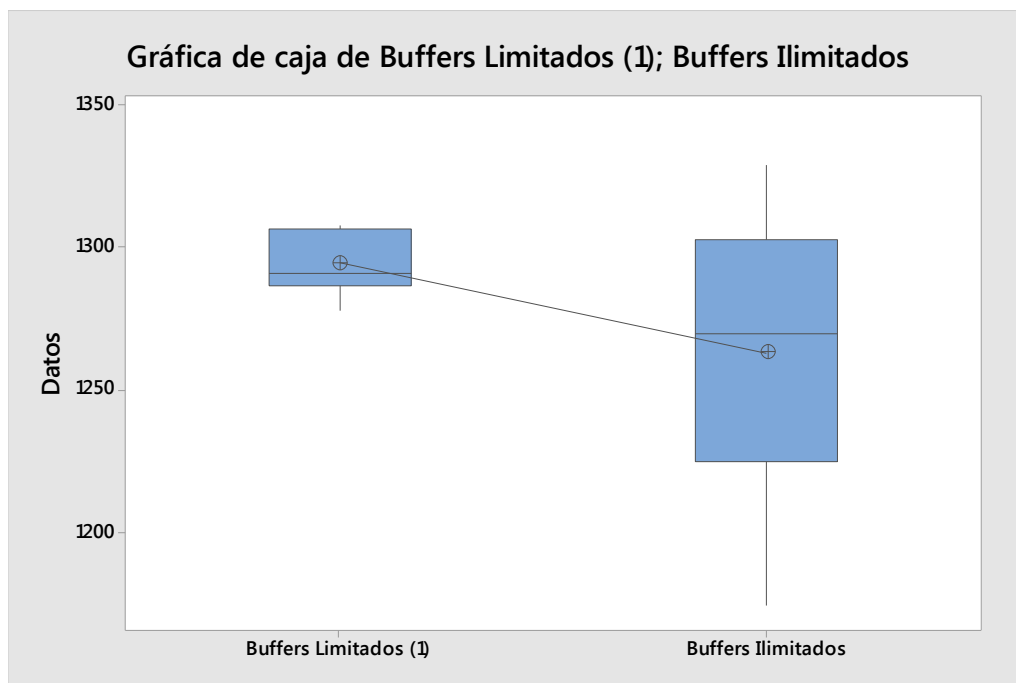
En la instancia i9j3k5-2 (la cual se compara para la prueba de hipótesis de medias con las respuestas de la instancia i9j3k5 de Moratto y Pérez) se obtuvo que como

Valor- $P=0,000 \leq 0,05$  existe evidencia suficiente para rechazar la hipótesis nula y así se asumen varianzas diferentes.

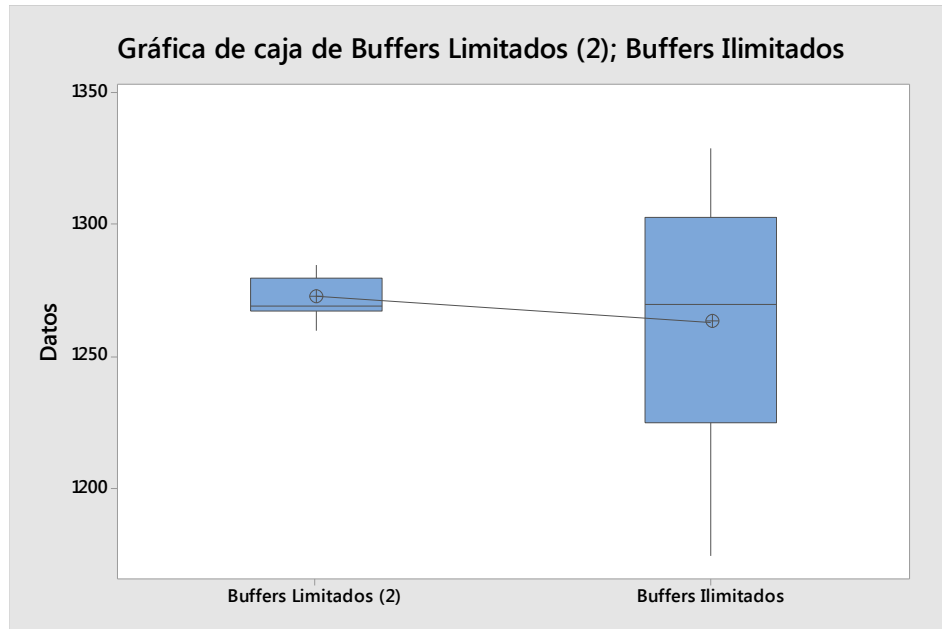
Para la prueba de hipótesis de medias fueron los siguientes, Valor  $T = 0,61$ , Valor  $P = 0,721$ ,  $GL = 10$ . Como Valor- $P=0,721 > 0,05$  no existe evidencia suficiente para rechazar la hipótesis nula y así se comprueba tomando en cuenta también el diagrama de cajas que la metaheurística basada en el *ICA* para el *HFS* con *buffers* limitados tamaño dos obtiene secuencias con *makespan* prácticamente iguales a los de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado.

Así, para la presente instancia el utilizar la metaheurística basada en el *ICA* con *buffers* limitados de tamaño dos o la metaheurística basada en el *ICA* con *buffers* ilimitados podría asegurar resultados de la misma calidad en el *makespan* de las secuencias. El diagrama de cajas se enseña más adelante en la figura 32.

**Figura 31.** Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia i9j3k5-1.



**Figura 32.** Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia i9j3k5-2.



**Tabla 74.** Resumen de los datos utilizados para la prueba de hipótesis de medias entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia i9j3k5-3.

#	CP	%Imp	% $\xi$	%Cr	%Rev	MO [ut]	Med [ut]	SD [ut]
<b>Buffers limitados (13)</b>	100	0,1	0,3	0,6	0,4	1259	1267,7	5,9264
<b>Buffers ilimitados (8)</b>	250	0,25	0,3	0,6	0,2	1175	1263,3	48,76

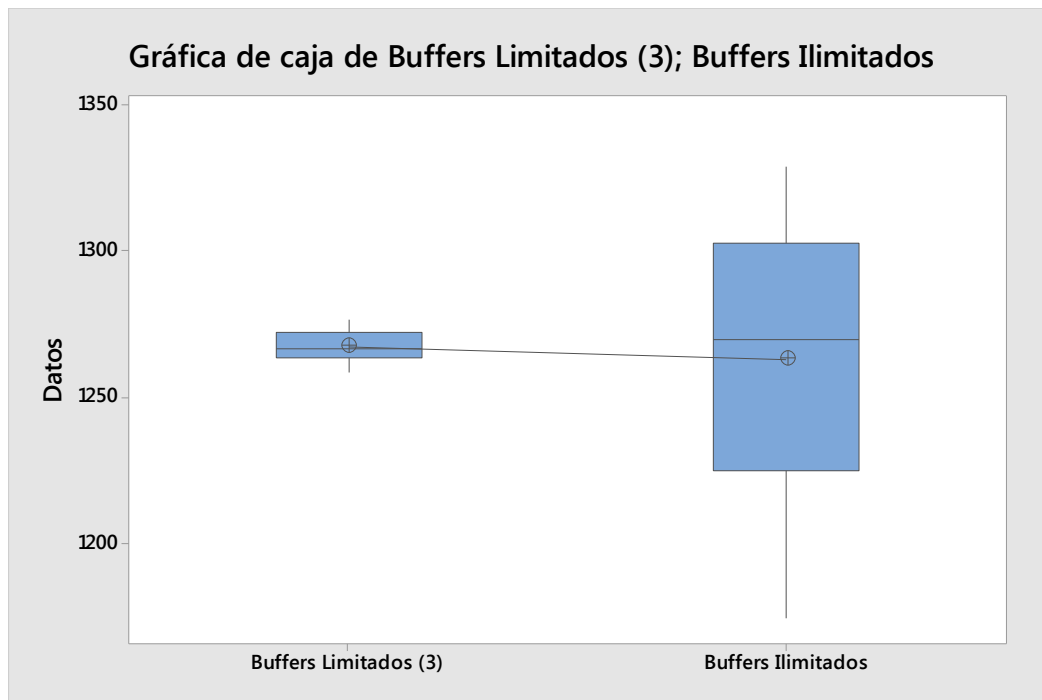
Finalizando con la instancia i9j3k5-3 (la cual se compara para la prueba de hipótesis de medias con las soluciones de la instancia i9j3k5 de Moratto y Pérez) se obtuvo que como Valor-P=0,000  $\leq$  0,05 existe evidencia suficiente para rechazar la hipótesis nula y así se asumen varianzas diferentes.

Mientras tanto, para el caso de la prueba de hipótesis de medias, los resultados fueron Valor T = 0,28, Valor P = 0,608, GL = 9. Como Valor-P=0,608 > 0,05 no existe evidencia suficiente para rechazar la hipótesis nula y así se comprueba tomando en cuenta también el diagrama de cajas y como era de suponerse con

base en la instancia anterior que la metaheurística basada en el *ICA* para el *HFS* con *buffers* limitados tamaño tres obtiene secuencias con *makespan* idénticas a los de la metaheurística basada en el *ICA* para el *HFS* con *buffers* de tamaño ilimitado.

Por último y para la presente instancia el uso de la metaheurística basada en el *ICA* con *buffers* limitados de tamaño tres o la metaheurística basada en el *ICA* con *buffers* ilimitados podría asegurar resultados de idéntica calidad en el *makespan* de las secuencias

**Figura 33.** Diagrama de cajas para la prueba de hipótesis de medias entre los resultados entre los resultados con *buffers* limitados y *buffers* ilimitados para la instancia i9j3k5-3.



**Anexo F.** Diseño experimental para determinar los factores más influyentes en el desempeño de la metaheurística basada en el *ICA* para el *HFS* con *buffers* limitados.

A continuación, se busca determinar y analizar cuáles son los factores más influyentes en la variable respuesta *makespan* que entrega la metaheurística basada en el *ICA*. Para lograr eso, se elige un diseño factorial fraccionado  $2^{k-1}$ , el cual se define como una fracción a la mitad del diseño factorial entero<sup>130</sup>. Como ya se mencionó en la sección de validación de la metaheurística, para el presente diseño se consideran 5 factores (*k*) y dos niveles para cada uno de los factores.

Para realizar el diseño se tienen en cuenta las respuestas que se presentan en el anexo G que en parte son las mismas utilizadas para la validación del *ICA* para *buffers* de tamaño limitado. Dichas soluciones fueron obtenidas con base en los tiempos de alistamiento y procesamiento de tareas presentados en la primera parte del anexo B.

Al igual que para la validación, las respuestas se obtienen con base a un código de programación realizado en *Matlab* R2012a (ver anexo C) y que a su vez se ejecutan en una computadora con procesador *Intel Core i5-4570*, 3.2 GHz con una memoria *RAM* de 8 *Gigabytes* y un sistema operativo de 64 *bits*. Finalmente, el diseño experimental se analiza en el *software* estadístico *Minitab 17*.

El rendimiento de la metaheurística basada en el algoritmo competitivo imperialista depende de los siguientes parámetros:

- Cantidad de países
- Porcentaje de países imperialistas
- Porcentaje de países que comparten información entre sí (también conocido como porcentaje de cruce de colonias).
- Peso de las colonias en el cálculo del poder total de imperio (%  $\xi$ ).

---

<sup>130</sup> MONTGOMERY, Douglas C. Diseño y análisis de experimentos. Limusa Wiley. 2da edición. 2004.

- Porcentaje de revolución de los países colonia.

Los niveles de cada uno de los parámetros presentados anteriormente son los mismos tenidos en cuenta para la validación del algoritmo y se presentan de nuevo en la siguiente tabla:

**Tabla 75.** Factores y niveles del ICA tenidos en cuenta para el diseño experimental.

Factor	Nivel bajo	Nivel alto
Número de países	100	250
% Imperialistas	0,10	0,25
% $\xi$	0,30	0,50
% Cruce	0,40	0,60
% Revolución	0,20	0,40

En la tabla presentada a continuación se listan los detalles del diseño factorial  $2^{5-1} = 2^4$ . La tabla enseña 16 de las 32 combinaciones de niveles de tratamientos posibles para un diseño factorial completo  $2^k$ . Cada una de las combinaciones se ejecutó 5 veces para todas las 20 instancias.

**Tabla 76.** Combinaciones de niveles de tratamiento del diseño factorial  $2^{k-1}$ .

#	A Número de Países	B % Imperialistas	C % $\xi$	D % Cruce	E % Revolución	Combinación de niveles de tratamientos
1	-	-	-	-	+	e
2	+	-	-	-	-	a
3	-	+	-	-	-	b
4	+	+	-	-	+	abe
5	-	-	+	-	-	c
6	+	-	+	-	+	ace
7	-	+	+	-	+	bce
8	+	+	+	-	-	abc
9	-	-	-	+	-	d
10	+	-	-	+	+	ade
11	-	+	-	+	+	bde
12	+	+	-	+	-	abd
13	-	-	+	+	+	cde
14	+	-	+	+	-	acd
15	-	+	+	+	-	bcd
16	+	+	+	+	+	abcde

Fuente: Minitab 12.

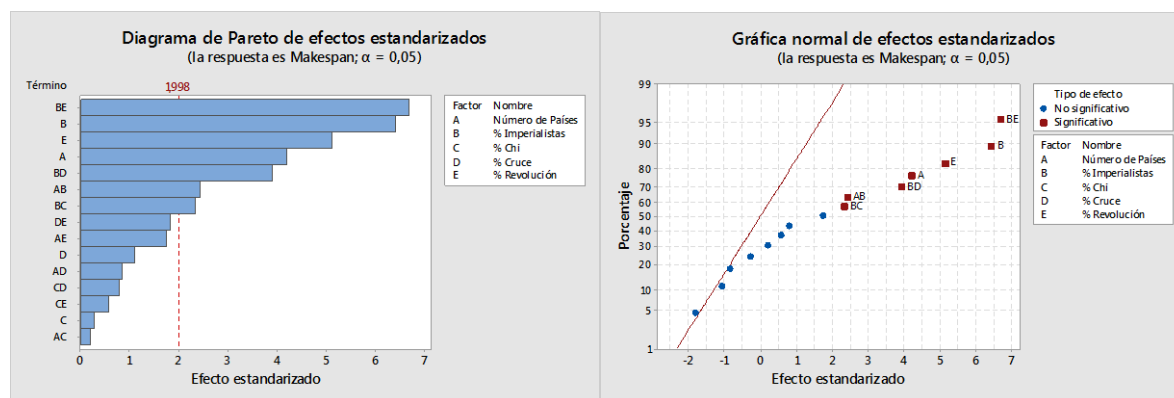
Ahora se presentan los análisis de varianzas (ANOVA) de las 20 instancias planteadas, de ellas se determina cuáles son los factores más incidentes en el rendimiento de la metaheurística basada en el *ICA* para *buffers* limitados mostrando también para cada una de ellas el diagrama de Pareto y la gráfica normal de efectos estandarizados.

- **Análisis de la varianza para la instancia i7j2k3-1.**

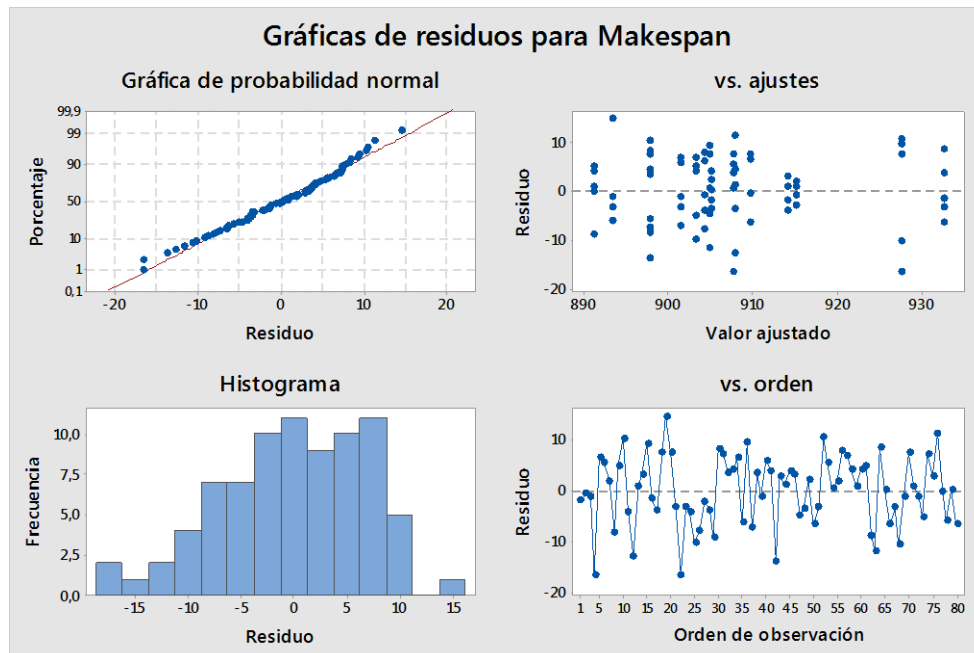
**Tabla 77.** ANOVA para la instancia i7j2k3-1.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
<b>Modelo</b>	15	9364	624,27	11,04	0
<b>Lineal</b>	5	4882,1	976,41	17,27	0
<b>Número de Países</b>	1	1001,1	1001,11	17,71	<b>0</b>
<b>% Imperialistas</b>	1	2322	2322,01	41,07	<b>0</b>
<b>% Chi</b>	1	4,5	4,51	0,08	0,778
<b>% Cruce</b>	1	66,6	66,61	1,18	0,282
<b>% Revolución</b>	1	1487,8	1487,81	26,32	<b>0</b>
<b>Interacciones de 2 términos</b>	10	4481,9	448,19	7,93	0
<b>Número de Países*% Imperialistas</b>	1	332,1	332,11	5,87	<b>0,018</b>
<b>Número de Países*% Chi</b>	1	2,1	2,11	0,04	0,847
<b>Número de Países*% Cruce</b>	1	40,6	40,61	0,72	0,4
<b>Número de Países*% Revolución</b>	1	171,1	171,11	3,03	0,087
<b>% Imperialistas*% Chi</b>	1	308,1	308,11	5,45	<b>0,023</b>
<b>% Imperialistas*% Cruce</b>	1	864,6	864,61	15,29	<b>0</b>
<b>% Imperialistas*% Revolución</b>	1	2520	2520,01	44,57	<b>0</b>
<b>% Chi*% Cruce</b>	1	35,1	35,11	0,62	0,434
<b>% Chi*% Revolución</b>	1	19	19,01	0,34	0,564
<b>% Cruce*% Revolución</b>	1	189,1	189,11	3,34	0,072
<b>Error</b>	64	3618,4	56,54		
<b>Total</b>	79	12982,4			

**Gráfica 15.** Diagrama de Pareto y gráfica normal de efectos estandarizados para el *makespan* de la instancia i7j2k3-1.



**Gráfica 16.** Diagrama de residuos para el *makespan* de la instancia i7j2k3-1.



Con base en los datos presentados en la tabla 77 del ANOVA y en la gráfica 15 del diagrama de Pareto y la gráfica normal de efectos estandarizados se concluye que para la instancia i7j2k3-1 los factores número de países, % de imperialistas, % de revolución y la interacción entre % de imperialistas y todos los demás factores influyen en la variable respuesta *makespan* ya que el valor-p de éstas es menor que 0,05. A su vez, el factor menos influyente es la interacción entre número de países y %  $\xi$  pues tiene el valor-p más alto de todos los factores con 0,847.

En su orden, el factor más influyente es la interacción entre % de imperialistas y % de revolución, en segundo lugar, el % de imperialistas, en tercer lugar, el % de revolución, en cuarto lugar, el número de países, en la quinta posición la interacción entre % de imperialistas y % de cruce luego la interacción entre número de países y % de imperialistas y finalmente la interacción entre % de imperialistas y %  $\xi$ .

Finalmente, con base en la gráfica 16 se confirman los supuestos de normalidad e independencia de los datos. Esto significa que los mismos provienen de una

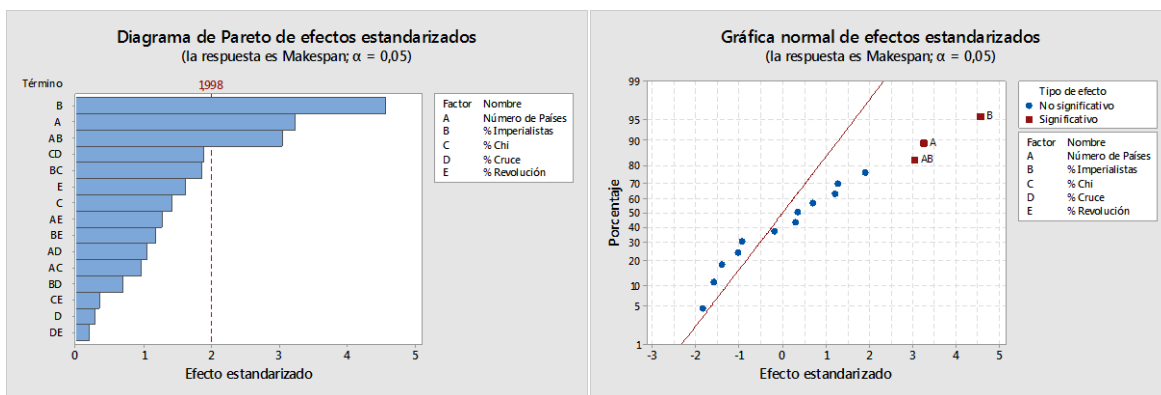
población con distribución normal y que son independientes entre uno y otro ya no que se observa ninguna tendencia de los datos a simple vista.

- **Análisis de la varianza para la instancia i7j2k3-2.**

**Tabla 78.** ANOVA para la instancia i7j2k3-2.

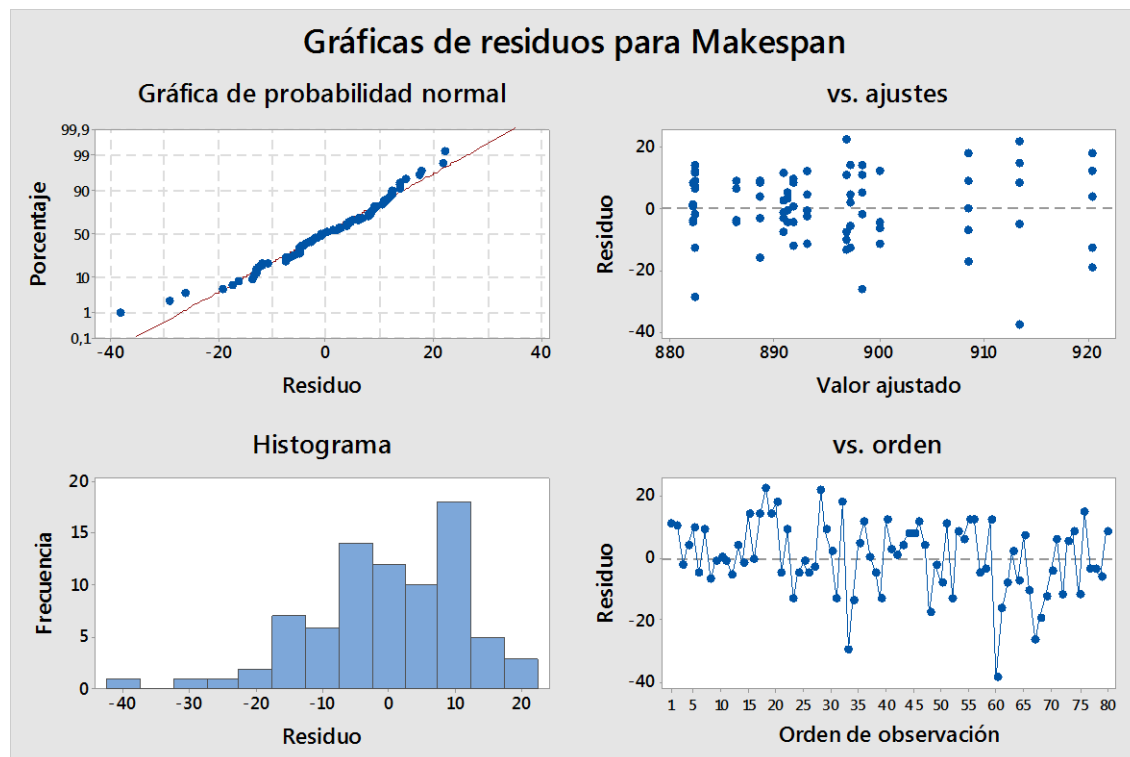
Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
<b>Modelo</b>	15	9384,5	625,63	3,87	0
<b>Lineal</b>	5	5825,3	1165,05	7,2	0
<b>Número de Países</b>	1	1702	1702,01	10,52	<b>0,002</b>
<b>% Imperialistas</b>	1	3367	3367,01	20,81	<b>0</b>
<b>% Chi</b>	1	324	324,01	2	0,162
<b>% Cruce</b>	1	13,6	13,61	0,08	0,773
<b>% Revolución</b>	1	418,6	418,61	2,59	0,113
<b>Interacciones de 2 términos</b>	10	3559,2	355,92	2,2	0,029
<b>Número de Países*% Imperialistas</b>	1	1505,1	1505,11	9,3	<b>0,003</b>
<b>Número de Países*% Chi</b>	1	148,5	148,51	0,92	0,342
<b>Número de Países*% Cruce</b>	1	177	177,01	1,09	0,299
<b>Número de Países*% Revolución</b>	1	262,8	262,81	1,62	0,207
<b>% Imperialistas*% Chi</b>	1	556,5	556,51	3,44	0,068
<b>% Imperialistas*% Cruce</b>	1	78	78,01	0,48	0,49
<b>% Imperialistas*% Revolución</b>	1	227,8	227,81	1,41	0,24
<b>% Chi*% Cruce</b>	1	577,8	577,81	3,57	0,063
<b>% Chi*% Revolución</b>	1	19	19,01	0,12	0,733
<b>% Cruce*% Revolución</b>	1	6,6	6,61	0,04	0,84
<b>Error</b>	64	10354,4	161,79		
<b>Total</b>	79	19738,9			

**Gráfica 17.** Diagrama de Pareto y gráfica normal de efectos estandarizados para el *makespan* de la instancia i7j2k3-2.



Para la instancia i7j2k3-2 observando los datos presentados en la tabla 78 del ANOVA y en la gráfica 17 del diagrama de Pareto y la gráfica normal de efectos estandarizados se concluye que los factores número de países, % de imperialistas y la interacción entre número de países y % de imperialistas influyen en la variable respuesta *makespan*. Mientras tanto, el factor menos influyente es % de cruce de información entre las colonias.

**Gráfica 18.** Diagrama de residuos para el *makespan* de la instancia i7j2k3-2.



El factor más influyente entre los ya mencionados es el % de imperialistas, seguido por el número de países y tercer lugar la interacción entre los dos primeros factores. Por último, con base en la gráfica 18 se confirman los supuestos de normalidad e independencia de los datos.

- **Análisis de la varianza para la instancia i7j3k3-1.**

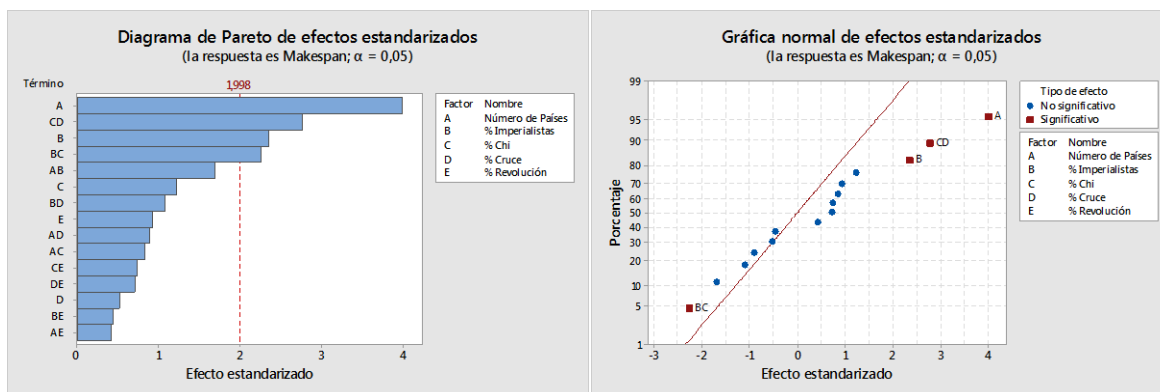
Continuando con la instancia i7j3k3-1, cuya ANOVA se presenta en la tabla 79, se concluye con base en la tabla misma y en la gráfica 19 del diagrama de Pareto y la

gráfica normal de efectos estandarizados que los factores número de países, % de imperialistas, la interacción entre % de imperialistas y %  $\xi$  y la interacción entre %  $\xi$  y % de cruce influyen en la variable respuesta *makespan*. Por su parte, el factor menos influyente es la interacción entre el número de países y el % de revolución.

**Tabla 79.** ANOVA para la instancia i7j3k3-1.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
<b>Modelo</b>	15	2198,19	146,546	2,92	0,001
<b>Lineal</b>	5	1208,86	241,772	4,82	0,001
<b>Número de Países</b>	1	800,11	800,112	15,94	<b>0</b>
<b>% Imperialistas</b>	1	277,51	277,512	5,53	<b>0,022</b>
<b>% Chi</b>	1	74,11	74,112	1,48	0,229
<b>% Cruce</b>	1	13,61	13,613	0,27	0,604
<b>% Revolución</b>	1	43,51	43,512	0,87	0,355
<b>Interacciones de 2 términos</b>	10	989,32	98,932	1,97	0,051
<b>Número de Países*% Imperialistas</b>	1	143,11	143,113	2,85	0,096
<b>Número de Países*% Chi</b>	1	35,11	35,113	0,7	0,406
<b>Número de Países*% Cruce</b>	1	40,61	40,612	0,81	0,372
<b>Número de Países*% Revolución</b>	1	9,11	9,113	0,18	0,672
<b>% Imperialistas*% Chi</b>	1	255,61	255,613	5,09	<b>0,027</b>
<b>% Imperialistas*% Cruce</b>	1	59,51	59,512	1,19	0,28
<b>% Imperialistas*% Revolución</b>	1	10,51	10,513	0,21	0,649
<b>% Chi*% Cruce</b>	1	382,81	382,813	7,62	<b>0,008</b>
<b>% Chi*% Revolución</b>	1	27,61	27,612	0,55	0,461
<b>% Cruce*% Revolución</b>	1	25,31	25,313	0,5	0,48
<b>Error</b>	64	3213,2	50,206		
<b>Total</b>	79	5411,39			

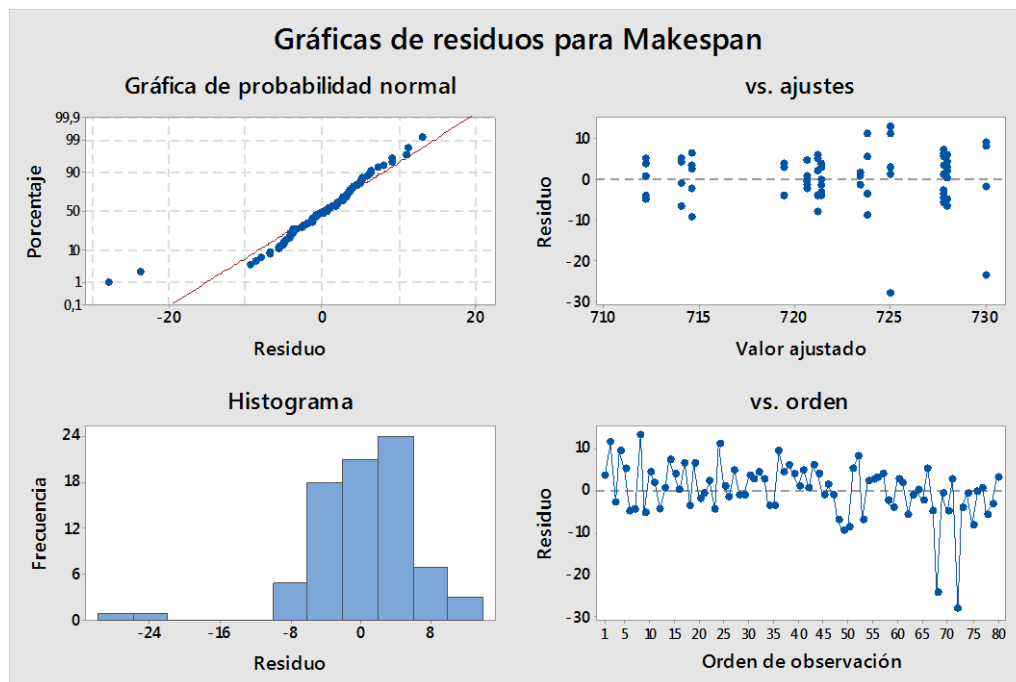
**Gráfica 19.** Diagrama de Pareto y gráfica normal de efector estandarizados para el *makespan* de la instancia i7j3k3-1.



Encabeza los factores más influyentes el número de países, seguido por la interacción entre %  $\xi$  y % de cruce, luego el % de imperialistas y, en cuarto lugar, la interacción entre % de imperialistas y %  $\xi$ .

También, con base en la gráfica 20 presentada a continuación se confirman los supuestos de normalidad e independencia de los datos.

**Gráfica 20.** Diagrama de residuos para el *makespan* de la instancia i7j3k3-1.

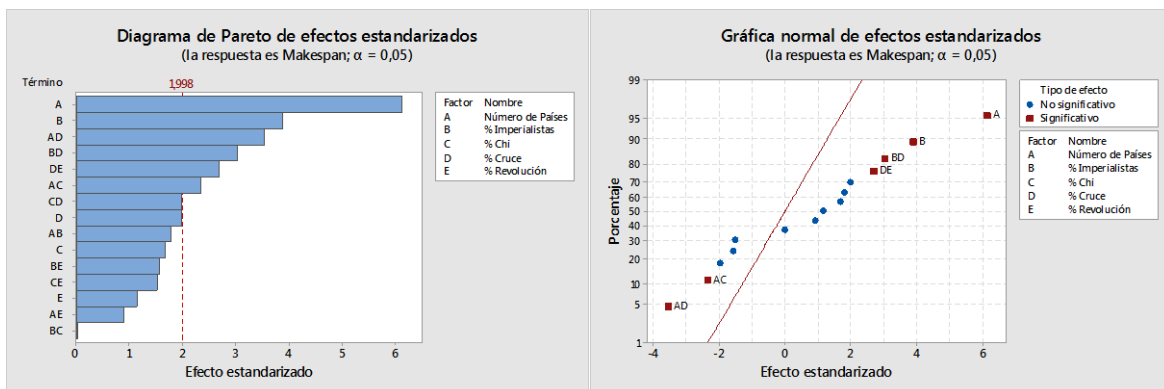


- Análisis de la varianza para la instancia i7j3k3-2.

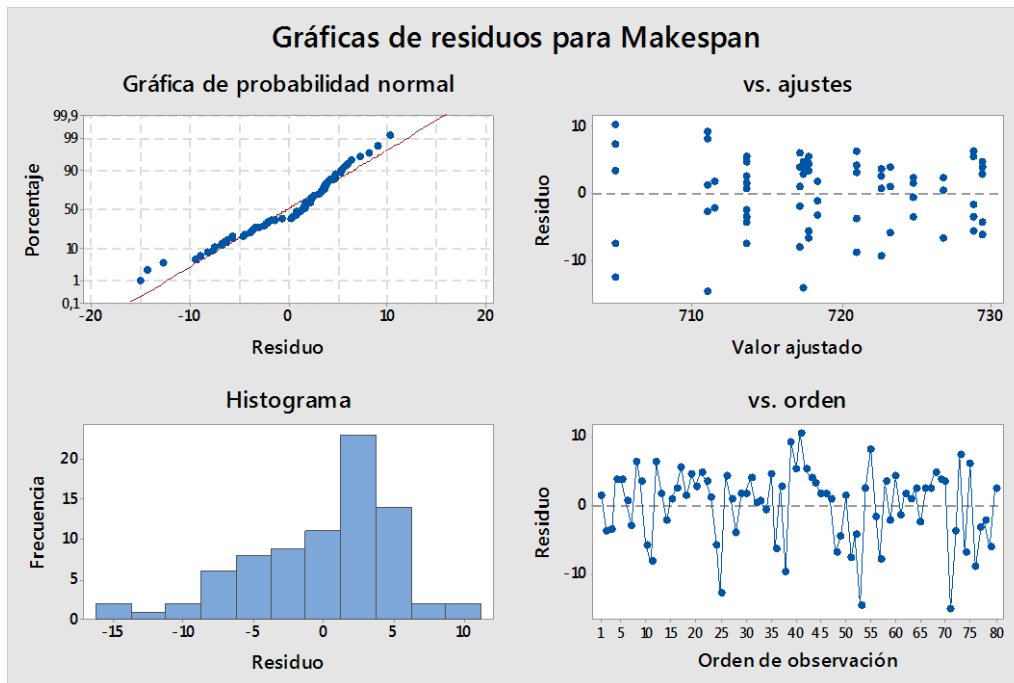
Tabla 80. ANOVA para la instancia i7j3k3-2.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Modelo	15	3611,49	240,77	7,21	0
Lineal	5	2031,86	406,37	12,16	0
Número de Países	1	1256,11	1256,11	37,6	<b>0</b>
% Imperialistas	1	505,01	505,01	15,12	<b>0</b>
% Chi	1	94,61	94,61	2,83	0,097
% Cruce	1	132,61	132,61	3,97	0,051
% Revolución	1	43,51	43,51	1,3	0,258
Interacciones de 2 términos	10	1579,62	157,96	4,73	0
Número de Países*% Imperialistas	1	108,11	108,11	3,24	0,077
Número de Países*% Chi	1	183,01	183,01	5,48	<b>0,022</b>
Número de Países*% Cruce	1	418,61	418,61	12,53	<b>0,001</b>
Número de Países*% Revolución	1	27,61	27,61	0,83	0,367
% Imperialistas*% Chi	1	0,01	0,01	0	0,985
% Imperialistas*% Cruce	1	308,11	308,11	9,22	<b>0,003</b>
% Imperialistas*% Revolución	1	82,01	82,01	2,46	0,122
% Chi*% Cruce	1	132,61	132,61	3,97	0,051
% Chi*% Revolución	1	78,01	78,01	2,34	0,131
% Cruce*% Revolución	1	241,51	241,51	7,23	<b>0,009</b>
Error	64	2138	33,41		
Total	79	5749,49			

Gráfica 21. Diagrama de Pareto y gráfica normal de efectores estandarizados para el *makespan* de la instancia i7j3k3-2.



**Gráfica 22.** Diagrama de residuos para el *makespan* de la instancia i7j3k3-2.



Analizando la tabla 80 donde se presenta la ANOVA y la gráfica 21 del diagrama de Pareto y la gráfica normal de efectos estandarizados de la instancia i7j3k3-2 se concluye que los factores más influyentes en la variable respuesta *makespan* son el número de países, el % de imperialistas, la interacción entre el número de países y el %  $\xi$ , la interacción entre el número de países y % de cruce, la interacción entre el % de imperialistas y el % de cruce y por último el % de cruce y el % de revolución. Por el contrario, el factor menos influyente es la interacción entre el % de imperialistas y el %  $\xi$ .

También se determina que el factor más influyente es el número de países, luego, el % de imperialistas, seguido por la interacción entre el número de países y % de cruce, tras estas, aparece la interacción entre el % de imperialistas y el % de cruce, seguidamente la interacción entre % de cruce y el % de revolución y finalmente, la interacción entre el número de países y el %  $\xi$ . A su vez, con base en la gráfica 22 se confirman los supuestos de normalidad e independencia de los datos.

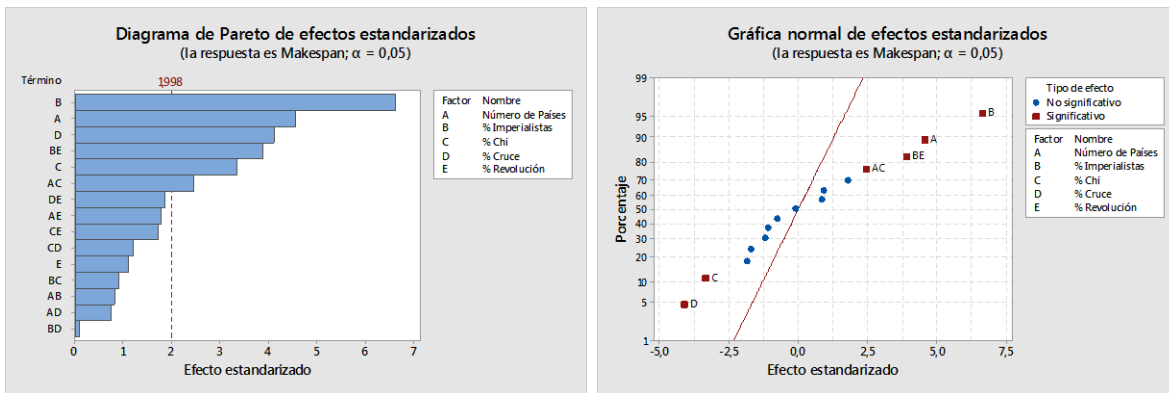
- Análisis de la varianza para la instancia i7j3k3-3.

Tabla 81. ANOVA para la instancia i7j3k3-3.

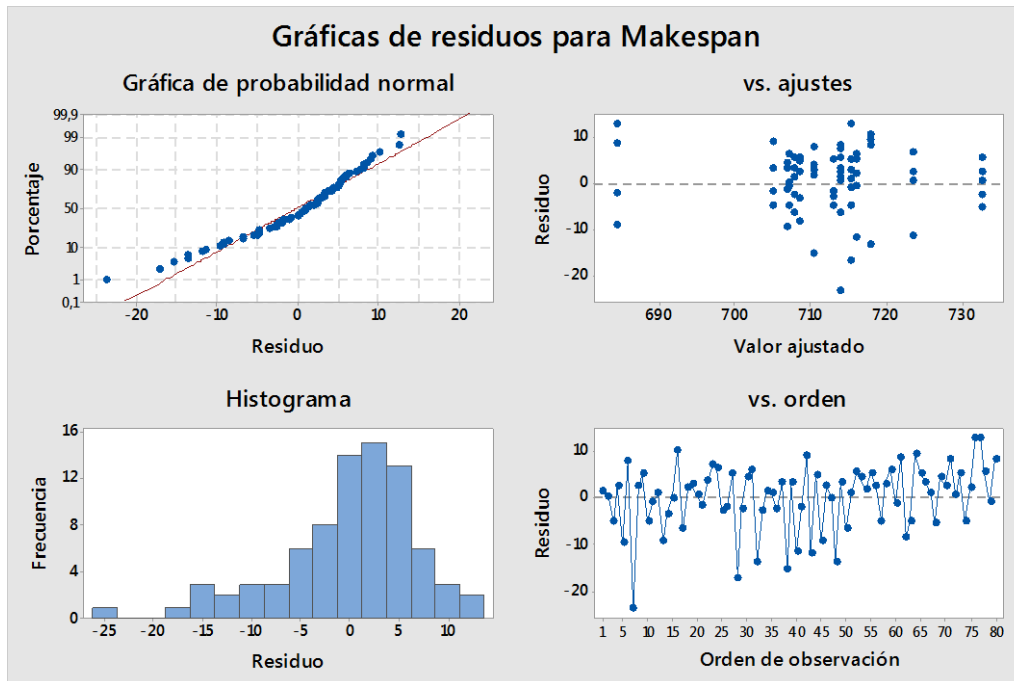
Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Modelo	15	7655,1	510,34	8,55	0
Lineal	5	5607,4	1121,47	18,79	0
Número de Países	1	1240,3	1240,31	20,78	0
% Imperialistas	1	2610,6	2610,61	43,74	0
% Chi	1	667	667,01	11,18	0,001
% Cruce	1	1015,3	1015,31	17,01	0
% Revolución	1	74,1	74,11	1,24	0,269
Interacciones de 2 términos	10	2047,7	204,77	3,43	0,001
Número de Países*% Imperialistas	1	40,6	40,61	0,68	0,412
Número de Países*% Chi	1	357	357,01	5,98	0,017
Número de Países*% Cruce	1	35,1	35,11	0,59	0,446
Número de Países*% Revolución	1	189,1	189,11	3,17	0,08
% Imperialistas*% Chi	1	49,6	49,61	0,83	0,365
% Imperialistas*% Cruce	1	0,6	0,61	0,01	0,92
% Imperialistas*% Revolución	1	904,5	904,51	15,16	0
% Chi*% Cruce	1	86,1	86,11	1,44	0,234
% Chi*% Revolución	1	177	177,01	2,97	0,09
% Cruce*% Revolución	1	208	208,01	3,49	0,066
Error	64	3819,6	59,68		
Total	79	11474,7			

Para el caso de la instancia i7j3k3-3 se concluye con base en la tabla 81 de ANOVA y la gráfica 23 del diagrama de Pareto y la gráfica normal de efectos estandarizados, presentada a continuación, que influyen la variable respuesta *makespan* en orden descendente de influjo los factores % de imperialistas, número de países, % de cruce, la interacción entre % de imperialistas y % de revolución, el %  $\xi$ , y la interacción entre número de países y %  $\xi$ . El factor menos influyente es la interacción entre % de imperialistas y % de cruce; también se confirma con base en la gráfica 24 los supuestos de normalidad e independencia de los datos.

**Gráfica 23.** Diagrama de Pareto y gráfica normal de efectos estandarizados para el *makespan* de la instancia i7j3k3-3.



**Gráfica 24.** Diagrama de residuos para el *makespan* de la instancia i7j3k3-3.



• **Análisis de la varianza para la instancia i7j2k5-1.**

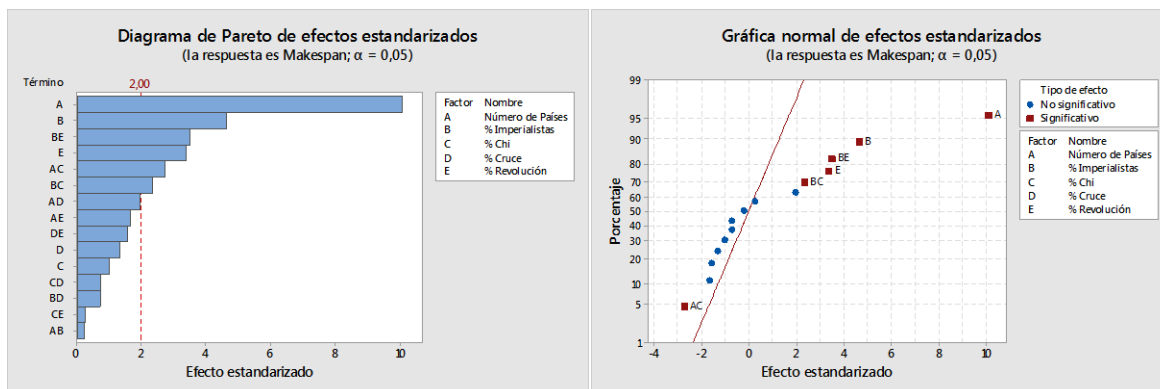
Con base en los datos presentados a continuación en la tabla 82 del ANOVA y en la gráfica 25 del diagrama de Pareto y la gráfica normal de efectos estandarizados se concluye que para la instancia i7j2k5-1 los factores número de países, % de imperialistas, % de revolución, la interacción entre número de países y %  $\xi$ , la

interacción entre % de imperialistas y %  $\xi$  y, por último, la interacción entre % de imperialistas y % de revolución influyen en la variable respuesta *makespan*. A su vez, el factor menos influyente es la interacción entre número de países y % de imperialistas.

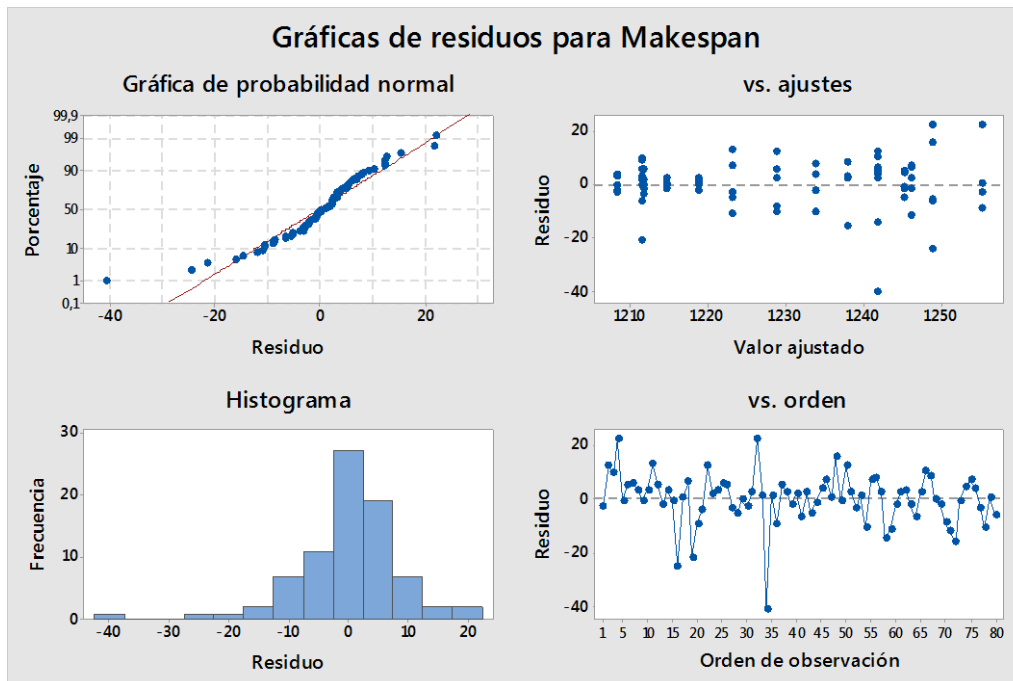
**Tabla 82.** ANOVA para la instancia i7j2k5-1.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
<b>Modelo</b>	15	18589,8	1239,3	11,54	0
<b>Lineal</b>	5	14771,7	2954,3	27,52	0
<b>Número de Países</b>	1	10927,8	10927,8	101,78	<b>0</b>
<b>% Imperialistas</b>	1	2322	2322	21,63	<b>0</b>
<b>% Chi</b>	1	108,1	108,1	1,01	0,319
<b>% Cruce</b>	1	189,1	189,1	1,76	0,189
<b>% Revolución</b>	1	1224,6	1224,6	11,41	<b>0,001</b>
<b>Interacciones de 2 términos</b>	10	3818,1	381,8	3,56	0,001
<b>Número de Países*% Imperialistas</b>	1	5,5	5,5	0,05	0,821
<b>Número de Países*% Chi</b>	1	800,1	800,1	7,45	<b>0,008</b>
<b>Número de Países*% Cruce</b>	1	409,5	409,5	3,81	0,055
<b>Número de Países*% Revolución</b>	1	300,3	300,3	2,8	0,099
<b>% Imperialistas*% Chi</b>	1	599,5	599,5	5,58	<b>0,021</b>
<b>% Imperialistas*% Cruce</b>	1	56,1	56,1	0,52	0,472
<b>% Imperialistas*% Revolución</b>	1	1320,3	1320,3	12,3	<b>0,001</b>
<b>% Chi*% Cruce</b>	1	56,1	56,1	0,52	0,472
<b>% Chi*% Revolución</b>	1	7,8	7,8	0,07	0,788
<b>% Cruce*% Revolución</b>	1	262,8	262,8	2,45	0,123
<b>Error</b>	64	6871,2	107,4		
<b>Total</b>	79	25461			

**Gráfica 25.** Diagrama de Pareto y gráfica normal de efectores estandarizados para el *makespan* de la instancia i7j2k5-1.



**Gráfica 26.** Diagrama de residuos para el *makespan* de la instancia i7j2k5-1.



En su orden, el factor más influyente es número de países, en segundo lugar, el % de imperialistas, en tercer lugar, la interacción entre % de imperialistas y % de revolución, en cuarto lugar, el % de revolución, en la quinta posición la interacción entre número de países y %  $\xi$ , y finalmente la interacción entre % de imperialistas y %  $\xi$ .

Finalmente, con base en la gráfica 26 se confirman los supuestos de normalidad e independencia de los datos.

- **Análisis de la varianza para la instancia i7j2k5-2.**

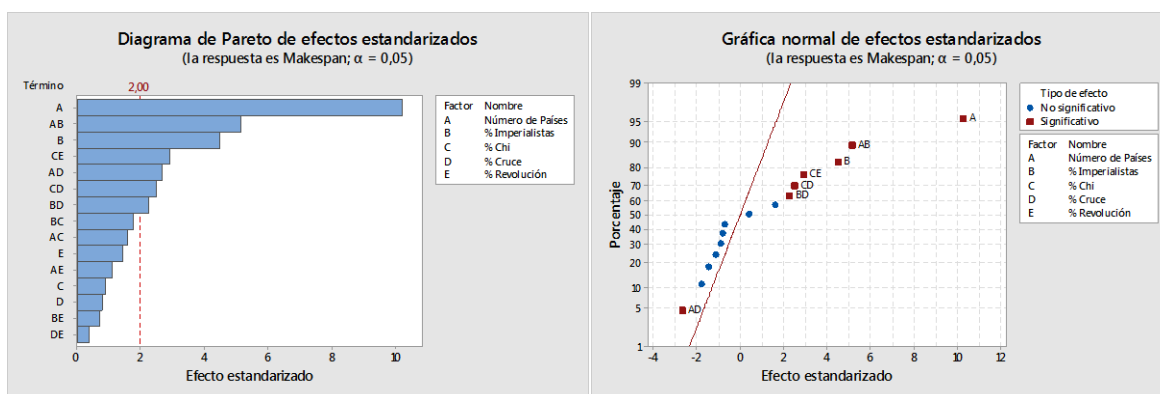
Para la instancia i7j2k5-2 observando los datos presentados más adelante en la tabla 83 del ANOVA y en la gráfica 27 del diagrama de Pareto y la gráfica normal de efectos estandarizados se concluye que los factores número de países, % de imperialistas, la interacción entre número de países y % de imperialistas, la interacción entre número de países y % de cruce, la interacción entre % de imperialistas y % de cruce y la interacción entre %  $\xi$  y % cruce y %  $\xi$  y %

revolución influyen en la variable respuesta *makespan*. Mientras tanto, el factor menos influyente es la interacción entre % de cruce y % de revolución.

**Tabla 83.** ANOVA para la instancia i7j2k5-2.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
<b>Modelo</b>	15	17313,2	1154,21	12,67	0
<b>Lineal</b>	5	11713,8	2342,77	25,72	0
<b>Número de Países</b>	1	9548,4	9548,45	104,84	<b>0</b>
<b>% Imperialistas</b>	1	1843,2	1843,2	20,24	<b>0</b>
<b>% Chi</b>	1	72,2	72,2	0,79	0,377
<b>% Cruce</b>	1	57,8	57,8	0,63	0,429
<b>% Revolución</b>	1	192,2	192,2	2,11	0,151
<b>Interacciones de 2 términos</b>	10	5599,3	559,93	6,15	0
<b>Número de Países*% Imperialistas</b>	1	2420	2420	26,57	<b>0</b>
<b>Número de Países*% Chi</b>	1	231,2	231,2	2,54	0,116
<b>Número de Países*% Cruce</b>	1	649,8	649,8	7,13	<b>0,01</b>
<b>Número de Países*% Revolución</b>	1	115,2	115,2	1,26	0,265
<b>% Imperialistas*% Chi</b>	1	296,4	296,45	3,26	0,076
<b>% Imperialistas*% Cruce</b>	1	470,5	470,45	5,17	<b>0,026</b>
<b>% Imperialistas*% Revolución</b>	1	48	48,05	0,53	0,47
<b>% Chi*% Cruce</b>	1	572,5	572,45	6,29	<b>0,015</b>
<b>% Chi*% Revolución</b>	1	781,3	781,25	8,58	<b>0,005</b>
<b>% Cruce*% Revolución</b>	1	14,4	14,45	0,16	0,692
<b>Error</b>	64	5828,8	91,08		
<b>Total</b>	79	23142			

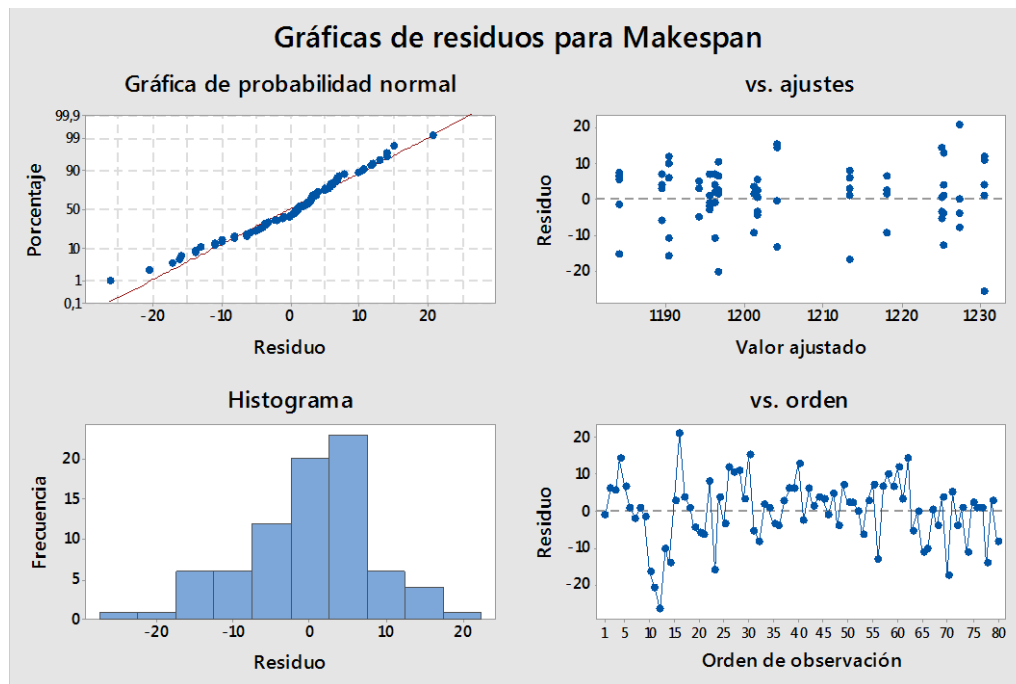
**Gráfica 27.** Diagrama de Pareto y gráfica normal de efector estandarizados para el *makespan* de la instancia i7j2k5-2.



El factor más influyente entre los ya mencionados es el número de países, seguido por la interacción entre número de países y % de imperialistas, en tercer lugar, el

% de imperialistas, en cuarto lugar, la interacción entre %  $\xi$  y % revolución, en quinto lugar, la interacción entre número de países y % de cruce, en el sexto puesto, la interacción entre %  $\xi$  y % cruce y finalmente, la interacción entre % de imperialistas y % de cruce. Por último, con base en la gráfica 28 de a continuación se confirman los supuestos de normalidad e independencia de los datos.

**Gráfica 28.** Diagrama de residuos para el *makespan* de la instancia i7j2k5-2.



- **Análisis de la varianza para la instancia i7j3k5-1.**

Continuando con la instancia i7j3k5-1, cuya ANOVA se presenta en la tabla 84 más adelante, se concluye con base en la tabla misma y en la gráfica 30 de más adelante también con el diagrama de Pareto y la gráfica normal de efectos estandarizados que los factores número de países, % de imperialistas, %  $\xi$ , % de revolución, la interacción entre número de países y %  $\xi$  y la interacción entre número de países y % de cruce influyen en la variable respuesta *makespan*. Por su parte, el factor menos influyente es el % de cruce.

Encabeza los factores más influyentes el %  $\xi$ , % de revolución, seguido por el % de imperialistas, luego el número de países, en cuarto lugar, la interacción entre número de países y % de cruce, en quinto lugar, el %  $\xi$  y finalmente, la interacción entre número de países y %  $\xi$ .

También, con base en la gráfica 29 presentada a continuación se confirman los supuestos de normalidad e independencia de los datos.

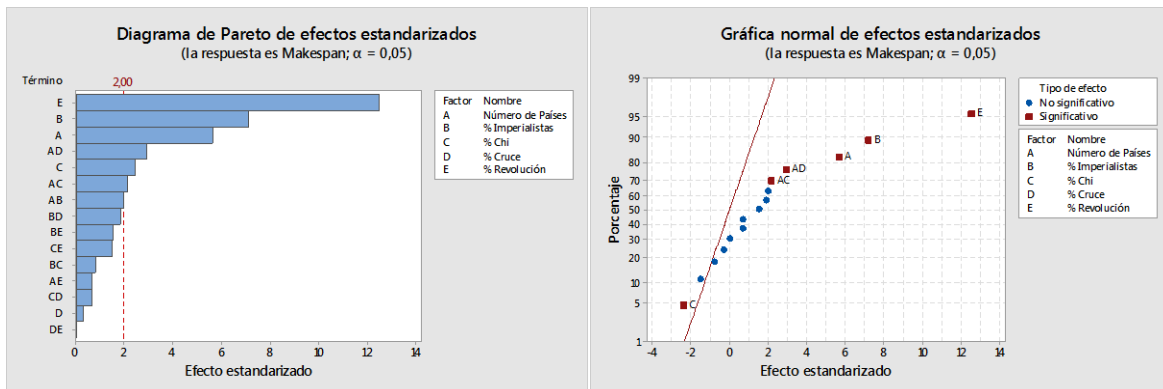
**Tabla 84.** ANOVA para la instancia i7j3k5-1.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Modelo	15	13096	873,07	18,17	0
Lineal	5	11808,9	2361,78	49,16	0
Número de Países	1	1531,2	1531,25	31,87	<b>0</b>
% Imperialistas	1	2464,2	2464,2	51,29	<b>0</b>
% Chi	1	281,3	281,25	5,85	<b>0,018</b>
% Cruce	1	5	5	0,1	0,748
% Revolución	1	7527,2	7527,2	156,67	<b>0</b>
Interacciones de 2 términos	10	1287,1	128,71	2,68	0,008
Número de Países*% Imperialistas	1	186	186,05	3,87	0,053
Número de Países*% Chi	1	217,8	217,8	4,53	<b>0,037</b>
Número de Países*% Cruce	1	414	414,05	8,62	<b>0,005</b>
Número de Países*% Revolución	1	22	22,05	0,46	0,501
% Imperialistas*% Chi	1	31,2	31,25	0,65	0,423
% Imperialistas*% Cruce	1	168,2	168,2	3,5	0,066
% Imperialistas*% Revolución	1	115,2	115,2	2,4	0,126
% Chi*% Cruce	1	22	22,05	0,46	0,501
% Chi*% Revolución	1	110,4	110,45	2,3	0,134
% Cruce*% Revolución	1	0	0	0	1
Error	64	3074,8	48,04		
Total	79	16170,8			

**Gráfica 29.** Diagrama de residuos para el *makespan* de la instancia i7j3k5-1.



**Gráfica 30.** Diagrama de Pareto y gráfica normal de efector estandarizados para el *makespan* de la instancia i7j3k5-1.



- Análisis de la varianza para la instancia i7j3k5-2.

Tabla 85. ANOVA para la instancia i7j3k5-2.

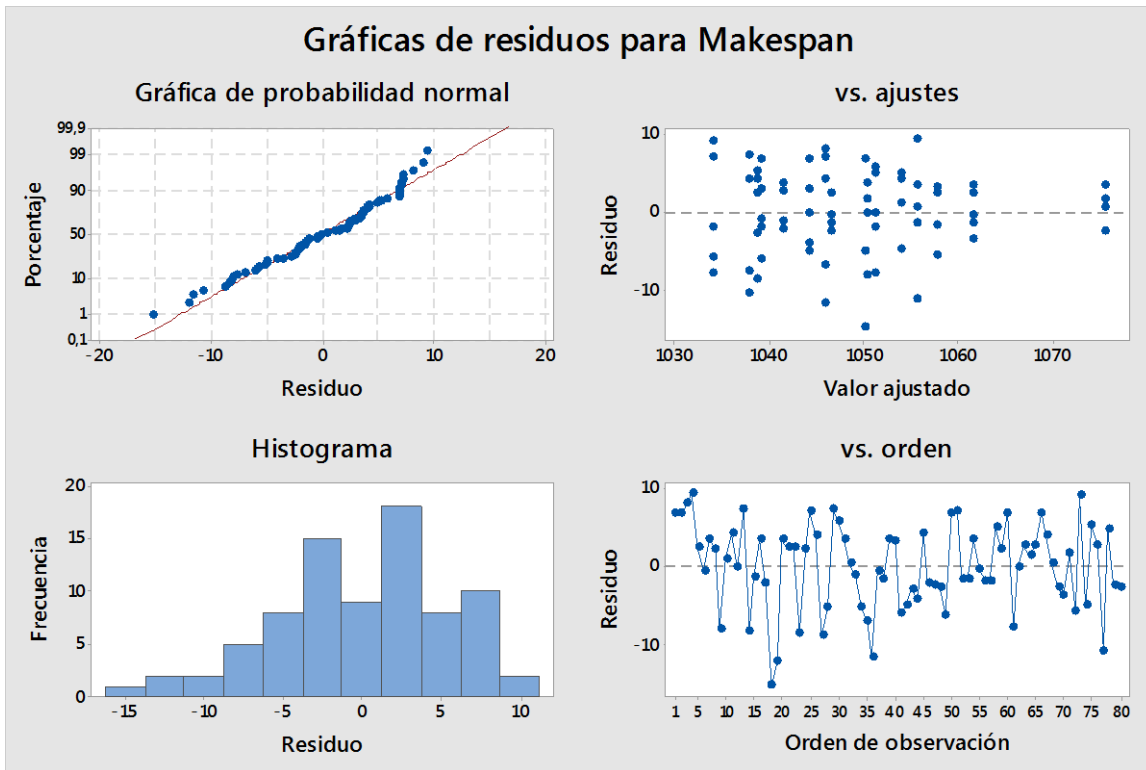
Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Modelo	15	8333,1	555,54	15,37	0
Lineal	5	6561,8	1312,37	36,32	0
Número de Países	1	4205	4205	116,36	<b>0</b>
% Imperialistas	1	387,2	387,2	10,71	<b>0,002</b>
% Chi	1	1140	1140,05	31,55	<b>0</b>
% Cruce	1	288,8	288,8	7,99	<b>0,006</b>
% Revolución	1	540,8	540,8	14,97	<b>0</b>
Interacciones de 2 términos	10	1771,3	177,13	4,9	0
Número de Países*% Imperialistas	1	2,5	2,45	0,07	0,795
Número de Países*% Chi	1	180	180	4,98	<b>0,029</b>
Número de Países*% Cruce	1	281,3	281,25	7,78	<b>0,007</b>
Número de Países*% Revolución	1	638,5	638,45	17,67	<b>0</b>
% Imperialistas*% Chi	1	135,2	135,2	3,74	0,058
% Imperialistas*% Cruce	1	36,5	36,45	1,01	0,319
% Imperialistas*% Revolución	1	130	130,05	3,6	0,062
% Chi*% Cruce	1	28,8	28,8	0,8	0,375
% Chi*% Revolución	1	72,2	72,2	2	0,162
% Cruce*% Revolución	1	266,4	266,45	7,37	<b>0,009</b>
Error	64	2312,8	36,14		
Total	79	10645,9			

Para la instancia i7j3k5-2 observando los datos presentados en la tabla 85 del ANOVA y en la gráfica 32, presentada a continuación, del diagrama de Pareto y la gráfica normal de efectos estandarizados se concluye que los factores número de países, % de imperialistas, %  $\xi$ , % de cruce, % de revolución, la interacción entre número de países y %  $\xi$ , la interacción entre número de países y % de cruce, la interacción entre número de países y % de revolución y la interacción entre % de cruce y % de revolución influyen en la variable respuesta *makespan*. Mientras tanto, el factor menos influyente la interacción entre el número de países y % de imperialistas.

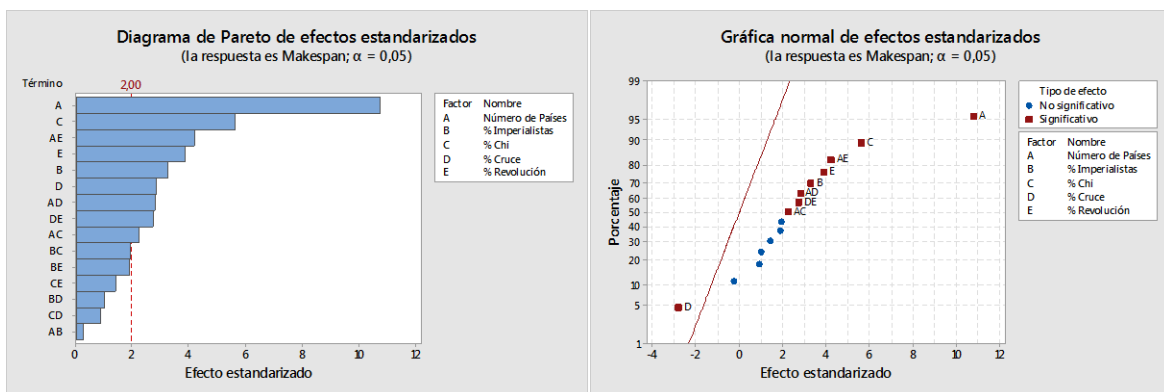
El factor más influyente entre los ya mencionados es el número de países, seguido por el %  $\xi$ , en tercer lugar, la interacción entre número de países y % de revolución, en cuarto lugar, % de revolución, seguidamente, % de imperialistas,

posteriormente, % de revolución, en séptimo lugar, la interacción entre número de países y % de cruce, en octavo lugar, la interacción entre % de cruce y % de revolución y finalmente, la interacción entre número de países y %  $\xi$ . Por último, con base en la gráfica 31 presentada a continuación se confirman los supuestos de normalidad e independencia de los datos.

**Gráfica 31.** Diagrama de residuos para el *makespan* de la instancia i7j3k5-2.



**Gráfica 32.** Diagrama de Pareto y gráfica normal de efectos estandarizados para el *makespan* de la instancia i7j3k5-2.



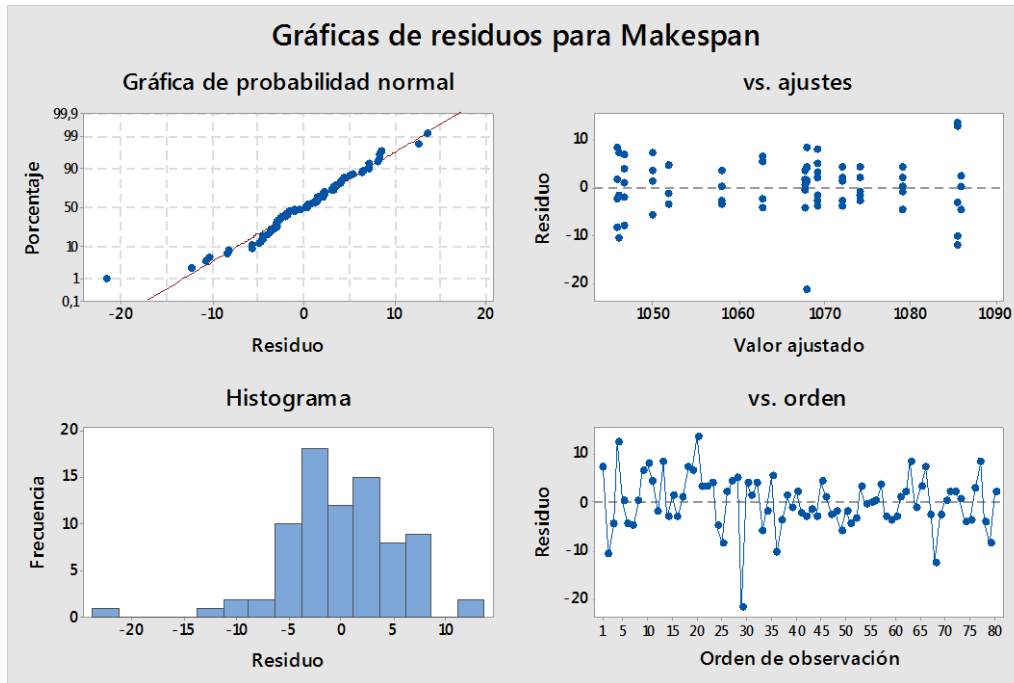
- Análisis de la varianza para la instancia i7j3k5-3.

Tabla 86. ANOVA para la instancia i7j3k5-3.

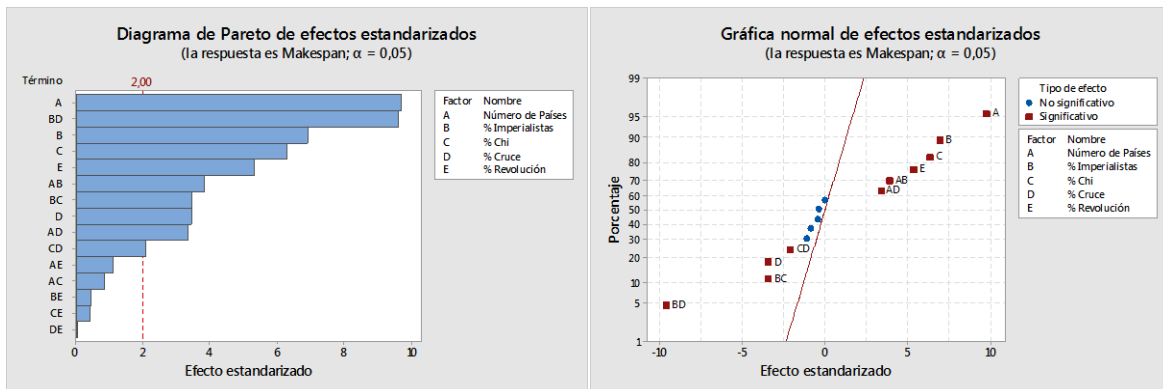
Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Modelo	15	13866,3	924,42	23,92	0
Lineal	5	8575,4	1715,09	44,38	0
Número de Países	1	3645	3645	94,32	0
% Imperialistas	1	1843,2	1843,2	47,7	0
% Chi	1	1531,2	1531,25	39,62	0
% Cruce	1	460,8	460,8	11,92	0,001
% Revolución	1	1095,2	1095,2	28,34	0
Interacciones de 2 términos	10	5290,9	529,09	13,69	0
Número de Países*% Imperialistas	1	572,4	572,45	14,81	0
Número de Países*% Chi	1	28,8	28,8	0,75	0,391
Número de Países*% Cruce	1	432,5	432,45	11,19	0,001
Número de Países*% Revolución	1	48,1	48,05	1,24	0,269
% Imperialistas*% Chi	1	460,8	460,8	11,92	0,001
% Imperialistas*% Cruce	1	3564,5	3564,45	92,24	0
% Imperialistas*% Revolución	1	8,5	8,45	0,22	0,642
% Chi*% Cruce	1	168,2	168,2	4,35	0,041
% Chi*% Revolución	1	7,2	7,2	0,19	0,667
% Cruce*% Revolución	1	0	0,05	0	0,971
Error	64	2473,2	38,64		
Total	79	16339,6			

Para el caso de la instancia i7j3k5-3 se concluye con base en la tabla 86 de ANOVA y la gráfica 34 del diagrama de Pareto y la gráfica normal de efectos estandarizados mostrada más adelante en la gráfica 34, presentada a continuación, que influyen la variable respuesta *makespan* en orden descendente de influjo los factores, número de países, y la interacción entre % de imperialistas y % de cruce, el % de imperialistas, %  $\xi$ , el % de revolución, la interacción entre número de países y % de imperialistas, la interacción entre % de imperialistas y %  $\xi$ , el % de cruce, la interacción entre el número de países y % de cruce y finalmente la interacción entre %  $\xi$  y % de cruce. El factor menos influyente es la interacción entre % de cruce y % de revolución; también se confirma con base en la gráfica 33 de más adelante los supuestos de normalidad e independencia de los datos.

**Gráfica 33.** Diagrama de residuos para el *makespan* de la instancia i7j3k5-3.



**Gráfica 34.** Diagrama de Pareto y gráfica normal de efectos estandarizados para el *makespan* de la instancia i7j3k5-3.

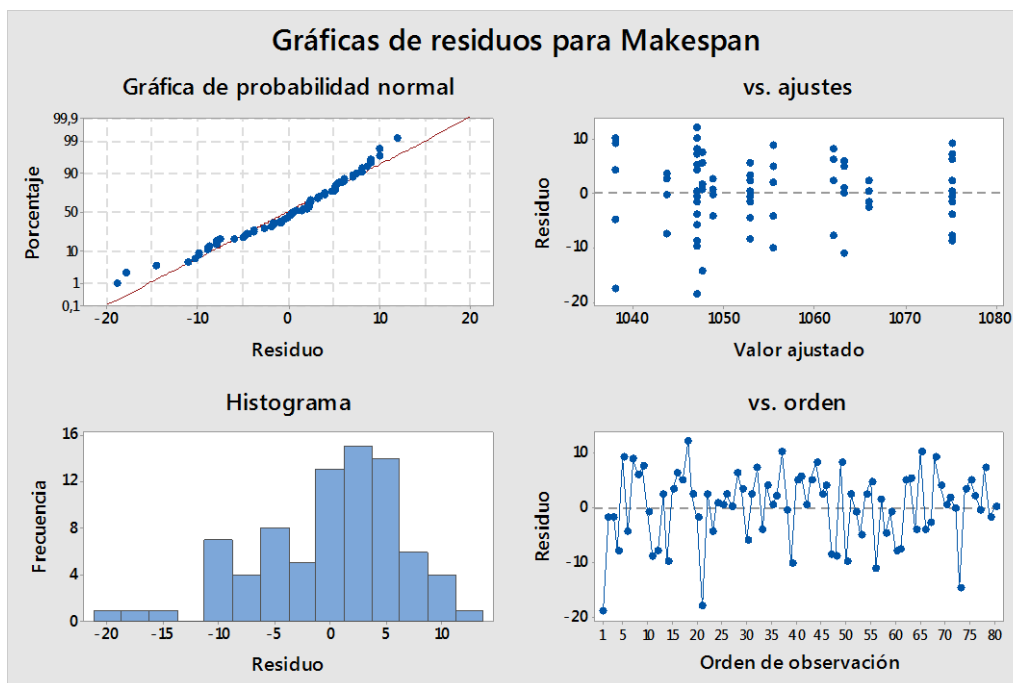


- Análisis de la varianza para la instancia i9j2k3-1.

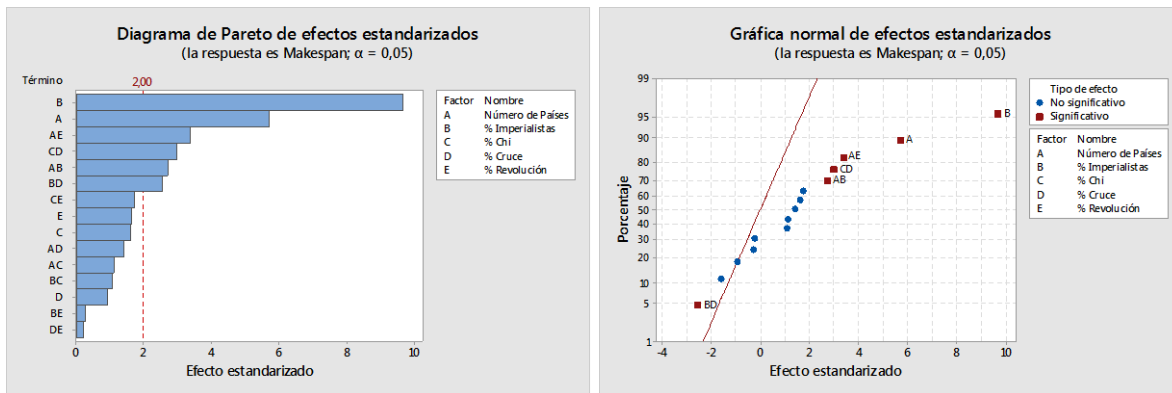
Tabla 87. ANOVA para la instancia i9j2k3-1.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Modelo	15	9020,5	601,37	11,59	0
Lineal	5	6852,2	1370,43	26,4	0
Número de Países	1	1683,6	1683,61	32,44	0
% Imperialistas	1	4851,6	4851,61	93,47	0
% Chi	1	132,6	132,61	2,55	0,115
% Cruce	1	46,5	46,51	0,9	0,347
% Revolución	1	137,8	137,81	2,66	0,108
Interacciones de 2 términos	10	2168,3	216,83	4,18	0
Número de Países*% Imperialistas	1	382,8	382,81	7,38	0,008
Número de Países*% Chi	1	66,6	66,61	1,28	0,262
Número de Países*% Cruce	1	103,5	103,51	1,99	0,163
Número de Países*% Revolución	1	588,6	588,61	11,34	0,001
% Imperialistas*% Chi	1	59,5	59,51	1,15	0,288
% Imperialistas*% Cruce	1	340,3	340,31	6,56	0,013
% Imperialistas*% Revolución	1	4,5	4,51	0,09	0,769
% Chi*% Cruce	1	465,6	465,61	8,97	0,004
% Chi*% Revolución	1	154	154,01	2,97	0,09
% Cruce*% Revolución	1	2,8	2,81	0,05	0,817
Error	64	3322	51,91		
Total	79	12342,5			

Gráfica 35. Diagrama de residuos para el *makespan* de la instancia i9j2k3-1.



**Gráfica 36.** Diagrama de Pareto y gráfica normal de efectos estandarizados para el *makespan* de la instancia i9j2k3-1.



Con base en los datos presentados en la tabla 87 del ANOVA y en la gráfica 36 del diagrama de Pareto y la gráfica normal de efectos estandarizados se concluye que para la instancia i9j2k3-1 los factores número de países, % de imperialistas, la interacción entre número de países y % de imperialistas, la interacción entre número de países y % de revolución, la interacción entre % de imperialistas y % de cruce y finalmente la interacción entre %  $\xi$  y % de cruce influyen en la variable respuesta *makespan* ya que el valor-p de éstas es menor que 0,05. A su vez, el factor menos influyente es la interacción entre % de cruce y % de revolución con 0,812.

En su orden, el factor más influyente es el % de imperialistas, en segundo lugar, el número de países, en tercer lugar, la interacción entre número de países y % de revolución, en cuarto lugar, la interacción entre número de países y % de imperialistas, en quinta posición, la interacción entre %  $\xi$  y % de cruce y finalmente, la interacción entre % de imperialistas y % de cruce.

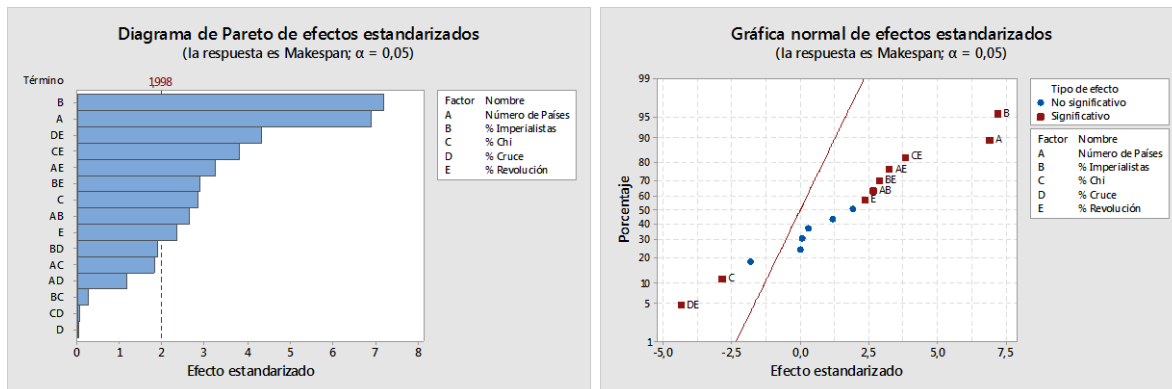
Finalmente, con base en la gráfica 35 se confirman los supuestos de normalidad e independencia de los datos. Esto significa que los mismos provienen de una población con distribución normal y que son independientes entre uno y otro ya no que se observa ninguna tendencia de los datos a simple vista.

- Análisis de la varianza para la instancia i9j2k3-2.

Tabla 88. ANOVA para la instancia i9j2k3-2.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Modelo	15	10034,2	668,95	12,06	0
Lineal	5	6280,5	1256,1	22,65	0
Número de Países	1	2645	2645	47,7	<b>0</b>
% Imperialistas	1	2880	2880	51,94	<b>0</b>
% Chi	1	451,3	451,25	8,14	<b>0,006</b>
% Cruce	1	0,1	0,05	0	0,976
% Revolución	1	304,2	304,2	5,49	<b>0,022</b>
Interacciones de 2 términos	10	3753,7	375,37	6,77	0
Número de Países*% Imperialistas	1	387,2	387,2	6,98	<b>0,01</b>
Número de Países*% Chi	1	186	186,05	3,36	0,072
Número de Países*% Cruce	1	76	76,05	1,37	0,246
Número de Países*% Revolución	1	583,2	583,2	10,52	<b>0,002</b>
% Imperialistas*% Chi	1	4	4,05	0,07	0,788
% Imperialistas*% Cruce	1	198,4	198,45	3,58	0,063
% Imperialistas*% Revolución	1	460,8	460,8	8,31	<b>0,005</b>
% Chi*% Cruce	1	0,2	0,2	0	0,952
% Chi*% Revolución	1	806,5	806,45	14,54	<b>0</b>
% Cruce*% Revolución	1	1051,2	1051,25	18,96	<b>0</b>
Error	64	3548,8	55,45		
Total	79	13583			

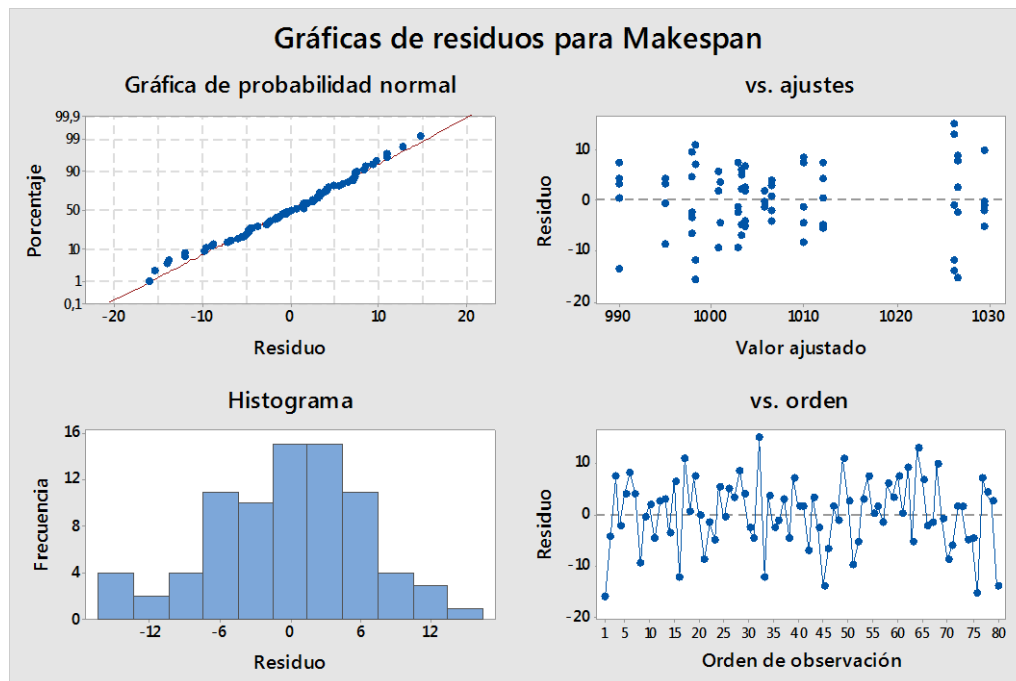
Gráfica 37. Diagrama de Pareto y gráfica normal de efector estandarizados para el *makespan* de la instancia i9j2k3-2.



Para la instancia i9j2k3-2 observando los datos presentados en la tabla 88 del ANOVA y en la gráfica 37 del diagrama de Pareto y la gráfica normal de efectos estandarizados se concluye que los factores número de países, % de

imperialistas,  $\% \xi$ ,  $\%$  de revolución, la interacción entre número de países y  $\%$  de imperialistas, la interacción entre número de países y  $\%$  de revolución, la interacción entre  $\%$  de imperialistas y  $\%$  de revolución, la interacción entre  $\% \xi$  y  $\%$  de revolución y la interacción entre  $\%$  de cruce y  $\%$  de revolución influyen en la variable respuesta *makespan*. Mientras tanto, el factor menos influyente es  $\%$  de cruce.

**Gráfica 38.** Diagrama de residuos para el *makespan* de la instancia i9j2k3-2.



El factor más influyente entre los ya mencionados es el  $\%$  de imperialistas, seguido por el número de países, en tercer lugar, la interacción entre  $\%$  de cruce y  $\%$  de revolución, luego, la interacción entre  $\% \xi$  y  $\%$  de revolución, en quinto lugar, la interacción entre número de países y  $\%$  de revolución, el siguiente es la interacción entre  $\%$  de imperialistas y  $\%$  de revolución, seguidamente, el  $\% \xi$ , en octavo lugar, la interacción entre número de países y  $\%$  de imperialistas y finalmente, el  $\%$  de revolución. Por último, con base en la gráfica 38 se confirman los supuestos de normalidad e independencia de los datos.

- **Análisis de la varianza para la instancia i9j3k3-1.**

Continuando con la instancia i9j3k3-1, cuya ANOVA se presenta en la tabla 89, se concluye con base en la tabla misma y en la gráfica 39 del diagrama de Pareto y la gráfica normal de efectos estandarizados que los factores número de países, % de imperialistas, % de revolución, la interacción entre % de imperialistas y % de revolución, la interacción entre %  $\xi$  y % de cruce y la interacción entre % de cruce y % de revolución influyen en la variable respuesta *makespan*. Por su parte, el factor menos influyente es la interacción entre el número de países y el % de imperialistas.

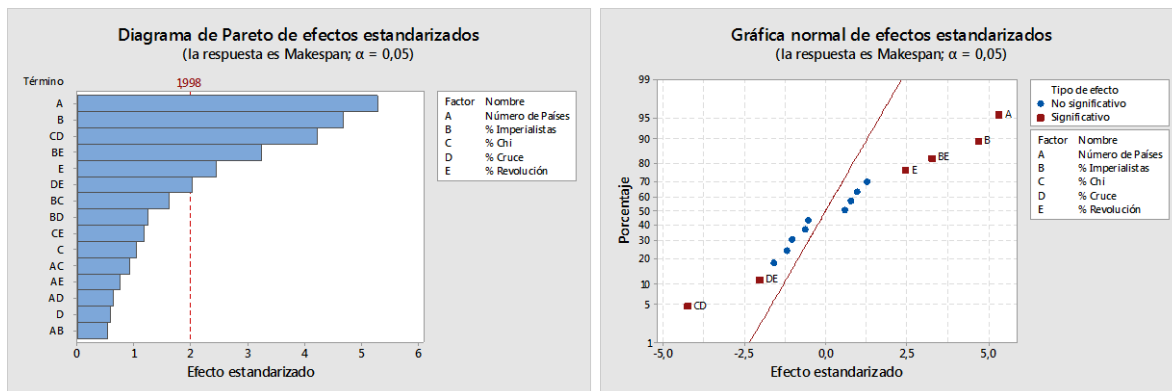
Encabeza los factores más influyentes el número de países, seguido el % de imperialistas, luego la interacción entre %  $\xi$  y % de cruce, en cuarto lugar, la interacción entre % de imperialistas y % de revolución, en quinto lugar, el % revolución y, por último, la interacción entre % de cruce y % de revolución.

También, con base en la gráfica 40 presentada a continuación se confirman los supuestos de normalidad e independencia de los datos.

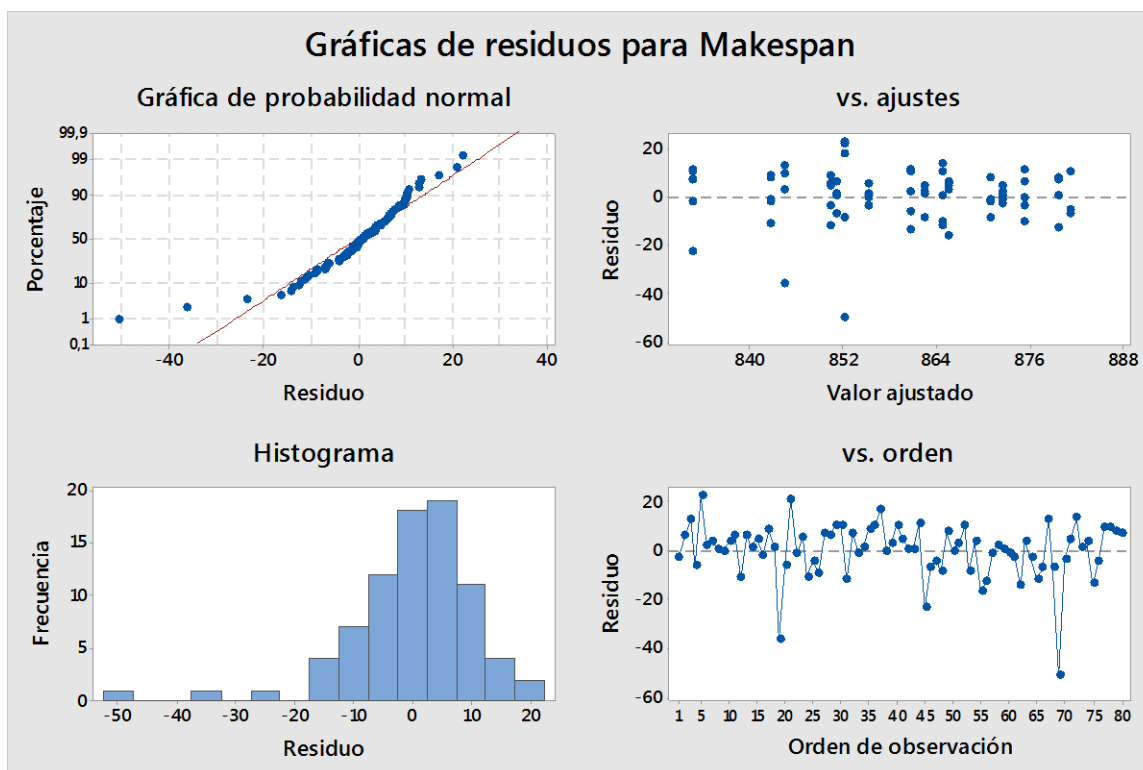
**Tabla 89.** ANOVA para la instancia i9j3k3-1.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
<b>Modelo</b>	15	14814,7	987,65	6,53	0
<b>Lineal</b>	5	8686,6	1737,31	11,49	0
<b>Número de Países</b>	1	4248,6	4248,61	28,11	<b>0</b>
<b>% Imperialistas</b>	1	3315,3	3315,31	21,93	<b>0</b>
<b>% Chi</b>	1	165,3	165,31	1,09	0,3
<b>% Cruce</b>	1	52,8	52,81	0,35	0,557
<b>% Revolución</b>	1	904,5	904,51	5,98	<b>0,017</b>
<b>Interacciones de 2 términos</b>	10	6128,1	612,81	4,05	0
<b>Número de Países*% Imperialistas</b>	1	43,5	43,51	0,29	0,593
<b>Número de Países*% Chi</b>	1	132,6	132,61	0,88	0,352
<b>Número de Países*% Cruce</b>	1	59,5	59,51	0,39	0,533
<b>Número de Países*% Revolución</b>	1	86,1	86,11	0,57	0,453
<b>% Imperialistas*% Chi</b>	1	391,6	391,61	2,59	0,112
<b>% Imperialistas*% Cruce</b>	1	241,5	241,51	1,6	0,211
<b>% Imperialistas*% Revolución</b>	1	1611	1611,01	10,66	<b>0,002</b>
<b>% Chi*% Cruce</b>	1	2726,1	2726,11	18,04	<b>0</b>
<b>% Chi*% Revolución</b>	1	214,5	214,51	1,42	0,238
<b>% Cruce*% Revolución</b>	1	621,6	621,61	4,11	<b>0,047</b>
<b>Error</b>	64	9674	151,16		
<b>Total</b>	79	24488,7			

**Gráfica 39.** Diagrama de Pareto y gráfica normal de efector estandarizados para el *makespan* de la instancia i9j3k3-1.



Gráfica 40. Diagrama de residuos para el *makespan* de la instancia i9j3k3-1.

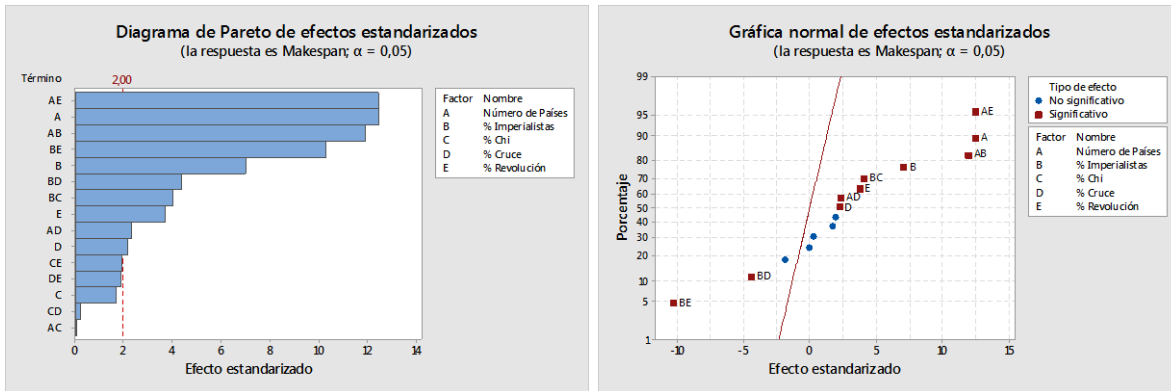


- **Análisis de la varianza para la instancia i9j3k3-2.**

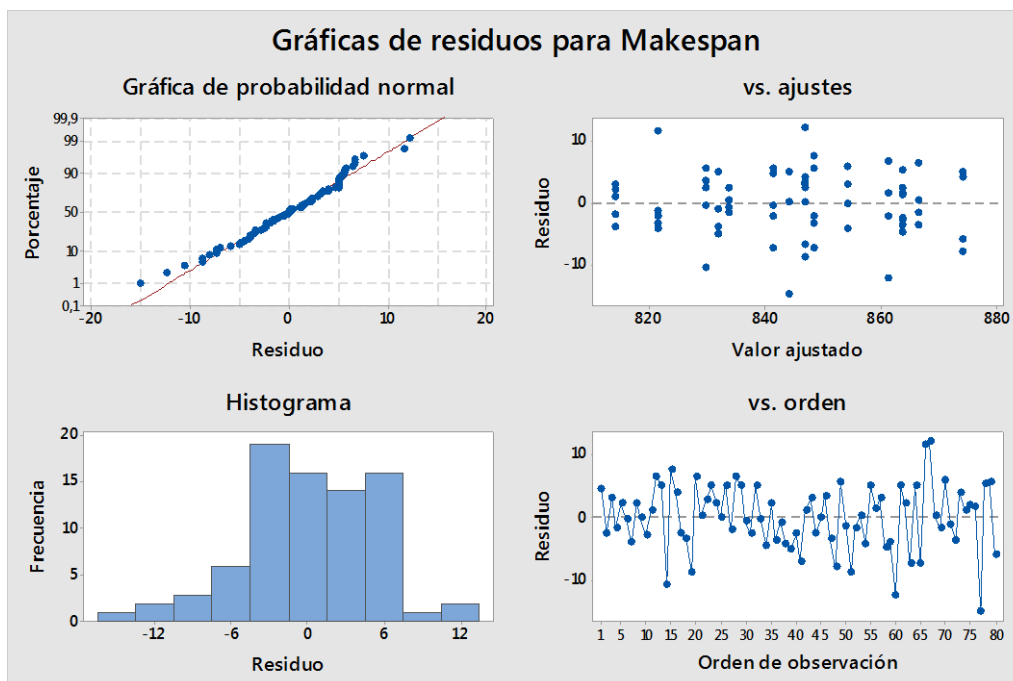
Tabla 90. ANOVA para la instancia i9j3k3-2.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
<b>Modelo</b>	15	22063,1	1470,88	45,27	0
<b>Lineal</b>	5	7358,7	1471,75	45,29	0
<b>Número de Países</b>	1	5056,2	5056,2	155,61	<b>0</b>
<b>% Imperialistas</b>	1	1602	1602,05	49,3	<b>0</b>
<b>% Chi</b>	1	92,4	92,45	2,85	0,097
<b>% Cruce</b>	1	156,8	156,8	4,83	<b>0,032</b>
<b>% Revolución</b>	1	451,3	451,25	13,89	<b>0</b>
<b>Interacciones de 2 términos</b>	10	14704,4	1470,44	45,25	0
<b>Número de Países*% Imperialistas</b>	1	4620,8	4620,8	142,21	<b>0</b>
<b>Número de Países*% Chi</b>	1	0,2	0,2	0,01	0,938
<b>Número de Países*% Cruce</b>	1	174,1	174,05	5,36	<b>0,024</b>
<b>Número de Países*% Revolución</b>	1	5056,2	5056,2	155,61	<b>0</b>
<b>% Imperialistas*% Chi</b>	1	530,5	530,45	16,32	<b>0</b>
<b>% Imperialistas*% Cruce</b>	1	627,2	627,2	19,3	<b>0</b>
<b>% Imperialistas*% Revolución</b>	1	3458,4	3458,45	106,43	<b>0</b>
<b>% Chi*% Cruce</b>	1	1,8	1,8	0,06	0,815
<b>% Chi*% Revolución</b>	1	120	120,05	3,69	0,059
<b>% Cruce*% Revolución</b>	1	115,2	115,2	3,55	0,064
<b>Error</b>	64	2079,6	32,49		
<b>Total</b>	79	24142,8			

**Gráfica 41.** Diagrama de Pareto y gráfica normal de efectos estandarizados para el *makespan* de la instancia i9j3k3-2.



**Gráfica 42.** Diagrama de residuos para el *makespan* de la instancia i9j3k3-2.



Analizando la tabla 90 donde se presenta la ANOVA y la gráfica 41 del diagrama de Pareto y la gráfica normal de efectos estandarizados de la instancia i9j3k3-2 se concluye que los factores más influyentes en la variable respuesta *makespan* son el número de países, el % de imperialistas, el % de cruce, el % de revolución, la interacción entre el número de países y % de imperialistas, la interacción entre el

número de países y % de cruce, la interacción entre el número de países y % de revolución y por último, la interacción entre el % de imperialistas y todos los demás factores. Por el contrario, el factor menos influyente es la interacción entre el número de países y %  $\xi$ .

También se determina que el factor más influyente es la interacción entre el número de países y % de revolución, luego, el % de imperialistas, seguido por la interacción entre el número de países y % de imperialistas, tras estas, aparece la interacción entre el % de imperialistas y el % de revolución, seguidamente el porcentaje de imperialistas, luego, la interacción entre el % de imperialistas y el % de cruce, seguido por la interacción entre el % de imperialistas y el %  $\xi$ , el % de revolución luego, la interacción entre el número de países y % de cruce posteriormente, y finalmente, el % de cruce . A su vez, con base en la gráfica 42 se confirman los supuestos de normalidad e independencia de los datos.

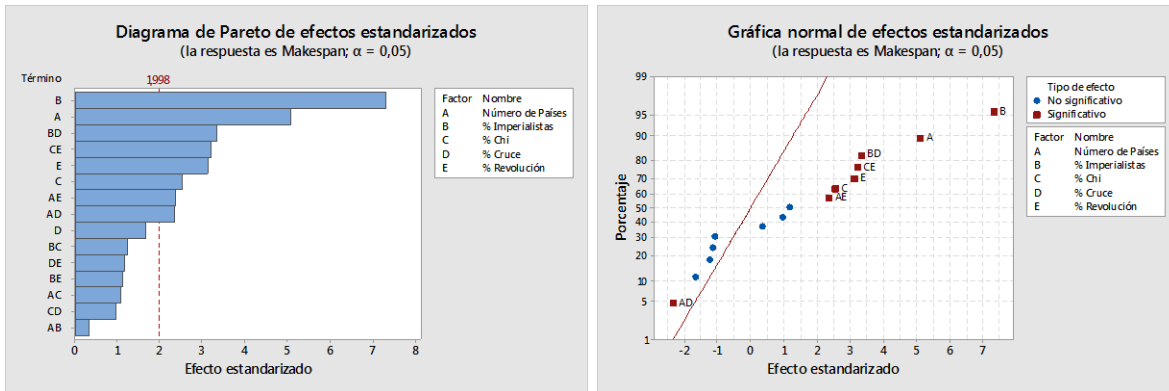
- Análisis de la varianza para la instancia i9j3k3-3.

Tabla 91. ANOVA para la instancia i9j3k3-3.

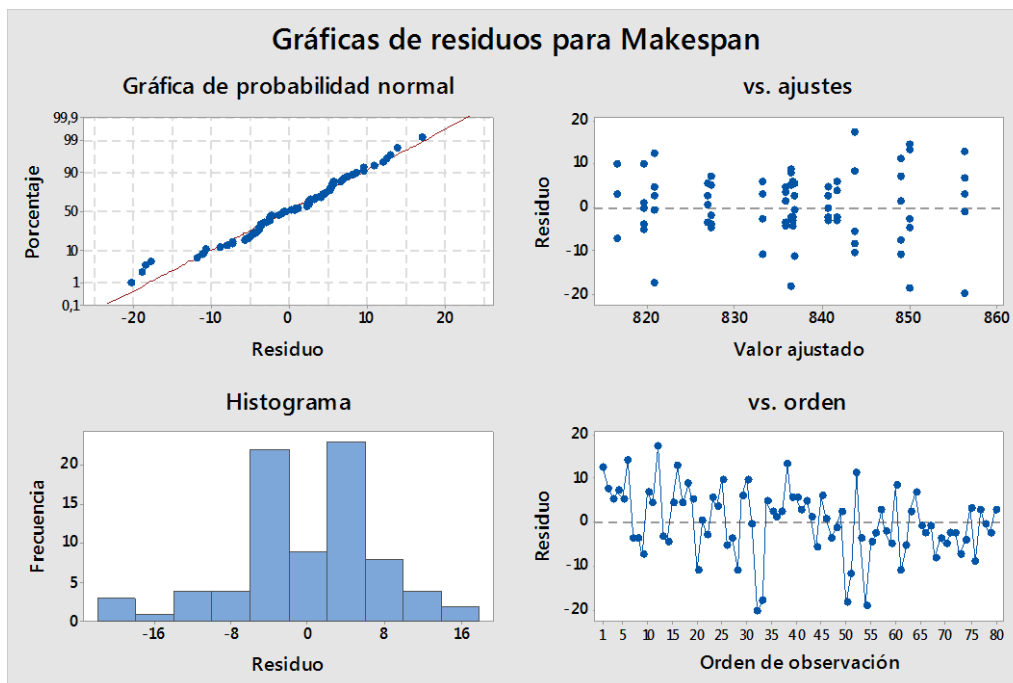
Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Modelo	15	9773,6	651,57	9,18	0
Lineal	5	7007,4	1401,49	19,74	0
Número de Países	1	1843,2	1843,2	25,96	0
% Imperialistas	1	3808,8	3808,8	53,65	0
% Chi	1	460,8	460,8	6,49	0,013
% Cruce	1	198,4	198,45	2,8	0,099
% Revolución	1	696,2	696,2	9,81	0,003
Interacciones de 2 términos	10	2766,1	276,61	3,9	0
Número de Países*% Imperialistas	1	8,4	8,45	0,12	0,731
Número de Países*% Chi	1	84	84,05	1,18	0,281
Número de Países*% Cruce	1	387,2	387,2	5,45	0,023
Número de Países*% Revolución	1	396,1	396,05	5,58	0,021
% Imperialistas*% Chi	1	110,5	110,45	1,56	0,217
% Imperialistas*% Cruce	1	793,8	793,8	11,18	0,001
% Imperialistas*% Revolución	1	92,4	92,45	1,3	0,258
% Chi*% Cruce	1	64,8	64,8	0,91	0,343
% Chi*% Revolución	1	732,1	732,05	10,31	0,002
% Cruce*% Revolución	1	96,8	96,8	1,36	0,247
Error	64	4544	71		
Total	79	14317,6			

Para el caso de la instancia i9j3k3-3 se concluye con base en la tabla 90 de ANOVA y la gráfica 43 del diagrama de Pareto y la gráfica normal de efectos estandarizados, presentada a continuación, que influyen la variable respuesta *makespan* en orden descendente de influjo los factores son el % de imperialistas, el número de países, la interacción entre % de imperialistas y % de cruce, la interacción entre %  $\xi$  y % de revolución, el % de revolución, el %  $\xi$ , la interacción entre número de países y el % de revolución y la interacción entre número de países y % de cruce. El factor menos influyente es la interacción entre número de países y % el % de imperialistas; también se confirma con base en la gráfica 44 los supuestos de normalidad e independencia de los datos.

**Gráfica 43.** Diagrama de Pareto y gráfica normal de efecto estandarizados para el *makespan* de la instancia i9j3k3-3.



**Gráfica 44.** Diagrama de residuos para el *makespan* de la instancia i9j3k3-3.



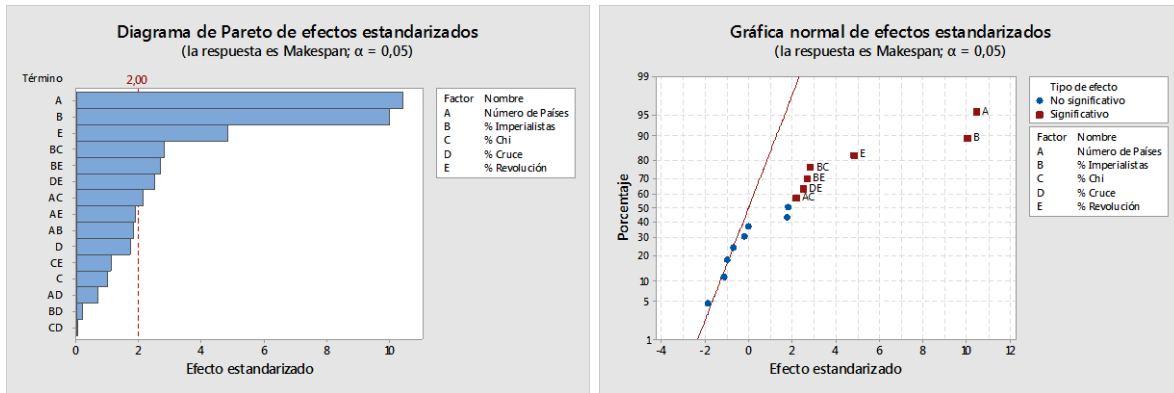
- **Análisis de la varianza para la instancia i9j2k5-1.**

Con base en los datos presentados a continuación en la tabla 92 del ANOVA y en la gráfica 45 del diagrama de Pareto y la gráfica normal de efectos estandarizados se concluye que para la instancia i9j2k5-1 los factores número de países, % de imperialistas, % de revolución, la interacción entre número de países y %  $\xi$ , la interacción entre % de imperialistas y %  $\xi$ , la interacción entre % de imperialistas y % de revolución y finalmente, la interacción entre % de cruce y % de revolución influyen en la variable respuesta *makespan*. A su vez, el factor menos influyente es la interacción entre %  $\xi$  y % de cruce.

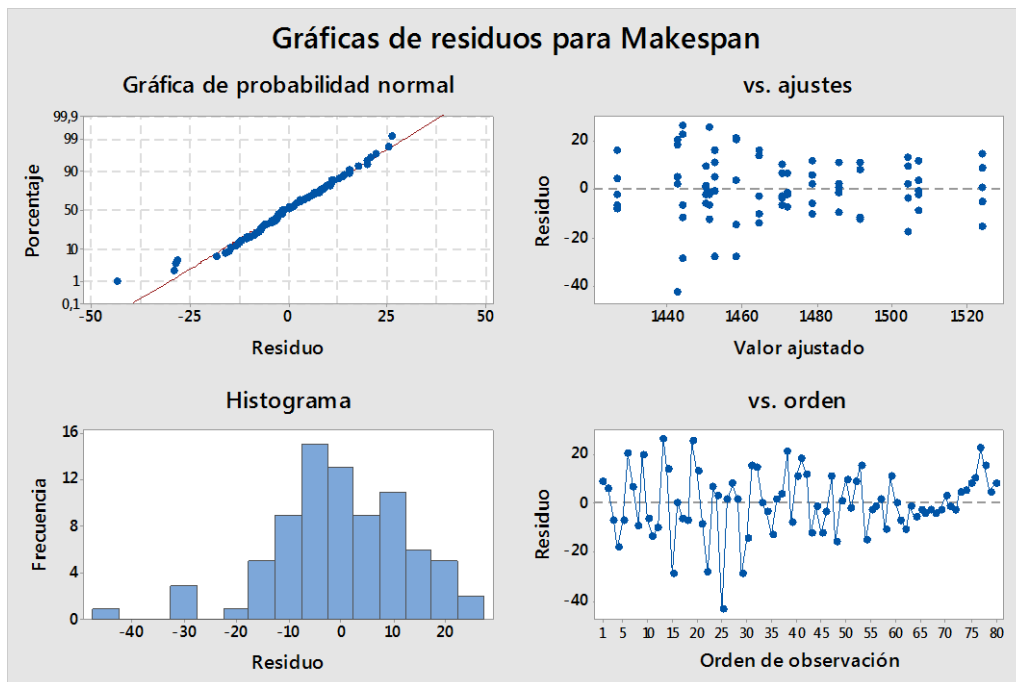
**Tabla 92.** ANOVA para la instancia i9j2k5-1.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
<b>Modelo</b>	15	54082,2	3605,5	18,07	0
<b>Lineal</b>	5	47151,6	9430,3	47,26	0
<b>Número de Países</b>	1	21681,1	21681,1	108,65	<b>0</b>
<b>% Imperialistas</b>	1	20002,8	20002,8	100,24	<b>0</b>
<b>% Chi</b>	1	201,6	201,6	1,01	0,319
<b>% Cruce</b>	1	599,5	599,5	3	0,088
<b>% Revolución</b>	1	4666,5	4666,5	23,39	<b>0</b>
<b>Interacciones de 2 términos</b>	10	6930,6	693,1	3,47	0,001
<b>Número de Países*% Imperialistas</b>	1	655,5	655,5	3,28	0,075
<b>Número de Países*% Chi</b>	1	931,6	931,6	4,67	<b>0,034</b>
<b>Número de Países*% Cruce</b>	1	99	99	0,5	0,484
<b>Número de Países*% Revolución</b>	1	702,1	702,1	3,52	0,065
<b>% Imperialistas*% Chi</b>	1	1593,1	1593,1	7,98	<b>0,006</b>
<b>% Imperialistas*% Cruce</b>	1	7,8	7,8	0,04	0,844
<b>% Imperialistas*% Revolución</b>	1	1436,5	1436,5	7,2	<b>0,009</b>
<b>% Chi*% Cruce</b>	1	0,3	0,3	0	0,969
<b>% Chi*% Revolución</b>	1	248,5	248,5	1,25	0,269
<b>% Cruce*% Revolución</b>	1	1256,1	1256,1	6,29	<b>0,015</b>
<b>Error</b>	64	12771,2	199,6		
<b>Total</b>	79	66853,4			

**Gráfica 45.** Diagrama de Pareto y gráfica normal de efecto estandarizados para el *makespan* de la instancia i9j2k5-1.



**Gráfica 46.** Diagrama de residuos para el *makespan* de la instancia i9j2k5-1.



En su orden, el factor más influyente es número de países, en segundo lugar, el % de imperialistas, en tercer lugar, el % de revolución, en cuarto lugar, % de imperialistas y %  $\xi$ , en la quinta posición, la interacción entre % de imperialistas y % de revolución, en la sexta posición, la interacción entre % de cruce y % de revolución, y finalmente la interacción entre número de países y %  $\xi$ .

Finalmente, con base en la gráfica 46 se confirman los supuestos de normalidad e independencia de los datos.

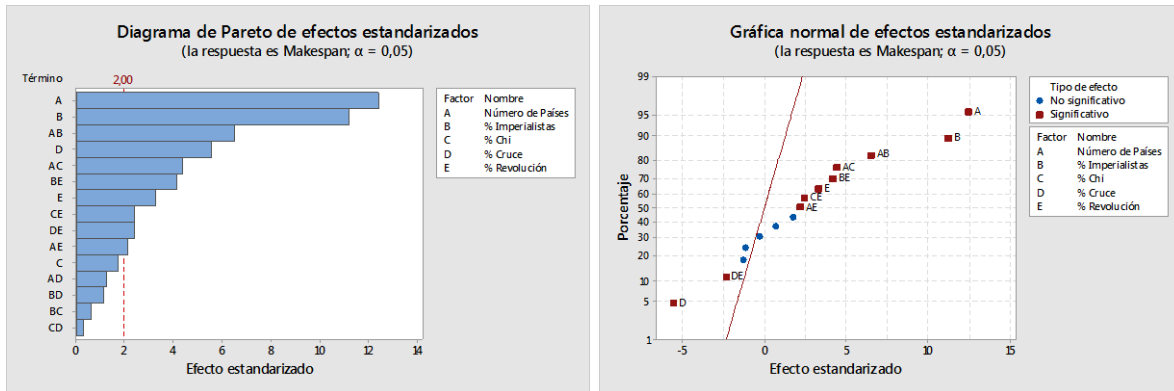
- **Análisis de la varianza para la instancia i9j2k5-2.**

Para la instancia i9j2k5-2 observando los datos presentados más adelante en la tabla 93 del ANOVA y en la gráfica 47 del diagrama de Pareto y la gráfica normal de efectos estandarizados se concluye que los factores número de países, % de imperialistas, el % de cruce, el % de revolución, la interacción entre número de países y % de imperialistas, la interacción entre número de países y % de cruce, la interacción entre número de países y % de revolución, la interacción entre % de imperialistas y % de revolución y la interacción entre %  $\xi$  y % de revolución y la interacción entre % cruce y % de revolución influyen en la variable respuesta *makespan*. Mientras tanto, el factor menos influyente es la interacción entre %  $\xi$  y % de cruce.

**Tabla 93.** ANOVA para la instancia i9j2k5-2.

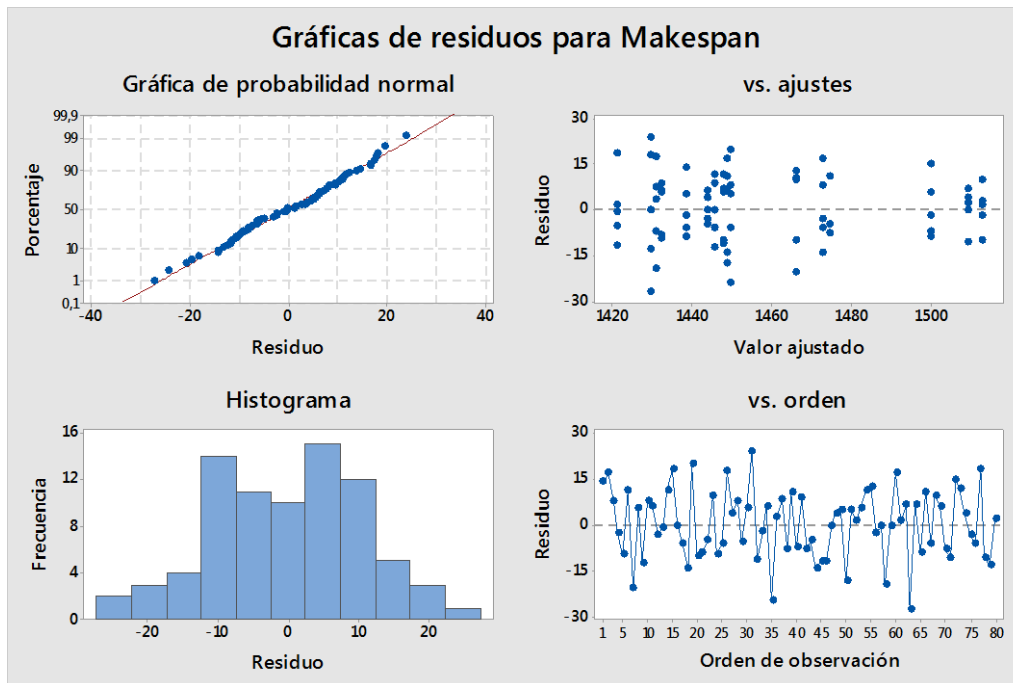
Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
<b>Modelo</b>	15	62506,7	4167,1	28,27	0
<b>Lineal</b>	5	47964,1	9592,8	65,08	0
<b>Número de Países</b>	1	22781,2	22781,2	154,56	<b>0</b>
<b>% Imperialistas</b>	1	18605	18605	126,23	<b>0</b>
<b>% Chi</b>	1	451,2	451,2	3,06	0,085
<b>% Cruce</b>	1	4560,2	4560,2	30,94	<b>0</b>
<b>% Revolución</b>	1	1566,4	1566,4	10,63	<b>0,002</b>
<b>Interacciones de 2 términos</b>	10	14542,6	1454,3	9,87	0
<b>Número de Países*% Imperialistas</b>	1	6265,8	6265,8	42,51	<b>0</b>
<b>Número de Países*% Chi</b>	1	2856,1	2856,1	19,38	<b>0</b>
<b>Número de Países*% Cruce</b>	1	245	245	1,66	0,202
<b>Número de Países*% Revolución</b>	1	684,4	684,4	4,64	<b>0,035</b>
<b>% Imperialistas*% Chi</b>	1	64,8	64,8	0,44	0,51
<b>% Imperialistas*% Cruce</b>	1	198,4	198,4	1,35	0,25
<b>% Imperialistas*% Revolución</b>	1	2508,8	2508,8	17,02	<b>0</b>
<b>% Chi*% Cruce</b>	1	16,2	16,2	0,11	0,741
<b>% Chi*% Revolución</b>	1	858	858	5,82	<b>0,019</b>
<b>% Cruce*% Revolución</b>	1	845	845	5,73	<b>0,02</b>
<b>Error</b>	64	9433,2	147,4		
<b>Total</b>	79	71939,9			

**Gráfica 47.** Diagrama de Pareto y gráfica normal de efectos estandarizados para el *makespan* de la instancia *i9j2k5-2*.



El factor más influyente entre los ya mencionados es el número de países, seguido por el % de imperialistas, en tercer lugar, la interacción entre número de países y % de imperialistas, en cuarto lugar, el porcentaje de cruce, la interacción entre número de países y %  $\xi$ , en quinto lugar, la interacción entre % de imperialistas y % de revolución, en el sexto puesto, en séptimo, el % de revolución, luego, la interacción entre %  $\xi$  y % de revolución, seguidamente, la interacción entre % cruce y % de revolución y finalmente, la interacción entre número de países y % de revolución. Por último, con base en la gráfica 48 de a continuación se confirman los supuestos de normalidad e independencia de los datos.

**Gráfica 48.** Diagrama de residuos para el *makespan* de la instancia i9j2k5-2.



- **Análisis de la varianza para la instancia i9j3k5-1.**

Continuando con la instancia i9j3k5-1, cuya ANOVA se presenta en la tabla 94 más adelante, se concluye con base en la tabla misma y en la gráfica 49 de más adelante también con el diagrama de Pareto y la gráfica normal de efectos estandarizados que los factores número de países, % de imperialistas, %  $\xi$ , % de cruce, % de revolución, la interacción entre número de países y % de imperialistas, la interacción entre % de imperialistas y %  $\xi$ , la interacción entre % de imperialistas y % de revolución y por último, la interacción entre % de cruce y % de revolución influyen en la variable respuesta *makespan*. Por su parte, el factor menos influyente es la interacción entre %  $\xi$  y % de cruce.

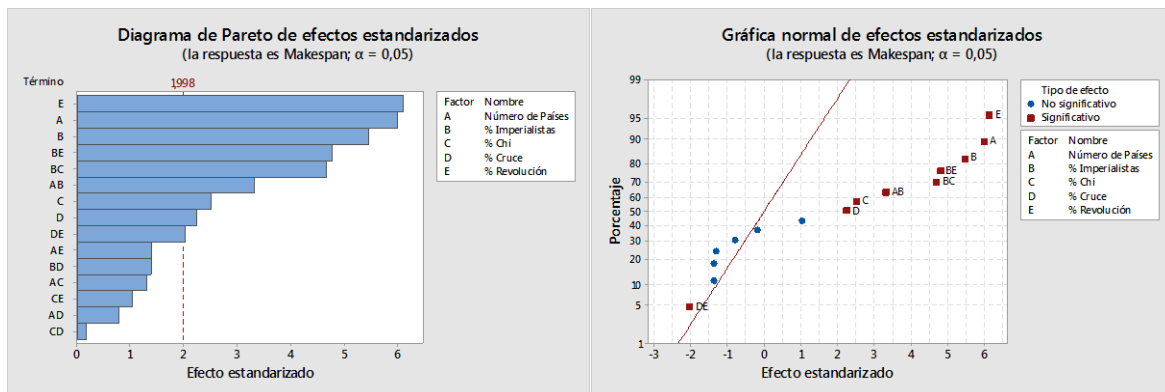
Encabeza los factores más influyentes el % de revolución, seguido por el número de países, luego, el % de imperialistas, en cuarto lugar, la interacción entre % de imperialistas y % de revolución, en quinto lugar, la interacción entre el % de imperialistas y el %  $\xi$ , la interacción entre número de países y el % de imperialistas

posteriormente, seguido de estos, el %  $\xi$ , luego, el % de cruce y finalmente la interacción entre el % de cruce y el % de revolución. También, con base en la gráfica 50 presentada a continuación se confirman los supuestos de normalidad e independencia de los datos.

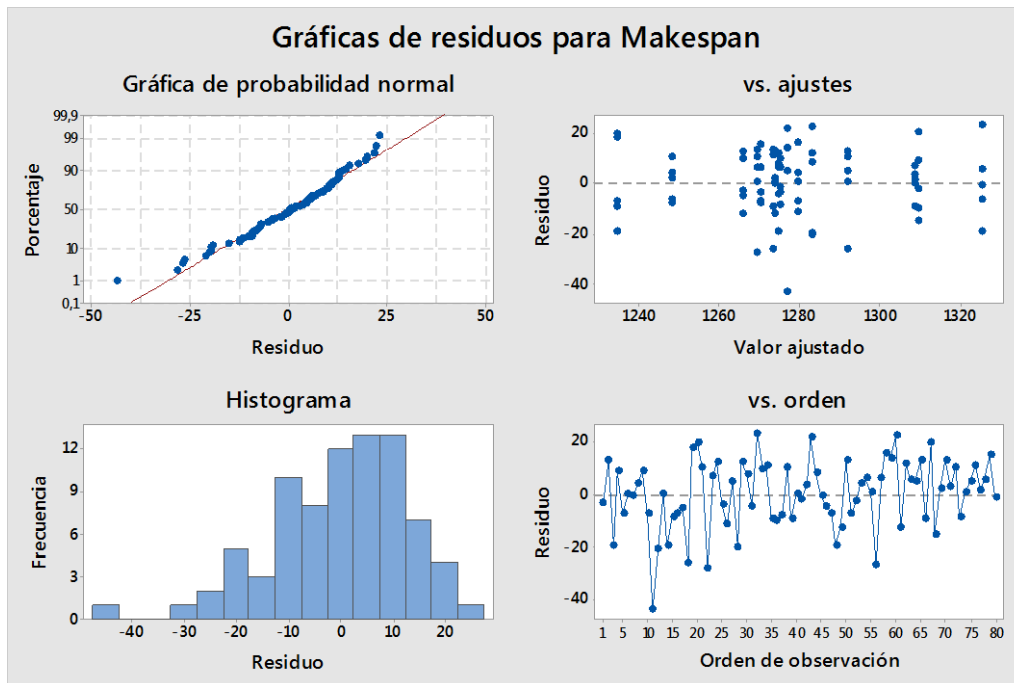
**Tabla 94.** ANOVA para la instancia i9j3k5-1.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
<b>Modelo</b>	15	37352,7	2490,18	12,1	0
<b>Lineal</b>	5	23513,8	4702,77	22,86	0
<b>Número de Países</b>	1	7372,8	7372,8	35,84	<b>0</b>
<b>% Imperialistas</b>	1	6125	6125	29,77	<b>0</b>
<b>% Chi</b>	1	1296	1296,05	6,3	<b>0,015</b>
<b>% Cruce</b>	1	1036,8	1036,8	5,04	<b>0,028</b>
<b>% Revolución</b>	1	7683,2	7683,2	37,35	<b>0</b>
<b>Interacciones de 2 términos</b>	10	13838,9	1383,89	6,73	0
<b>Número de Países*% Imperialistas</b>	1	2268,4	2268,45	11,03	<b>0,001</b>
<b>Número de Países*% Chi</b>	1	352,8	352,8	1,71	0,195
<b>Número de Países*% Cruce</b>	1	130,1	130,05	0,63	0,43
<b>Número de Países*% Revolución</b>	1	396,1	396,05	1,93	0,17
<b>% Imperialistas*% Chi</b>	1	4500	4500	21,87	<b>0</b>
<b>% Imperialistas*% Cruce</b>	1	396	396,05	1,93	0,17
<b>% Imperialistas*% Revolución</b>	1	4712,4	4712,45	22,91	<b>0</b>
<b>% Chi*% Cruce</b>	1	7,2	7,2	0,03	0,852
<b>% Chi*% Revolución</b>	1	217,8	217,8	1,06	0,307
<b>% Cruce*% Revolución</b>	1	858,1	858,05	4,17	<b>0,045</b>
<b>Error</b>	64	13166,8	205,73		
<b>Total</b>	79	50519,5			

**Gráfica 49.** Diagrama de Pareto y gráfica normal de efector estandarizados para el *makespan* de la instancia i9j3k5-1.



**Gráfica 50.** Diagrama de residuos para el *makespan* de la instancia i9j3k5-1.



- Análisis de la varianza para la instancia i9j3k5-2.

Tabla 95. ANOVA para la instancia i9j3k5-2.

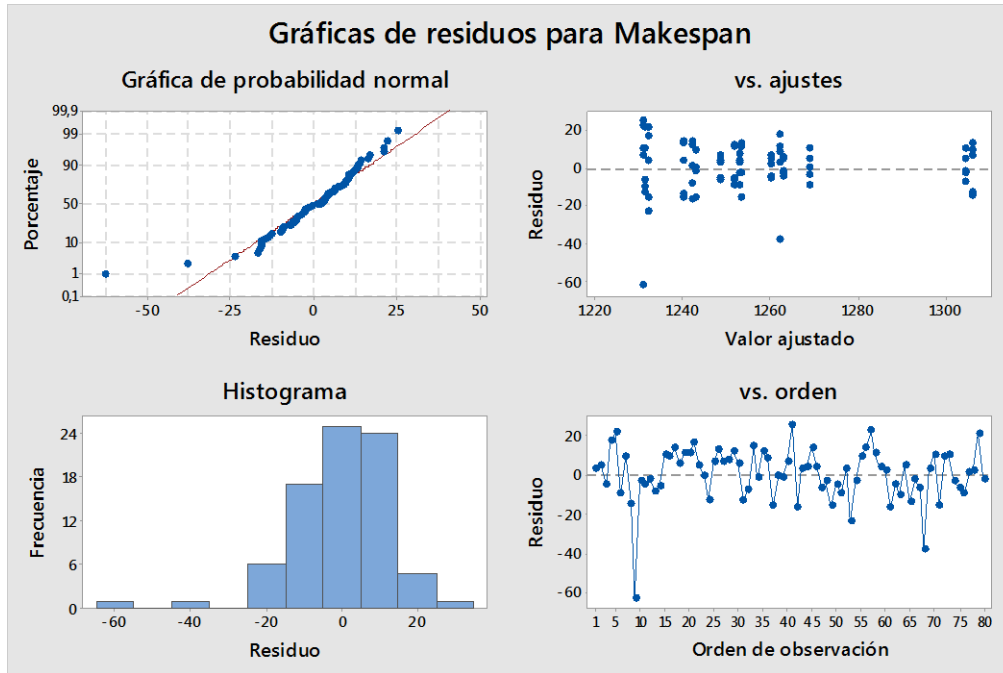
Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Modelo	15	38269,3	2551,3	11,61	0
Lineal	5	26709,2	5341,8	24,31	0
Número de Países	1	19750,6	19750,6	89,88	<b>0</b>
% Imperialistas	1	3740,1	3740,1	17,02	<b>0</b>
% Chi	1	2279,1	2279,1	10,37	<b>0,002</b>
% Cruce	1	556,5	556,5	2,53	0,116
% Revolución	1	382,8	382,8	1,74	0,192
Interacciones de 2 términos	10	11560,1	1156	5,26	0
Número de Países*% Imperialistas	1	825,6	825,6	3,76	0,057
Número de Países*% Chi	1	5395,6	5395,6	24,55	<b>0</b>
Número de Países*% Cruce	1	74,1	74,1	0,34	0,563
Número de Países*% Revolución	1	132,6	132,6	0,6	0,44
% Imperialistas*% Chi	1	891,1	891,1	4,06	<b>0,048</b>
% Imperialistas*% Cruce	1	23,1	23,1	0,11	0,747
% Imperialistas*% Revolución	1	1	1	0	0,946
% Chi*% Cruce	1	117,6	117,6	0,54	0,467
% Chi*% Revolución	1	165,3	165,3	0,75	0,389
% Cruce*% Revolución	1	3934	3934	17,9	<b>0</b>
Error	64	14063,2	219,7		
Total	79	52332,5			

Para la instancia i9j3k5-2 observando los datos presentados en la tabla 95 del ANOVA y en la gráfica 52, presentada a continuación, del diagrama de Pareto y la gráfica normal de efectos estandarizados se concluye que los factores número de países, % de imperialistas, la interacción entre número de países y %  $\xi$ , la interacción entre % de imperialistas y %  $\xi$  y, por último, la interacción entre % de cruce y % de revolución influyen en la variable respuesta *makespan*. Mientras tanto, el factor menos influyente la interacción entre % de imperialistas y % de revolución.

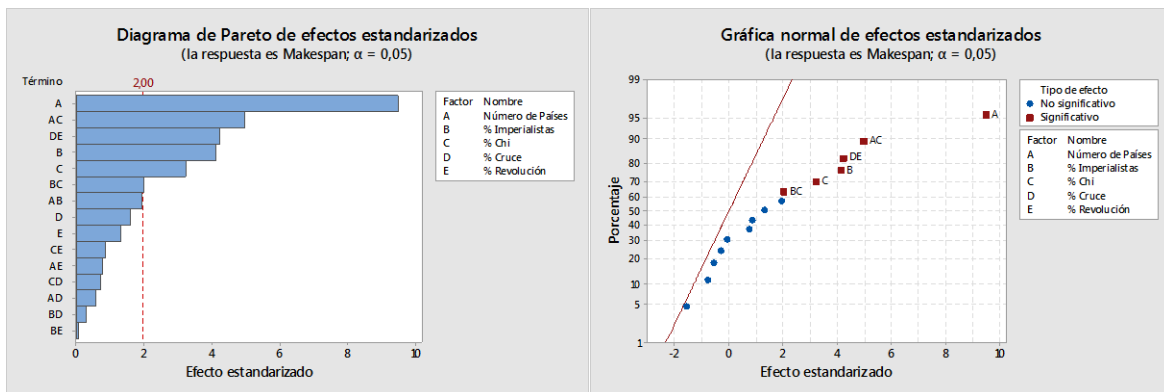
El factor más influyente entre los ya mencionados es el número de países, seguido la interacción entre número de países y %  $\xi$ , en tercer lugar, la interacción entre % de cruce y % de revolución, en cuarto lugar, % de imperialistas, seguidamente, el %  $\xi$  y finalmente, la interacción entre % de imperialistas y %  $\xi$ . Por último, con

base en la gráfica 51 presentada a continuación se confirman los supuestos de normalidad e independencia de los datos.

**Gráfica 51.** Diagrama de residuos para el *makespan* de la instancia i9j3k5-2.



**Gráfica 52.** Diagrama de Pareto y gráfica normal de efectores estandarizados para el *makespan* de la instancia i7j3k5-2.



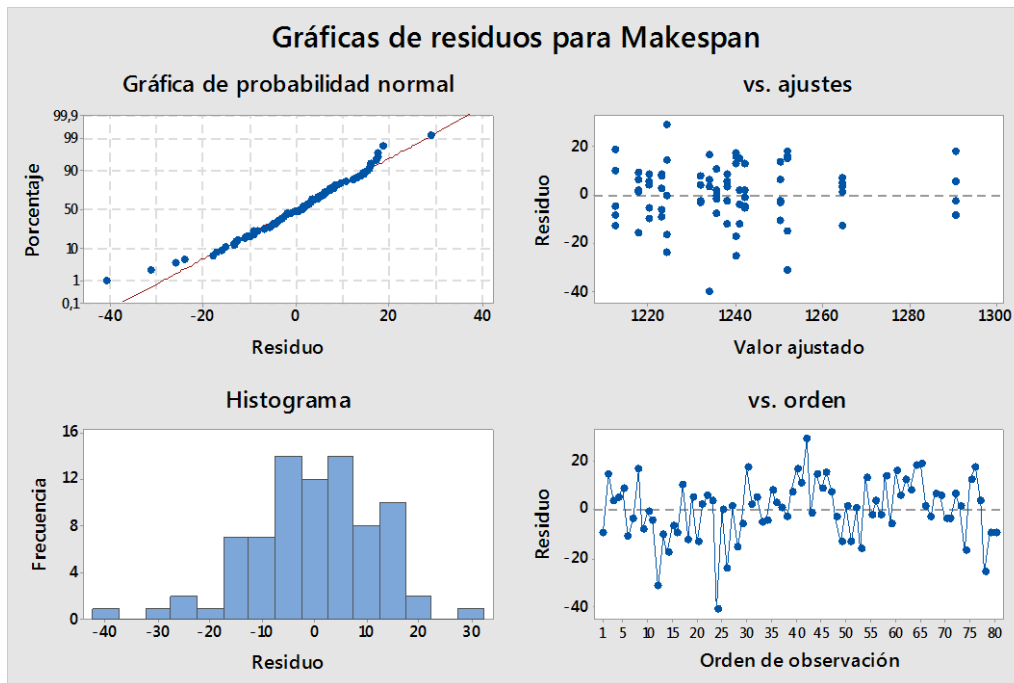
- Análisis de la varianza para la instancia i9j3k5-3.

Tabla 96. ANOVA para la instancia i9j3k5-3.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Modelo	15	28484,4	1898,96	10,38	0
Lineal	5	16744	3348,81	18,3	0
Número de Países	1	9636	9636,05	52,67	<b>0</b>
% Imperialistas	1	5712,2	5712,2	31,22	<b>0</b>
% Chi	1	1,8	1,8	0,01	0,921
% Cruce	1	441,8	441,8	2,41	0,125
% Revolución	1	952,2	952,2	5,2	<b>0,026</b>
Interacciones de 2 términos	10	11740,3	1174,03	6,42	0
Número de Países*% Imperialistas	1	405	405	2,21	0,142
Número de Países*% Chi	1	1513,8	1513,8	8,27	<b>0,005</b>
Número de Países*% Cruce	1	5	5	0,03	0,869
Número de Países*% Revolución	1	1584,2	1584,2	8,66	<b>0,005</b>
% Imperialistas*% Chi	1	312,1	312,05	1,71	0,196
% Imperialistas*% Cruce	1	490	490,05	2,68	0,107
% Imperialistas*% Revolución	1	3726,5	3726,45	20,37	<b>0</b>
% Chi*% Cruce	1	551,2	551,25	3,01	0,087
% Chi*% Revolución	1	3150,1	3150,05	17,22	<b>0</b>
% Cruce*% Revolución	1	2,4	2,45	0,01	0,908
Error	64	11709,6	182,96		
Total	79	40194			

Para el caso de la instancia i9j3k5-3 se concluye con base en la tabla 96 de ANOVA y la gráfica 54 del diagrama de Pareto y la gráfica normal de efectos estandarizados mostrada más adelante, presentada a continuación, que influyen la variable respuesta *makespan* en orden descendente de influjo los factores, número de países, el % de imperialistas, la interacción entre el % de imperialistas y % de revolución, la interacción entre el %  $\xi$  y % de revolución, la interacción entre el número de países y el % de revolución, la interacción entre número de países y %  $\xi$  y por último, el % de revolución. El factor menos influyente es el %  $\xi$ ; también se confirma con base en la gráfica 53 de más adelante los supuestos de normalidad e independencia de los datos.

**Gráfica 53.** Diagrama de residuos para el *makespan* de la instancia i9j3k5-3.



**Gráfica 54.** Diagrama de Pareto y gráfica normal de efector estandarizados para el *makespan* de la instancia i9j3k5-3.

