

UN ALGORITMO HÍBRIDO PARA EL PROBLEMA DE RUTEO DE VEHÍCULOS  
CON TIEMPOS DE VIAJE ESTOCÁSTICOS Y VENTANAS DE TIEMPO  
SUAVE

DIANA MARCELA OVIEDO REYES  
KAROL MELISSA VALDIVIESO GÓMEZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO MECÁNICAS  
ESCUELA DE ESTUDIOS INDUSTRIALES Y EMPRESARIALES  
BUCARAMANGA  
2016

UN ALGORITMO HÍBRIDO PARA EL PROBLEMA DE RUTEO DE VEHÍCULOS  
CON TIEMPOS DE VIAJE ESTOCÁSTICOS Y VENTANAS DE TIEMPO  
SUAVE

DIANA MARCELA OVIEDO REYES  
KAROL MELISSA VALDIVIESO GÓMEZ

Trabajo de grado para optar al título de:  
Ingeniero Industrial

Director:  
HENRY LAMOS DIAZ  
PhD en Física- Matemática

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO MECÁNICAS  
ESCUELA DE ESTUDIOS INDUSTRIALES Y EMPRESARIALES  
BUCARAMANGA  
2016

## **AGRADECIMIENTOS**

*A Sebastián Díaz por poner a nuestra disposición todas sus capacidades y conocimientos, demostrando un fuerte compromiso con este proyecto.*

*A nuestros familiares y amigos quienes nos animaron a continuar adelante durante la investigación.*

## CONTENIDO

	Pág.
INTRODUCCIÓN .....	15
1. PLANTEAMIENTO DEL PROBLEMA.....	18
2. JUSTIFICACIÓN .....	20
3. OBJETIVOS .....	22
3.1 OBJETIVO GENERAL .....	22
3.2 OBJETIVOS ESPECÍFICOS.....	22
4. REVISIÓN DE LA LITERATURA .....	23
5. MARCO DE ANTECEDENTES.....	34
6. MARCO TEÓRICO.....	40
6.1 COMPLEJIDAD COMPUTACIONAL .....	40
6.2 OPTIMIZACIÓN COMBINATORIA .....	42
6.3 HEURÍSTICA .....	46
6.4 METAHEURÍSTICA .....	47
6.4.1 Metaheurísticas basadas en la trayectoria .....	50
6.4.2 Metaheurísticas basadas en población.....	52
6.5 OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS .....	54
6.5.1 Algoritmo básico de PSO .....	56
6.5.2 Parámetros .....	57
6.5.3 PSO con coeficientes de constricción.....	58
6.5.4 Topología Poblacional .....	59
6.6 GRASP.....	61
6.7 MULTIPLE PHASE NEIGHBORHOOD SEARCH - GRASP .....	62
6.7.1 Pasos del Algoritmo MPNS-GRASP.....	63
6.8 ENS (Expanding Neighborhood Search) .....	64
6.9 PATH RELINKING .....	68
6.10 VEHICLE ROUTING PROBLEM (VRP).....	70
6.10.1 Variantes básicas del VRP .....	72
6.10.2 Otras variantes del VRP.....	73
6.10.3 Stochastic Vehicle Routing Problem.....	75

6.10.4 VRP with Stochastic Travel Times and Soft Time Windows.....	76
7. INSTANCIAS.....	81
8. AJUSTE DE PARÁMETROS DEL ALGORITMO .....	84
8.1 Parámetros del MPNS-GRASP .....	84
8.2 Parámetros del ENS .....	84
8.3 Parámetros del PSO .....	85
9. LÍMITES DEL ESPACIO DE BÚSQUEDA.....	88
10. REPRESENTACIÓN DE LA SOLUCIÓN .....	90
10.1 MÉTODO DE CODIFICACIÓN Y DECODIFICACIÓN .....	91
11. ETAPAS DEL ALGORITMO PARA EL VRPSTTW .....	94
12. APLICACIÓN DE LA METAHEURÍSTICA PROPUESTA A UN PROBLEMA DEL TSP .....	105
13. CÁLCULO DE LA FUNCIÓN OBJETIVO .....	118
14. DEFINICIÓN DE LOS COSTOS PARA LA FUNCIÓN OBJETIVO .....	122
15. DEFINICIÓN DEL NÚMERO DE ESCENARIOS.....	123
16. RESULTADOS COMPUTACIONALES .....	127
17. DEFINICIÓN DEL NÚMERO DE ITERACIONES.....	131
17.1 Análisis de la varianza para el número de iteraciones .....	132
18. CONCLUSIONES.....	143
19. RECOMENDACIONES .....	145
BIBLIOGRAFÍA .....	146

## LISTA DE FIGURAS

	Pág.
Figura 1. Problemas P, NP, NP-hard, NP-complete.....	41
Figura 2. Clasificación de las metaheurísticas.....	49
Figura 3. Algunas Topologías del PSO.....	59
Figura 4. Estrategia de Movimientos de búsqueda local en círculos restringidos (CRLSM).....	65
Figura 5. Estrategia de Ampliación del Barrio de búsqueda.....	67
Figura 6. Distribución de los clientes en el área geográfica para las instancias de Solomon (1987) .....	82
Figura 7. Ruta inicial .....	91
Figura 8. Representación de la solución.....	93
Figura 9. Creación de la lista RCL.....	96
Figura 10. Aplicación de la estrategia CRLSM.....	99
Figura 11. Actualización de la posición y la velocidad en PSO.....	102
Figura 12. Matriz de distancias TSP .....	105
Figura 13. Grafo del problema del TSP.....	106
Figura 14. Duplas seleccionadas.....	109
Figura 15. Escenarios para el cálculo de la función objetivo.....	120
Figura 16. Gráfica Distancia Vs Número de escenarios.....	124
Figura 17. Resultados Distancia Vs Número de Iteraciones para la instancia R101.....	132

## LISTA DE TABLAS

	Pág.
Tabla 1. Pseudocódigo PSO.....	56
Tabla 2. Parámetros PSO, configuraciones típicas.....	86
Tabla 3. Codificación de la ruta.....	92
Tabla 4. Decodificación de la ruta.....	92
Tabla 5. Parámetros iniciales para el MPNS- GRASP.....	109
Tabla 6. Lista Ordenada de todos los arcos que componen el problema.....	109
Tabla 7. Soluciones Iniciales.....	110
Tabla 8. Arcos que componen la primera ruta con sus respectivas distancias.....	111
Tabla 9. Soluciones mejoradas por ENS.....	112
Tabla 10. Parámetros iniciales del modelo PSO.....	113
Tabla 11. Codificación de las partículas.....	113
Tabla 12. Velocidad y posición de la partícula 2 en la primera iteración.....	115
Tabla 13. Decodificación del vector posición $X_i$ .....	115
Tabla 14. Solución objetivo, inicial y nueva solución encontrada con la estrategia Path Relinking.....	117
Tabla 15. Definición de los Costos del modelo.....	122
Tabla 16. Resultados del costo y tiempo computacional para 50 escenarios.....	124
Tabla 17. Resultados del costo y tiempo computacional para 100 .....	125
Tabla 18. Resultados Computacionales del algoritmo propuesto.....	129
Tabla 19. Orden aleatorio de las corridas.....	133
Tabla 20. Resultados Instancia C101.....	135
Tabla 21. Análisis de Varianza Instancia C101.....	135
Tabla 22. Resultados Instancia R101.....	135
Tabla 23. Análisis de Varianza Instancia R101.....	136
Tabla 24. Resultados Instancia RC101.....	136
Tabla 25. Análisis de Varianza Instancia RC101.....	136
Tabla 26. Resultados Instancia C201.....	137
Tabla 27. Análisis de Varianza Instancia C201.....	137

Tabla 28. Resultados Instancia R201.....	137
Tabla 29. Análisis de Varianza Instancia R201.....	138
Tabla 30. Resultados Instancia RC201.....	138
Tabla 31. Análisis de Varianza Instancia RC201.....	138
Tabla 32. Prueba Tukey Instancia C101.....	139
Tabla 33. Prueba Tukey Instancia R101.....	140
Tabla 34. Prueba Tukey Instancia RC101.....	141
Tabla 35. Prueba Tukey Instancia R201.....	141
Tabla 36. Prueba Tukey Instancia RC201.....	142

## LISTA DE ANEXOS

ANEXO A. Código de Programación

ANEXO B. Instancias

ANEXO C. Artículo Académico

**NOTA: Estos anexos pueden ser consultados en la Base de Datos de la Biblioteca de la Universidad Industrial de Santander.**

## RESUMEN

**TÍTULO:** “UN ALGORITMO HÍBRIDO PARA EL PROBLEMA DE RUTEO DE VEHÍCULOS CON TIEMPOS DE VIAJE ESTOCÁSTICOS Y VENTANAS DE TIEMPO SUAVE”.\*

**AUTORES:**

OVIEDO REYES, Diana Marcela.

VALDIVIESO GÓMEZ, Karol Melissa.\*\*

**PALABRAS CLAVE:**

Problema de Ruteo de vehículos, Tiempos de viaje estocásticos, Optimización por enjambre de partículas, MPNS-GRASP, ENS, Path Relinking.

**DESCRIPCIÓN:**

En esta investigación se implementó un algoritmo híbrido de PSO para resolver el problema de ruteo de vehículos con tiempos de viaje estocásticos y ventanas de tiempo suave. El híbrido se formó a partir de cuatro metaheurísticas que enfocan sus esfuerzos en diferentes puntos críticos de la optimización para entregar en conjunto soluciones de alta calidad. MPNS-GRASP es la primera metaheurística, la cual hace uso de procedimientos voraces y aleatorios para generar soluciones iniciales; estas soluciones pasan a ser mejoradas en una segunda etapa a través de la metaheurística Expanding Neighborhood Search quien pretende eliminar arcos de gran longitud mediante el uso del intercambio 2-opt enmarcado en la estrategia de búsqueda local por círculos restringidos. Finalmente, Path Relinking, actúa como agente de post optimización, encontrando las mejores soluciones a través de la exploración de caminos entre soluciones elites. La optimización por enjambre de partículas (PSO) actúa como algoritmo central y se encarga de la convergencia del problema a una respuesta cercana a la óptima.

El algoritmo es probado en 18 problemas de las instancias de Solomon, reconocidas en la literatura como banco de prueba para el VRP con ventanas de tiempo. Los resultados obtenidos presentan variaciones con respecto a las mejores soluciones conocidas en la literatura que oscilan entre el 0% (alcanzando la mejor solución conocida en la literatura) y un 23%.

---

\* Proyecto de grado.

\*\* Facultad de Ingenierías Físico Mecánicas. Escuela de Estudios Industriales y Empresariales. Programa de Ingeniería Industrial. Director: PhD Henry Lamos Díaz.

## ABSTRACT

**TITLE:** “A HYBRID ALGORITHM FOR THE VEHICLE ROUTING PROBLEM WITH STOCHASTIC TRAVEL TIMES AND SOFT TIME WINDOWS”\*

**AUTHORS:**

OVIEDO REYES, Diana Marcela

VALDIVIESO GÓMEZ, Karol Melissa\*\*

**KEYWORDS:** Vehicle Routing Problem, Stochastic Travel Times, Particle Swarm Optimization, MPNS-GRASP, ENS, Path Relinking.

**DESCRIPTION:**

In this work, a hybrid *particle swarm optimization* (PSO) algorithm was implemented to solve the vehicle routing problem with stochastic travel times and soft time windows. The algorithm was formed based on four metaheuristic techniques that focus their solving effort on different critical points of the optimization. Firstly, Multiple Phase Neighborhood Search - Greedy Randomized Adaptive Search Procedure (MPNS-GRASP) arrange an initial set of solutions by using random and greedy procedures. Secondly, Expanding Neighborhood is based on a method called circle restricted local search moves (CRLSM), this strategy restricts the search into circles around the candidate for deletion edges. Finally, Path Relinking acts as a post optimization agent and tries to find the best solutions by means of path exploring inside the elite group of solutions. PSO is the core of the algorithm and is responsible for the convergence of the problem.

This study tests the algorithm in 18 problems in the well-known data sets given by Solomon, which are recognized as a test bench for the Vehicle Routing Problem (VRP) with time windows. The final results are in the range of 0% (reaching the best solution given by the literature) and 23%.

---

\* Graduation Project.

\*\* Faculty of Engineering physicomechanical. School of Industrial and Business Studies. Directed by Henry Lamos Diaz PhD.

## INTRODUCCIÓN

El entorno global en el que se desenvuelven las empresas actualmente, caracterizado por la amplia competencia, da lugar al aumento de las exigencias del cliente en la entrega oportuna de los bienes, razón por la cual la gestión de la cadena de abastecimiento se convierte en un punto clave para la mejora del desempeño empresarial.

“La acelerada apertura comercial del país a raíz de la suscripción de varios tratados comerciales, convierte a la logística en una variable fundamental para competir en el mercado. Por lo tanto, para que Colombia pueda lograr una real transformación productiva y así cumplir con sus metas de competitividad para el año 2032, es necesario desarrollar las capacidades logísticas del país, posibilitadas por una infraestructura básica apropiada y unos servicios competitivos de transporte y almacenamiento de cargas”<sup>1</sup>.

La poca competitividad de los productos nacionales en el mercado internacional se debe entre otras, a la elevada participación de los costos de transporte en la operación logística, los cuales se ven reflejados en el precio final del producto. Este hecho evidencia la importancia de enfocar las acciones de mejora al fortalecimiento de procedimientos inherentes al movimiento de mercancías <sup>2</sup>.

---

<sup>1</sup> CONSEJO PRIVADO DE COMPETITIVIDAD. Infraestructura, transporte y logística [en línea] <<http://www.compitem.com.co/site/wp-content/uploads/2012/10/6-Infraestructura-Transporte-y-Logistica.pdf> > [citado en 10 de agosto de 2015].

<sup>2</sup> UNIVERSIDAD DEL ROSARIO. Elevados costos de transporte [en línea]. <<http://www.urosario.edu.co/Universidad-Ciencia-Desarrollo/ur/Fasciculos-Anteriores/Tomo-II-2007/Fasciculo-15/ur/Elevados-costos-de-transporte/> > [citado en 10 de Agosto de 2015].

Esta situación conduce a la preocupación de las empresas por diseñar rutas efectivas de distribución, que satisfagan los requerimientos de los clientes y aumenten la calidad del servicio. Es así, como el problema de ruteo de vehículos se vuelve protagonista en la logística de distribución.

El Problema de Ruteo de Vehículos (VRP) es el nombre genérico dado al conjunto de problemas que envuelven la visita de clientes a través de flotas de transporte <sup>3</sup>. En la formulación clásica del VRP, los elementos que componen el problema son determinísticos, sin embargo, para que la adopción de modelos teóricos muestre resultados más cercanos a la realidad, se hace necesario que los problemas consideren en su formulación la incertidumbre generada por factores propios de esta.

En un núcleo urbano, se hace evidente que el tiempo necesario para viajar entre dos puntos no depende únicamente de la distancia, sino que las fluctuaciones de tráfico afectan considerablemente la velocidad de desplazamiento, hecho que se traduce en grandes variaciones en los tiempos de viaje <sup>4</sup>. Muchas causas se le atribuyen a este fenómeno; algunas son completamente aleatorias, como los accidentes de tráfico o las condiciones meteorológicas; otras están relacionadas con la dirección y la hora del día en que se realiza el viaje, como las horas pico que producen embotellamientos diariamente. Con base en lo anterior, surge el estudio del problema de ruteo de vehículos con tiempos de viaje estocásticos.

---

<sup>3</sup> CHRISTOFIDES, Nicos. The Vehicle Routing Problem. R.A.I.R.O. En: Recherche Opérationnelle. Vol.10, No. 5 (1976). p.55-70.

<sup>4</sup> BALSEIRO, S.R., LOISEAU, I., RAMONET, J. An Ant Colony algorithm hybridized with insertion heuristics for the Time Dependent Vehicle Routing Problem with Time Windows. En: Computers & Operations Research [Base de datos en línea]. Vol. 38, no.6 (junio 2011); p. 954-966 [citado en 1 agosto de 2015]. Disponible en ScienceDirect Research Databases.

El presente trabajo se centra en la investigación de este problema y le añade una restricción de ventana de tiempo suave como forma de considerar el nivel de servicio en el mismo. Para ello se propone un algoritmo híbrido de optimización por enjambre de partículas mejorado mediante un algoritmo de inicialización (MPNS-GRASP), un algoritmo de ampliación del vecindario de búsqueda (ENS) y un algoritmo de post-optimización, que combina elementos entre las mejores soluciones para generar soluciones más cercanas a la óptima.

## 1. PLANTEAMIENTO DEL PROBLEMA

La solución al problema de ruteo de vehículos (VRP) consiste en determinar un conjunto de rutas que comienzan y terminan en un depósito, cada una realizada por un solo vehículo, de manera que se minimice el costo de transporte global y se cumpla con las exigencias de los clientes y las restricciones operativas.<sup>5</sup>

El problema de ruteo de vehículos ha evolucionado enormemente desde su introducción en los años cincuenta, dejando de lado los parámetros determinísticos para enfocarse en factores estocásticos propios de la realidad. De esta manera, se introduce la incertidumbre en el nivel de planificación para así lograr rutas más eficaces y por ende aumentar la calidad del servicio<sup>6</sup>. Parámetros comunes considerados en estos problemas son demandas estocásticas, clientes estocásticos y tiempos de viaje estocásticos.

El presente trabajo se centra en el problema de ruteo de vehículos con tiempos de viaje estocásticos, que obedecen una distribución de probabilidad conocida y considera la disponibilidad del cliente, definida a través de una ventana de tiempo suave (establecida por el mismo), que penaliza el costo en caso de violación (entregas tempranas o tardías).

El supuesto de tiempos de viaje estocásticos se da por la necesidad de incorporar en la formulación del problema, aspectos de congestión vehicular, condiciones

---

<sup>5</sup> MOGHADDAM, Babak Farhang; RUIZ, Rubén y SADJADI, Seyed Jafar. Vehicle routing problem with uncertain demands: An advanced particle swarm algorithm. En: Computers & Industrial Engineering [base de datos en línea]. Vol. 62, no. 1 (Feb. 2012); p. 306-317, ISSN 0360-8352. [citado en 23 de agosto de 2015] Disponible en ScienceDirect Research Databases.

<sup>6</sup> TAŞ, Op.cit., p. 789-799.

climáticas, capacidades del conductor, entre otros, que causan que no se conozca el tiempo exacto que demorará un vehículo en cubrir su ruta.

La complejidad de este problema, clasificado como NP-hard, no permite obtener soluciones óptimas a partir de métodos exactos en tiempos computacionales razonables, por lo que se hace uso de métodos aproximados para encontrar soluciones cercanas a la óptima. El enfoque presentado para dar solución a este problema es un algoritmo de enjambre de partículas (PSO) mejorado, mediante la combinación de tres metaheurísticas (MPNS-GRASP, ENS y PR) que reúnen algoritmos de inicialización para encontrar soluciones iniciales más eficientes y uno de post-optimización que explora la trayectoria entre soluciones factibles, combinando elementos entre ellas.

La eficiencia del algoritmo planteado se evalúa en las instancias del benchmarking propuestas por Solomon, debido a su amplia aceptación y uso en la literatura del VRP.

## 2. JUSTIFICACIÓN

Los modelos de negocio orientados a cumplir con las expectativas del cliente en la entrega efectiva de bienes, han llevado al desarrollo de nuevos modelos que tengan en cuenta el abastecimiento de los bienes en el momento y lugar que el cliente necesita con la calidad requerida.

Uno de los modelos que ha sido y es objeto de estudio actualmente por la comunidad académica es el problema de ruteo de vehículos (VRP). El VRP fue inicialmente estudiado por Dantzig y Ramser (1959) y debido a su importancia en la distribución de bienes se han desarrollado múltiples variantes. Algunas de estas incluyen VRP con ventanas de tiempo (VRPTW) (Fisher, 1997), VRP heterogéneo (HVRP) (Golden, 1984), VRP abierto (OVRP) (Li y Tian, 2006 y Li, 2009), VRP con viajes de regreso (VRPB) (Deif y Bodin, 1984) y el problema de ruteo de vehículos multi-periodo (Zapfel y Bögl, 2008). El VRP es un Problema de Optimización Combinatoria de clase NP-hard.

En las últimas décadas, con el fin de considerar el componente estocástico y adaptar el problema a condiciones más reales, como la incertidumbre de la demanda, el tiempo de viaje, la aparición de nuevos clientes, etc., la investigación se ha dirigido al estudio de estos problemas, que presentan mayor dificultad que los problemas determinísticos.

El diseño de rutas, considerando un horario determinista y estático es un supuesto fuerte para un entorno de la vida real, ya que da lugar a operaciones ineficientes en la programación de viajes. Para generar modelos aplicables al problema de ruteo de vehículos que se asemejen a entornos más reales, es necesario considerar la disponibilidad del cliente y el hecho de que las rutas de vehículos son operadas en una red de tráfico que tiene diferentes niveles de congestión y condiciones

meteorológicas, en función de la hora del día, los cuales generan incertidumbre en los tiempos de viaje entre clientes. Esta incertidumbre puede causar que las entregas de los bienes se retrasen o adelanten, lo que implica costos de penalización debido a los acuerdos previos establecidos con los clientes. Por tanto, el presente trabajo se enfoca en la investigación del problema de ruteo de vehículos con tiempos de viaje estocásticos y ventanas de tiempo suave.

El problema descrito anteriormente se centra en considerar los tiempos de viaje estocásticos y ventanas de tiempo suaves en el nivel de planificación, para evaluar con precisión las llegadas de vehículos a las instalaciones del cliente por medio de un algoritmo de enjambre de partículas (PSO) mejorado, mediante la combinación de tres metaheurísticas (MPNS-GRASP, ENS y PR) que reúnen algoritmos de inicialización para encontrar soluciones iniciales más eficientes y uno de post optimización que explora la trayectoria entre soluciones factibles combinando elementos entre ellas, alcanzando así soluciones aproximadas a la solución óptima.

### **3. OBJETIVOS**

#### **3.1 OBJETIVO GENERAL**

Construir un algoritmo híbrido de metaheurísticas para la solución del problema de ruteo de vehículos con tiempos de viaje estocásticos y ventanas de tiempo suave.

#### **3.2 OBJETIVOS ESPECÍFICOS**

- ∴ Desarrollar un enfoque que permita combinar las metaheurísticas Optimización por Enjambre de Partículas (PSO), Multiple Phase Neighborhood Search–Greedy Randomized Adaptive Search Procedure (MPNS-GRASP), Expanding Neighborhood Search (ENS) y Path Relinking (PR) para la solución del problema de ruteo de vehículos con tiempos de viaje estocásticos y ventanas de tiempo suaves.
  
- ∴ Implementar el algoritmo propuesto en el software Matlab.
  
- ∴ Evaluar la eficiencia del algoritmo planteado a partir de las instancias del benchmarking propuestas por Solomon.
  
- ∴ Elaborar un artículo académico con los análisis, resultados y conclusiones del tema estudiado.

#### 4. REVISIÓN DE LA LITERATURA

El primer registro en la literatura sobre el VRP data de 1959 con el estudio abordado por **Dantzig y Ramser**<sup>7</sup>, que asigna estaciones de servicio a una flota de camiones (proveedores de gasolina), de tal forma que la demanda de cada estación sea satisfecha y el kilometraje total cubierto por la flota se minimice; por lo cual formulan un modelo de programación lineal para la obtención de una solución cercana a la óptima.

Posteriormente, en 1964, **Clarke y Wright** mejoran el enfoque propuesto por Dantzig y Ramser, al buscar la ruta óptima para una flota de camiones de diferentes capacidades desde un depósito central hasta un número de puntos de entrega. El modelo de solución desarrollado para el problema se basa en un procedimiento iterativo que permite la rápida selección de una ruta óptima o casi óptima <sup>8</sup>.

Tres años más tarde, **Pullen y Webb** en 1967, reconocen la pertinencia de la inclusión del enfoque de ventanas de tiempo en el VRP tradicional, como forma de agregarle al mismo, restricciones de la vida real <sup>9</sup>.

Luego en 1983, **Solomon** aplica algunas heurísticas como el método de ahorro, el vecino más cercano, así como heurísticas de inserción y de barrido, para resolver

---

<sup>7</sup> DANTZIG, G. y RAMSER, J. The Truck Dispatching Problem. En: Management Science. Vol. 6, no. 1 (Oct., 1959); p. 80-91.

<sup>8</sup> CLARKE, G y WRIGHT, J. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points, Citado por LAPORTE, Gilbert. The vehicle routing problem. An overview of exact and approximate algorithms. En: European Journal of Operational Research. Vol 59, no 3 (Jun., 1992); p. 345-358, ISSN 0377-2217.

<sup>9</sup> PULLEN, H y WEBB, M. A Computer Application to a Transport Scheduling Problem, Citado por, ROCHA, L.; GONZÁLEZ, C. y ORJUELA, J. Una revisión al estado del arte del problema de ruteo de vehículos: Evolución histórica y métodos de solución. En: Ingeniería. Vol.16, no. 2 (2011); p. 35-55.

el problema del VRPTW, de las cuales el método de inserción arroja resultados muy buenos y consistentes <sup>10</sup>.

Posteriormente en 1985, **Savelsbergh** plantea un algoritmo de búsqueda local para el problema de ruteo de vehículos con ventanas de tiempo fundamentado en el concepto de k-intercambio, y considera en la formulación del problema encontrar las soluciones iniciales <sup>11</sup>.

Desde ese entonces, se han estudiado distintas metaheurísticas para la solución del VRPTW, tales como búsqueda tabú (E. Taillard, P. Badeau, M. Gendreau, F. Guertin, J.-Y. Potvin) y (J.-F. Cordeau, G. Laporte, A. Mercier), recocido simulado (W.C. Chiang, R.A. Russell) y (Czech ZJ, Czarnas P.), algoritmos genéticos (Thangiah SR, Nygard KE, Juell PL.) y (J. Berger, M. Barkaoui), estrategias de evolución (J. Homberger, H. Gehring) y Sistemas de Colonia de Hormigas (L.M. Gambardella, R. Taillard, G. Agazzi) <sup>12</sup>.

**Dengying Jiang, Wenxia Jiang y Zhangcan Huang**<sup>13</sup>, en el año 2008 proponen una estrategia de optimización al problema de ruteo de vehículos con ventanas de tiempo en dos etapas: la primera, se encamina a garantizar la diversidad de la solución inicial, para lo que se utiliza el PFIH (Push Forward Insertion Heuristic) y la segunda, propone un sistema híbrido a partir de la combinación de una Búsqueda

---

<sup>10</sup> SOLOMON, M. Algorithms for the vehicle routing and scheduling problems with time window Constraints, Citado por CHENG, Chi-Bin, WANG, Keng-Pin. Solving a vehicle routing problem with time windows by a decomposition technique and a genetic algorithm. En: Expert Systems with Applications [base de datos en línea]. Vol. 36, no. 4 (May., 2009); p. 7758-7763, ISSN 0957-4174. [citado en 5 de agosto de 2015] Disponible en ScienceDirect Research Databases.

<sup>11</sup> SAVELBERGH, M.W.P. Local Search in Routing Problems with Time Windows. En: Annals of Operations Research [base de datos en línea]. Vol. 4, No. 1 (jun.,1985); p. 285-305 [citado en 10 de agosto de 2015] Disponible en SpringerLink Research DataBases.

<sup>12</sup> BALSEIRO, Op.cit., p.954-966.

<sup>13</sup> JIANG, Dengying.; JIAN, Wenxia. y HUANG, Zhangcan. A Hybrid Algorithm for Vehicle Routing Problem with Time Windows. En: Springer Berlin Heidelberg [base de datos en línea] Vol. 5370 (dic. 2008); p. 198-205. [Citado en 1 de agosto de 2015] Disponible en SpringerLink Research Databases.

extendida en el vecindario (LNS) y Recocido Simulado (SA), con el fin de ampliar el espacio de búsqueda y por ende evitar la captura de un óptimo local.

Posteriormente, en el año 2009, **Chi-Bin Cheng y Keng-Pin Wang**<sup>14</sup> encuentran una solución al VRPTW basados en una técnica de descomposición, la cual pretende dividir el problema inicial en un problema principal y un conjunto de sub problemas del agente viajero, con restricciones de ventana de tiempo. Esta descomposición no sólo reduce el tamaño del problema, sino que permite el uso de procedimientos de solución más simples. El problema principal es resuelto a partir de un algoritmo genético y cada sub problema mediante un algoritmo heurístico sencillo. La solución del problema original se obtiene a través de repetidas interacciones entre el problema principal y el conjunto de sub problemas. El efecto del método de descomposición se compara con la solución del problema original (sin el uso de este enfoque), llegando a la conclusión que esa técnica proporciona una mejor solución.

En el 2012, **Jin Ling**<sup>15</sup> realiza un estudio sobre el VRP con ventanas de tiempo, que a diferencia de los problemas comunes del VRPTW no aborda únicamente el objetivo de minimizar los tiempos viaje o distancias recorridas, si no que propone disminuir el consumo total de combustible, factor relevante para las economías que desean llevar a cabo su operación logística de manera amigable con el medio ambiente. Para solucionar este problema propone un algoritmo basado en búsqueda tabú con un procedimiento descendente de vecindario de variables aleatorias (RVND), que utiliza una ruta paralela adaptiva de construcción heurística, seis métodos de búsqueda en el vecindario y mecanismos de orden aleatorio. Los

---

<sup>14</sup> CHENG, Op.cit., p. 7758-7763.

<sup>15</sup> LI, Jin. Vehicle routing problem with time windows for reducing fuel consumption. En: Journal of Computers. Vol. 7, No. 12 (Dic.,2012); p. 3020-3027.

resultados muestran que la solución de reducir al mínimo el consumo total de combustible no se logra con la solución tradicional del VRPTW.

Simultáneamente al estudio de variantes del VRP determinista (como el VRPTW), otros autores como **Stewart y Golden**, tratan de acercar más la formulación de los modelos a las situaciones de la vida real, introduciendo al VRP componentes aleatorios, dando lugar al Problema de Ruteo de vehículos con variables estocásticas (SVRP) <sup>16</sup>. Este problema presenta tres variantes, la primera cuando la demanda de cada consumidor es una variable aleatoria (VRPSD) por **Laporte y Loveaux** en 1987; la segunda cuando los tiempos de servicio y tiempos de viaje son variables aleatorias (VRPSTT) por **Laporte, Loveaux y Mercure** en 1992 y finalmente cuando la disponibilidad del consumidor es una variable aleatoria (SVRP-SN) por **Jaillet** en 1985. <sup>17</sup>

Años más tarde, **Van Landeghem** en 1988 y **Koskosidis, Powell y Solomon** en 1992, introducen los términos de ventanas de tiempo duras y ventanas de tiempo suaves, respectivamente; debido a que en la práctica el grado de la restricción de las ventanas de tiempo puede variar, es decir en el primer caso, la violación de las ventanas de tiempo genera rechazo a los pedidos y en el segundo, la violación no genera el rechazo, pero sí la penalización en el costo.<sup>18</sup>

---

<sup>16</sup> STEWART, W y GOLDEN, B. Stochastic vehicle routing: a comprehensive approach, Citado por TAŞ, Duygu; DELLAERT, Nico; WOENSEL, Tom van y DE KOK, Ton. Vehicle routing problem with stochastic travel times including soft time windows and service costs. En: Computers & Operations Research [base de datos en línea]. Vol. 40, No. 1 (Ene., 2013); p. 214-224, ISSN 0305-0548 [citado en 6 de agosto de 2015] Disponible en ScienceDirect Research Databases.

<sup>17</sup> LAPORTE, G; LOUVEAUX, F; MERCURE, H. y JAILLET. VRP homogéneo, Citado por, ROCHA, L.; GONZÁLEZ, C. y ORJUELA, J. Una revisión al estado del arte del problema de ruteo de vehículos: Evolución histórica y métodos de solución. En: Ingeniería. Vol.16, no. 2 (2011); p. 35-55.

<sup>18</sup> VAN LANDEGHEM, H. A bi-criteria heuristic for the vehicle routing problem with time windows. Citado por, ROCHA, L.; GONZÁLEZ, C. y ORJUELA, J. Una revisión al estado del arte del problema de ruteo de vehículos: Evolución histórica y métodos de solución. En: Ingeniería. Vol.16, no. 2 (2011); p. 35-55.

Relativamente poco esfuerzo se ha dedicado al estudio del VRPSTT, el cual encuentra su origen en el problema del agente viajero (m-TSP) con tiempos de viaje estocástico (m-TSPST), estudiado por **Laporte, Louveaux y Mercure** en 1992 y **Lambert, Laporte y Louveaux** en 1993. La investigación de Laporte, Louveaux y Mercure, considera el problema del agente viajero con tiempos de viaje y servicio estocásticos, en el cual el vehículo incurre en una penalización por exceder el límite de duración de la ruta, para esto, los autores proponen un modelo de oportunidad limitada junto con un modelo de recurso de dos y tres índices, basados en un algoritmo de Branch and Cut <sup>19</sup>.

Un años más tarde, **Lambert, Laporte y Louveaux** <sup>20</sup> introducen el VRPSTT (basándose en el m-TSPST) al analizar el diseño de rutas para la recolección de dinero de sucursales bancarias en flotas de 10, 15, y 20 vehículos, con tiempos de viaje estocásticos, mediante una formulación matemática cimentada en programas enteros, y un algoritmo heurístico basado en el algoritmo de ahorro por Clarke y Wright (1964).

Posteriormente, **Park y Song**<sup>21</sup> en 1997, presentan tres heurísticas (Técnica de búsqueda prioritaria de proximidad (PRT), Técnica de inserción (INT), Técnica de Ahorro (SAT)) desarrolladas por la ampliación y modificación de heurísticas existentes para problemas de programación de vehículos convencionales, con el fin de encontrar solución al problema de ruteo de vehículos con velocidad variable en

---

<sup>19</sup>LAPORTE, G; LOUVEAUX, F; Y MERCURE, H. The vehicle routing problem with stochastic travel times, Citado por LI, Xiangyong; TIAN, Peng y LEUNG, Stephen C.H. Vehicle routing problems with time windows and stochastic travel and service times: Models and algorithm. En: International Journal of Production Economics [base de datos en línea]. Vol. 125, No. 1 (May., 2010); p.137-145, ISSN 0925-5273. [citado en 3 de agosto de 2015] Disponible en ScienceDirect Research Databases.

<sup>20</sup> LAMBERT, V., LAPORTE, G; LOUVEAUX, F. Designing collection routes through bank branches. En: Computers & Operations Research. Vol. 20, No. 7 (1993); p. 783–791.

<sup>21</sup> PARK, Yang-Byung y SONG, Sung-Hun. Vehicle scheduling problems with time-varying speed. En: Computers & Industrial Engineering [base de datos en línea]. Vol. 33, No. 3–4 (Dic.,1997); p. 853–856 [citado en 2 de Agosto de 2015] Disponible en ScienceDirect Research Databases.

el tiempo (VSPTS), el cual es una generalización del problema de programación de vehículos, en el que la velocidad de desplazamiento entre dos puntos depende de las zonas por las que transite y la hora del día (características atribuidas al VRPSTT). El modelo propuesto es un modelo simple, que estima las velocidades de desplazamiento de tiempo variable en los VSPTS y alivia a su vez la carga de los problemas relacionados con los datos. Se concluye que el método de ahorro es el mejor entre las tres heurísticas.

En el 2003, **Xie**<sup>22</sup> amplía el modelo de programación con oportunidad limitada para el VRPSTT considerando la misma capacidad de los vehículos, y utilizando un algoritmo de generación que evidencia cierto grado de éxito.

Tiempo después, en el 2007, **Woensel et al.**<sup>23</sup> creó un esquema de modelado innovador, para capturar el comportamiento estocástico de los tiempos de viaje, que introduce el componente de la congestión del tráfico basado en la teoría de colas. En este modelo, se considera la velocidad como resultado del proceso estocástico de diferentes vehículos que interactúan entre sí en determinadas circunstancias (por ejemplo, el mal tiempo), que, combinada con la distancia a recorrer, lleva al tiempo de viaje esperado. Así mismo, este enfoque analítico permite el cálculo de la varianza en los tiempos de viaje, proporcionando una evaluación de las rutas en función de la incertidumbre involucrada.

---

<sup>22</sup> XIE, B. Research on stochastic vehicle routing problems. China, 2003, Ph.D. thesis. Xinan Jiaotong University.

<sup>23</sup> WOENSEL, Tom Van; KERBACHE, Laoucine; PEREMANS, Herbert y VANDAELE, Nico. A queueing framework for routing problems with time-dependent travel times. En: Journal of Mathematical Modelling and Algorithms [base de datos en línea]. Vol. 6, No. 1 (Mar.,2007); p. 151–173 [citado en 4 de agosto de 2015] Disponible en SpringerLink Research Databases.

Dos años más tarde, **Shao et al.**<sup>24</sup> desarrollan un algoritmo híbrido de optimización por enjambre de partículas que mejora la búsqueda de la ruta y la eficacia en el VRPSTT. El algoritmo considera un modelo de probabilidad con restricciones de capacidad en los vehículos, cuya característica distintiva es que la aleatoriedad de las restricciones satisface al menos un cierto nivel de confianza. Adicionalmente, los autores incorporan un diseño de nodos, utilizando el pensamiento de codificación que transforma el VRPSTT en un problema casi continuo y convierten la función objetivo en una función de aptitud, la cual utiliza un número grande  $M$  como coeficiente de castigo para hacer frente a las restricciones, permitiendo que la solución no factible sea convertida en un gran valor, que se elimina en el proceso de iteración, por lo que el algoritmo converge gradualmente en una solución factible. Este híbrido también incluye un operador dinámico de vecindario, que expande la búsqueda en un área cada vez más grande hasta cubrir la población total, así como un valor inercial de peso  $\omega$ , que proporciona un equilibrio entre las capacidades globales y locales de expansión.

Basándose en Woensel et al, **Lecluyse et al.**<sup>25</sup> introducen la variabilidad del flujo de tráfico al modelo, lo que permite una evaluación de las rutas con base en la incertidumbre en cuestión, brindándole a la ruta óptima un tiempo de viaje un poco más largo, pero más fiable. Esta metodología incorpora en la función objetivo el equilibrio entre el tiempo de viaje promedio y su varianza.

---

<sup>24</sup> SHAO, Z; GAO, S y WANG, S. A hybrid particle swarm optimization algorithm for vehicle routing problem with stochastic travel time. En: Fuzzy information and engineering [base de datos en línea]. Vol. 54 (2009); p. 566–74 [citado en 4 de agosto de 2015] Disponible en SpringerLink Research Databases.

<sup>25</sup> LECLUYSE, C; WOENSEL, T. Van y PEREMANS, H. Vehicle routing with stochastic time-dependent travel times. En: 4OR: A Quarterly Journal of Operations Research [Base de datos en línea]. Vol. 7, No. 4 (nov.,2009); p. 363–377 [citado en 5 de Agosto de 2015] Disponible en SpringerLink Research Databases.

**Connors y Sumalee** en el 2009, junto con **Chen y Zhou** en el 2010, estudian los tiempos de viaje como estocásticos desde el punto de vista del equilibrio de los viajeros. Dicho estudio considera un marco de modelado donde el valor y las probabilidades percibidas de los resultados en los tiempos de viaje se obtienen a través de transformaciones no lineales de los tiempos de viaje reales y sus probabilidades.<sup>26</sup>

En el 2010, **Xiangyong Li, Peng Tian, Stephen CH Leung**,<sup>27</sup> examinan el problema de VRP en el que los tiempos de viaje y de servicios son estocásticos, incluyendo una restricción de ventana de tiempo asociada a cada cliente. Este problema fue formulado como un modelo de oportunidad con programación limitada, para después ser modelado como un problema de programación estocástica con recurso, en términos de diferentes criterios de optimización, donde se diseñan los conjuntos de ruta en la primera etapa, y se minimizan los costos de transporte previstos en la segunda etapa; ambos modelos, consideran dos tipos de restricciones aleatorias: Duración del recorrido y ventanas de tiempo (con un nivel dado de confianza  $\alpha$ ). Para solucionar de manera eficiente estos dos modelos, los autores proponen una heurística basada en búsqueda tabú y crean la solución inicial a partir de una solución VRPTW determinista, esta solución es generada por un algoritmo de ahorro modificado (Clarke y Wright, 1964), que tiene en cuenta las limitaciones de las ventanas de tiempo. Además, utilizan tres operadores diferentes: 2-Opt, Intercambio y Reubicar, para generar soluciones vecinas que se apliquen a la ruta elegida al azar de la solución actual, con el fin de buscar mejoras potenciales.

---

<sup>26</sup>CONNORS, R y SUMALEE, A. A network equilibrium model with travellers' perception of stochastic travel times, Citado por ZHANG, Tao; CHAOVALITWONGSE, W.A y ZHANG, Yuejie. Scatter search for the stochastic travel-time vehicle routing problem with simultaneous pick-ups and deliveries. En: Computers & Operations Research [base de datos en línea]. Vol. 39, No. 10 (oct., 2012); p. 2277-2290, ISSN 0305-0548 [citado en 5 de agosto de 2015] Disponible en ScienceDirect Research Databases.

<sup>27</sup> LI, Op.cit., p.137-145.

**Duygu Tas, Nico Dellaert, Tom van Woensel, Ton de Kok,**<sup>28</sup> en el 2013, proponen el primer modelo que separa los costos de transporte (distancia total recorrida, el número de vehículos utilizados y total de horas extra de pilotos) y los costos de servicios (llegadas anticipadas o tardías) para el problema de ruteo de vehículos con ventanas de tiempo suaves y tiempos de viaje estocásticos, donde, las violaciones de las ventanas de tiempo son atendidas directamente en la función objetivo y el aspecto estocástico en los tiempos de viaje desempeña un papel importante en los cálculos de ambos componentes del costo. El enfoque de solución propuesto por los autores está constituido por tres fases; en la primera fase, construyen una solución inicial, basada en la heurística de inserción I1 de Solomon, esta solución es mejorada por una metaheurística de Búsqueda Tabú que incluye una aplicación de memoria a mediano plazo, y por último, utilizan un método de post-optimización que ajusta la hora de salida de cada vehículo asignado desde el depósito, para reducir el costo total del servicio en la ruta correspondiente. A partir de esto, se encuentra que los mejores resultados se obtienen gracias a los arreglos incorporados en la obtención de la solución inicial.

Después, **D. Taş, M. Gendreau, N. Dellaert, T. Van Woensel, A.G. de Kok,**<sup>29</sup> encaminan su estudio hacia el problema de ruteo de vehículos con ventanas de tiempo suaves, considerando los tiempos de viaje estocásticos con una distribución de probabilidad conocida. El objetivo de este problema es minimizar la suma de los costos de transporte, los cuales se deben a la distancia total recorrida, el número de vehículos utilizados y el tiempo extra de espera total de los conductores, así

---

<sup>28</sup> TAŞ, Duygu; DELLAERT, Nico; WOENSEL, Tom van y DE KOK, Ton. Vehicle routing problem with stochastic travel times including soft time windows and service costs. En: Computers & Operations Research [base de datos en línea]. Vol. 40, No. 1 (Ene., 2013); p. 214-224, ISSN 0305-0548 [citado en 6 de agosto de 2015] Disponible en ScienceDirect Research Databases.

<sup>29</sup> TAŞ, D; GENDREAU, M; DELLAERT, N; WOENSEL, T. van y DE KOK, A.G. Vehicle routing with soft time windows and stochastic travel times: A column generation and branch-and-price solution approach. En: European Journal of Operational Research [base de datos en línea]. Vol. 236, No. 3 (ago., 2014); p.789-799, ISSN 0377-2217 [citado en 6 de agosto de 2015] Disponible en ScienceDirect Research Databases.

como los costos del servicio incurridos por llegadas tempranas y tardías. Los autores proponen un método de solución exacto basado en el algoritmo de generación de columnas incrustado dentro de un esquema de Branch and Price para obtener soluciones enteras. En el procedimiento, el problema principal fue modelado como un problema de partición clásica y el sub problema de precios para cada vehículo, corresponde a un problema de Ruta Primaria corta con limitaciones de recursos (ESPPRC). El modelo presenta soluciones factibles en problemas de mediano y gran tamaño.

Otras variantes del VRP tratadas en el mismo año consideran en la formulación del problema, las ventanas de tiempo (2013). **Raúl Baños, Julio Ortega, Consolación Gil, Antonio L. Márquez y Francisco de Toro**,<sup>30</sup> centran su investigación en resolver el problema de ruteo de vehículos con ventanas de tiempo, adicionándole un enfoque multi objetivo que pretende minimizar la distancia total recorrida por los vehículos para el servicio al cliente, así como minimizar el desequilibrio de las cargas de trabajo (distancias recorridas / bienes entregados por los vehículos). El modelo propuesto es un algoritmo evolutivo basado en Pareto, utilizando un recocido simulado que le brinda al procedimiento un comportamiento de exploración - explotación equilibrada.

También, **Slim Belhaiza, Pierre Hansen y Gilbert Laporte**,<sup>31</sup> proponen una heurística que incorpora el concepto de memoria adaptativa utilizada en la búsqueda tabú, en el marco de la búsqueda variable de vecindario, para resolver el

---

<sup>30</sup> BAÑOS, Raúl; ORTEGA, Julio; GIL, Consolación; MÁRQUEZ, Antonio L. y DE TORO, Francisco. A hybrid meta-heuristic for multi-objective vehicle routing problems with time windows. En: Computers & Industrial Engineering [base de datos en línea]. Vol. 65, No. 2 (jun., 2013); p. 286-296, ISSN 0360-8352 [citado en 7 de agosto de 2015] Disponible en ScienceDirect Research Databases.

<sup>31</sup> BELHAIZA, Slim; HANSEN, Pierre y LAPORTE, Gilbert. A hybrid variable neighborhood tabu search heuristic for the vehicle routing problem with multiple time windows. En: Computers & Operations Research. [Base de datos en línea] Vol. 52, Part B (dic., 2014); p. 269-281, ISSN 0305-0548. [citado en 9 de agosto de 2015] Disponible en ScienceDirect Research Databases.

VRP, donde los clientes ofrecen una selección de plazos de entrega, (múltiples ventanas de tiempo).

En el 2014, **Duygu Tas, Nico Dellaert, Tom Van Woensel, Ton de Kok**,<sup>32</sup> examinan el problema de ruteo de vehículos con ventanas de tiempo suave, donde los tiempos de viaje se modelan con respecto a la dependencia en el tiempo y la incertidumbre. La investigación es dividida en dos fases, la primera, excluyendo los tiempos de servicio con el fin de poder derivar las distribuciones exactas de los valores de media y varianza de los tiempos de llegada, y la segunda, aumentando la complejidad del problema, incluyendo los tiempos de servicio en los que se necesitan aproximaciones de las distribuciones del tiempo de llegada, ya que las distribuciones exactas de estas no se pueden derivar. Para dar solución a este problema, los autores proponen la construcción de dos metaheurísticas, Búsqueda Tabú (TS) y Búsqueda grande de vecinadario adaptativa (ANLS), mejorando las soluciones a través de la implementación de un método de post-optimización. Los métodos construidos se desempeñan mejor en una red dinámica y estocástica dependiente del tiempo, que sobre una red que considera la independencia del tiempo.

---

<sup>32</sup> TAŞ, Duygu; DELLAERT, Nico; WOENSEL, Tom van y DE KOK, Ton. The time-dependent vehicle routing problem with soft time windows and stochastic travel times. En: *Transportation Research Part C: Emerging Technologies* [base de datos en línea]. Vol. 48 (Nov., 2014); p.66-83, ISSN 0968-090X [citado en 2 de Agosto de 2015] Disponible en ScienceDirect Research Databases.

## 5. MARCO DE ANTECEDENTES

En la Universidad Industrial de Santander se han desarrollado proyectos a nivel de pregrado y maestría que buscan dar solución al problema de ruteo de vehículos con variables estocásticas.

En el 2012, Adriana Lozada Díaz y Ricardo Andrés Cadena González <sup>33</sup> desarrollaron el proyecto “Solución del problema de ruteo de vehículos con ventanas de tiempo (VRPTW) mediante métodos heurísticos” en donde se estudiaron y sistematizaron cinco heurísticas de construcción de rutas que permiten solucionar las instancias más comunes del VRPTW de manera rápida y eficiente.

El VRPTW, Ruteo de Vehículos con Ventanas de Tiempo por sus siglas en inglés, describe de manera más adecuada la actividad de distribución física, al tomar en consideración tanto restricciones espaciales como temporales. Básicamente, para esta variante del problema de ruteo, se busca servir la demanda de un conjunto de clientes esparcidos geográficamente, dentro de un intervalo de tiempo donde se permite el servicio. Los autores proponen dar solución a este problema a través de métodos heurísticos de construcción de ruta programados computacionalmente y miden la eficiencia de los algoritmos con un banco de prueba de 56 instancias del tamaño de 100 nodos clasificadas en los grupos Random (R), Clúster (C) y Random-Clúster (RC) de Solomon (1987).

Las heurísticas estudiadas por los autores se dividen en heurísticas de construcción de rutas como el algoritmo de ahorros de Clarke & Wright, que pretende resumir la

---

<sup>33</sup> LOZADA DÍAZ, Adriana y CADENA GONZÁLEZ, Ricardo Andrés. Solución del problema de ruteo de vehículos con ventanas de tiempo (VRPTW) mediante métodos heurísticos. Bucaramanga, 2012. Tesis de Pregrado de Ingeniería Industrial. Universidad Industrial de Santander.

distancia que se genera cuando se unen dos clientes en una ruta, en lugar de servir a cada uno de ellos en rutas diferentes, y Heurísticas de inserción secuencial como la heurística de inserción más cercana, o del vecino más cercano, donde se busca una solución basada en la cercanía de dos nodos o clientes adyacentes usando una métrica o medida que hace un balance ponderado entre la cercanía geográfica de los clientes y el tiempo de recorrido respectivo de un nodo a otro. Las tres siguientes heurísticas tratadas en el proyecto, son las heurísticas de inserción de Solomon (Solomon 1987) (I1, I2 e I3), que fueron inicialmente desarrolladas para el problema de Ruteo de Vehículos con Ventanas de Tiempo; básicamente estos algoritmos eligen un criterio para comenzar un itinerario y a continuación unas reglas de inserción de clientes. Los métodos estudiados fueron optimizados a través de la implementación de la técnica de intercambio 2-Opt.

Los resultados obtenidos muestran que la heurística de inserción I1 se destaca como la más eficiente para las instancias de corto horizonte de programación de tipo Random (R) y Random-clúster (RC) y la heurística del vecino más cercano entrega mejores resultados respecto a los de la literatura de referencia en todos los grupos de instancias (R, C y RC) para las instancias con largo horizonte de programación.

En el 2013, Camilo Cruz Jiménez y Ludy Juliana González Villamizar<sup>34</sup> en su trabajo de grado “Desarrollo de un algoritmo híbrido para la resolución del problema de ruteo de vehículos con entrega y recogida simultáneas (VRPSPD)” proponen una solución a una variante del VRP en el cual además de entregar bienes a los clientes, la flota de vehículos debe recoger otros bienes desde la ubicación de los clientes de manera simultánea, considerando las respectivas restricciones de capacidad y

---

<sup>34</sup> CRUZ JIMÉNEZ, Camilo; GONZÁLEZ VILLAMIZAR, Ludy Juliana. Desarrollo de un algoritmo híbrido para la resolución del problema de ruteo de vehículos con entrega y recogida simultáneas (VRPSPD). Bucaramanga, 2013. Tesis de Pregrado de Ingeniería Industrial. Universidad Industrial de Santander.

tiempo con el objetivo de minimizar costos, distancias o tiempos de viaje. El análisis de este problema está directamente ligado a los estudios de logística inversa. Las aplicaciones del VRPSPD son encontradas en sistemas de distribución donde los productos que ya cumplieron su vida útil requieren de un adecuado procesamiento o disposición final, así como en formatos comerciales donde el producto no vendido se puede devolver al fabricante o aquellos en los que se pueden reincorporar elementos a la cadena productiva o reutilizarlos tales como pallets, contenedores, envases, entre otros. Para esto los autores plantean un algoritmo híbrido utilizando el enfoque de Enjambre de Partículas combinado con heurísticas de construcción y mejoramiento de rutas que operan en las fases de codificación de la solución (heurística de inserción más barata y 2-opt). La heurística de inserción más barata aporta al híbrido la evaluación de todas las posiciones en las que un nuevo cliente pueda ser insertado en una ruta existente donde haya al menos 1 cliente asignado. Por otro lado, la heurística 2-opt de búsqueda local, se utiliza con el fin de re optimizar las rutas que se construyen en la fase de decodificación de la solución.

El algoritmo se evaluó en 18 instancias de gran tamaño introducidas por Tang y Galvão que involucran entre 100 y 400 clientes. A partir del desarrollo de este proyecto se pudo comprobar que para problemas con gran número de clientes, el algoritmo presenta mayor dificultad en encontrar soluciones que tengan menor desviación en su costo con respecto a las mejores soluciones conocidas, además los autores encontraron que la interacción entre el tamaño del problema y la capacidad de los vehículos implica variaciones significativas en la obtención de resultados, ya que a medida que el problema aumenta de tamaño y se reduce el umbral de la restricción de la capacidad, el algoritmo se hace menos eficiente.

En el mismo año, Francia Del Pilar Acosta Niño Y Daniela Angélica Osorio Osorio<sup>35</sup>, investigaron el problema de ruteo de vehículos con demandas estocásticas, en su trabajo “Un método evolutivo de colonia de hormigas para la solución del problema de ruteo de vehículos con demandas estocásticas”. En el VRPSD, la demanda de los clientes es una variable aleatoria, es decir está caracterizada por una distribución de probabilidad; y su complejidad radica en que solo es posible conocer la demanda real en el momento que el vehículo llega al cliente. Este comportamiento aleatorio incurre en penalizaciones cuando la demanda de la ruta excede la capacidad del vehículo, produciendo un “fallo de ruta”.

Con el fin de encontrar una buena solución a este problema los autores fundamentan su solución en la metaheurística Colonia de Hormigas tradicional, mejorándola mediante la implementación de algunos operadores evolutivos para ampliar la exploración de los agentes hacia nuevas zonas del espacio de búsqueda (hormigas artificiales) y, una heurística de búsqueda local, con el fin de explotar las mejores soluciones (intercambio 2-Opt). Estos algoritmos aceleran la convergencia del algoritmo para encontrar soluciones cercanas a la óptima en problemas de optimización complejos como los NP- Hard. El banco de prueba empleado para validar la eficacia y la eficiencia del método propuesto, fue el generado en la propuesta de investigación realizada por Silvia Galván. Para el estudio del efecto de la variación de los parámetros del algoritmo EACO en el valor de la función objetivo, se llevó a cabo un diseño experimental donde se determinaron dos niveles en cada uno de los factores contemplados, en el nivel 1 el número de clientes tomaba el valor de 100 y en el nivel 2 tomaba el valor de 200. Los resultados revelaron que el método propuesto supera el rendimiento del algoritmo AS tradicional en las ocho instancias comparadas, hecho que se logró gracias a la capacidad que presentaba

---

<sup>35</sup> ACOSTA NIÑO, Francia del Pilar; OSORIO OSORIO, Daniela Angélica. Un método evolutivo de colonia de hormigas para la solución del problema de ruteo de vehículos con demandas estocásticas. Bucaramanga, 2013. Tesis de pregrado de Ingeniería Industrial. Universidad Industrial de Santander.

el algoritmo de equilibrar la exploración y explotación. Sin embargo, se observó que el algoritmo presentó mejor desempeño en las instancias con 100 clientes, con una diferencia porcentual superior al 1%, con respecto a las de 200 clientes.

En el trabajo de investigación de Natalia Alejandra Gélves Tello y Ricardo Andrés Mora Moreno,<sup>36</sup> realizado en 2015, se parte de la idea de que para que los problemas de ruteo de vehículos presentados en la literatura se asemejen más a las situaciones de la vida real es necesario involucrar variables estocásticas en los planteamientos y considerar el uso de métodos aproximados para encontrar la solución. En este estudio se implementó el método heurístico denominado optimización por espiral para la solución del problema de ruteo de vehículos con demandas estocásticas (VRPSD), con el enfoque a priori y la estrategia de reabastecimiento preventivo para un solo vehículo. El método se mejoró mediante la heurística del vecino más cercano con el fin de inicializar las rutas y posteriormente se utilizó la mutación (operador evolutivo), para ampliar la zona de exploración de las espirales. Por último, se utilizó una heurística de búsqueda local llamada K-intercambio con el propósito de intensificar la búsqueda en la vecindad de soluciones óptimas encontradas.

Adicionalmente se realizó un diseño de experimentos  $2^3$ , con el fin de determinar la influencia de cada factor en la función objetivo. Las instancias utilizadas en este trabajo de investigación fueron las propuestas por la ingeniera Silvia Andrea Galván Núñez en el proyecto “Optimización por simulación basado en un sistema evolutivo de optimización de enjambre de partículas para el problema de ruteo de vehículos con demandas estocásticas”, para de esta manera verificar la efectividad del algoritmo. Los resultados se compararon con un algoritmo evolutivo de optimización

---

<sup>36</sup> GÉLVES TELLO, Natalia Alejandra; MORA MORENO, Ricardo Andrés. Solución del problema de ruteo de vehículos con demandas estocásticas (VRPSD) mediante la metaheurística optimización por espiral. Bucaramanga, 2015. Tesis de pregrado de Ingeniería Industrial. Universidad Industrial de Santander.

por enjambre de partículas, evidenciándose mejoras de hasta el 5.71% en todas las instancias.

## 6. MARCO TEÓRICO

### 6.1 COMPLEJIDAD COMPUTACIONAL

Cuando se pretende resolver un problema de optimización es necesario clasificarlo previamente en función de la teoría de la complejidad algorítmica. Los criterios de la complejidad algorítmica son el espacio y el tiempo, es decir, cuánta memoria y cuántos pasos se requieren para resolver el problema.<sup>37</sup>

Un requerimiento mínimo para que un algoritmo sea considerado eficiente es que su tiempo de ejecución o corrida sea polinomial:  $O(n)^c$ , donde  $c$  es alguna constante y  $n$ , el tamaño de las entradas. Los investigadores reconocieron de manera oportuna que no todos los problemas se pueden resolver rápidamente, no obstante, les tomó más tiempo averiguar cuáles pueden y cuáles no.<sup>38</sup>

Existen tres clases de problemas de decisión: P, NP, y co-NP.<sup>39</sup>

Los problemas tipo P son aquellos problemas de decisión que pueden ser resueltos en tiempo polinomial. Intuitivamente, se considera que pueden ser resueltos de forma rápida. La mayoría de problemas de ordenación y de búsqueda pertenecen a esta clase.

Los problemas NP, son aquellos que tienen la propiedad de verificar en tiempo polinomial que la respuesta a un problema sea *SÍ*. Es decir, se puede validar una

---

<sup>37</sup> BAÑOS NAVARRO, Raúl. Metaheurísticas Híbridas para Optimización Mono-objetivo y Multi-Objetivo: Paralelización y Aplicaciones. Almería, 2006, 100p. Trabajo de Grado (Doctor en Informática). Universidad de Almería. Departamento de Arquitectura de computadores y electrónica.

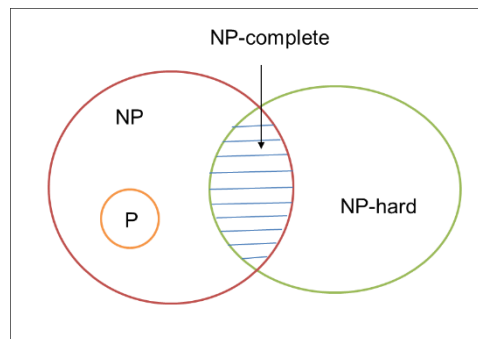
<sup>38</sup> ERICKSON, Jeff. Algorithms: NP-Hardness [en línea]. <<http://www.cs.uiuc.edu/~jeffe/teaching/algorithms>> [citado en 15 de junio de 2016].

<sup>39</sup> IGLESIAS MARTÍNEZ, José, *et al.* Complejidad Computacional. Grado Ingeniería informática. Teoría de Autómatas y Lenguajes formales. [en línea] <<http://ocw.uc3m.es/ingenieria-informatica/teoria-de-automatasy-lenguajes-formales/material-de-clase-1/tema-8-complejidad-computacional>> En: Universidad Carlos III de Madrid. [Citado el 12 de Julio del 2014].

respuesta (SÍ) rápidamente, si se tiene en frente. Los problemas co-NP que son esencialmente opuestos a los problemas NP, es decir, si la respuesta a un problema en co-NP es *NO*, entonces se considera que este hecho se puede verificar en tiempo polinomial. Cuando un problema NP se puede resolver en tiempo polinomial se puede decir que  $P=NP$ . Los problemas NP-Hard no tienen ningún algoritmo polinómico que los resuelva, por tanto, utiliza algoritmos que ofrezcan una respuesta aproximada a la óptima en un tiempo computacional razonable.

Finalmente, los problemas NP-completos son los problemas NP más difíciles, en el sentido de que es menos probable que pertenezcan a la clase P, ya que si se pudiera encontrar la forma de resolver cualquier problema NP-completo rápidamente (en un tiempo polinómico), entonces sería posible utilizar ese algoritmo para resolver todos los problemas NP rápidamente.

**Figura 1. Problemas P, NP, NP-hard, NP-completo**



Fuente: Methods and Analysis of Algorithms.<sup>40</sup>

---

<sup>40</sup> BASU, S.K. DESIGN Methods and Analysis of Algorithms. Second Edition. New Delhi: PHI Learning Private Limite, 2013. 369 p.

El Problema de Ruteo de Vehículos (VRP) es por definición un problema de optimización de clase NP-hard, lo que indica que el esfuerzo computacional requerido para resolverlo incrementa exponencialmente de acuerdo al tamaño de las entradas.<sup>41</sup>

## 6.2 OPTIMIZACIÓN COMBINATORIA

La optimización se concibe como el proceso de intentar encontrar la mejor solución posible a un problema de optimización, generalmente en un tiempo limitado. Los problemas de optimización se pueden dividir en dos categorías: aquellos en que la solución está codificada mediante valores reales y aquellos cuya solución está codificada en valores enteros. La optimización combinatoria pertenece al segundo grupo.<sup>42</sup>

La Optimización combinatoria tiene sus raíces en la combinatoria, la investigación de operaciones y las ciencias de la computación teórica. La motivación principal para su aplicación, es que miles de problemas de la vida real pueden ser formulados como problemas abstractos de optimización combinatoria.

Un problema de optimización combinatoria  $P(S, f)$  puede ser definido por:

- ∴ Un conjunto de variables,  $X = \{ X_1, \dots, X_n \}$ .
- ∴ Dominio de las variables,  $D_1, \dots, D_n$ .
- ∴ Restricciones entre variables.
- ∴ Una función objetivo “ $f$ ” para ser minimizada o maximizada. Donde  $f: D_1 \times \dots \times D_n \rightarrow \mathbb{R}^+$ .

---

<sup>41</sup> OCHOA, Carlos Alberto, *et al.* Logistics Management and Optimization through Hybrid Artificial Intelligence Systems. United States of America: IGI Global, 2012. 511p.

<sup>42</sup> DUARTE MUÑOZ, Abraham. Metaheurística. España: S.L.-DYKINSON, 2007. 220 p.

El conjunto de todas las posibles asignaciones factibles es:

$$S = \{s = \{(X_1, V_1) \dots, (X_n, V_n)\} \mid V_i \in D_i\}$$

Dado que  $s$  satisface todas las restricciones.

Donde  $S$  es el conjunto de soluciones candidatas en el espacio de búsqueda. Para resolver un problema de optimización combinatoria se tiene que encontrar una solución  $s_1 \in S$  que minimice o maximice el valor de la función objetivo; entonces,  $f(s_1) \leq f(s_n)$  si se desea minimizar la función objetivo o  $f(s_1) > f(s_n)$  si se desea maximizar la función objetivo.<sup>43</sup>

**6.2.1 Métodos de Solución:** El problema del agente viajero (TSP), el problema de asignación cuadrática (QAP), los problemas de horarios y programación, etc., son problemas de optimización combinatoria. Para hacerles frente se han desarrollado dos tipos de algoritmos, exactos y aproximados.

**6.2.1.1 Algoritmos Exactos:** Los algoritmos exactos garantizan una solución óptima en un tiempo acotado para cada instancia de tamaño finito de un problema de optimización. Los algoritmos exactos son aquellos que parten de una formulación con modelos de programación lineal (enteros) o similares, y llegan a una solución factible (entera) gracias a algoritmos de acotamiento del conjunto de soluciones factibles.

---

<sup>43</sup> BLUM, C. y ROLI, A. Metaheuristics in combinatorial optimization: Overview and conceptual Comparison. En: ACM Computing Surveys (CSUR). Vol. 35, No. 3 (Sep., 2003); p. 268-308.

Sin embargo, es importante resaltar que para los problemas de optimización combinatoria con característica NP- hard, no hay un algoritmo que arroje una respuesta en tiempo polinomial; por lo que los métodos exactos necesitarían tiempo de cálculo exponencial, el cual es demasiado alto para fines prácticos. Es así como los métodos aproximados han recibido más atención en los últimos 30 años.<sup>44</sup>

- ∴ **Método Simplex:** El método simplex fue propuesto por George Dantzig en 1947, con el fin de resolver problemas de larga escala en el marco de la programación lineal. Su funcionamiento consiste en determinar todos los puntos extremos de una solución para comprobar si existe alguna solución factible; una vez conocidos, se procede a probar si un punto extremo arroja una solución óptima en la función objetivo, si no pasa la prueba de optimalidad, se pasa a un punto extremo adyacente para hacer la misma prueba. El proceso se detiene hasta que se encuentre un punto extremo óptimo o cuando ninguna de las aristas de la actual posición conlleve a una tasa de mejoramiento positiva, por lo que establece a esta como la solución óptima.<sup>45</sup>
  
- ∴ **Branch and Bound:** El algoritmo de Branch and Bound explora el espacio de búsqueda de manera más eficiente que una búsqueda exhaustiva. Este organiza la búsqueda para rechazar muchos subconjuntos que están garantizados serán sub óptimos sin calcular sus valores J (una función criterio que se utiliza para evaluar la calidad de la característica del subconjunto seleccionado) y por lo tanto corre drásticamente más rápido que una búsqueda exhaustiva. El algoritmo de BB se ha empleado con éxito en

---

<sup>44</sup> MARTÍ, R. Procedimientos metaheurísticos en optimización combinatoria. En: Departament d'Estadística i Investigació Operativa. Facultat de Matemàtiques. Universitat de València. Vol. 1, No. 1 (2003); p. 1-60.

<sup>45</sup> YANG, X. Introduction to Mathematical Optimization: From Linear Programming to Metaheuristics. Reino Unido: Cambridge, GBR: Cambridge International Science Publishing, 2008. 150 p.

muchas tareas, tales como minimizar el tiempo total de terminación en un problema de planificación de tareas, lograr una distribución óptima en un entorno de fabricación y optimizar el diseño de una red óptica.<sup>46</sup>

**6.2.1.2 Métodos Aproximados:** Los métodos aproximados sacrifican la garantía de encontrar soluciones óptimas en aras de obtener buenas soluciones en una cantidad significativamente menor de tiempo. Entre los métodos aproximados básicos se distinguen principalmente los métodos constructivos y los métodos de búsqueda local.

- ∴ **Algoritmos Constructivos:** Generan soluciones desde cero mediante la adición de componentes a una solución parcial, inicialmente vacía, hasta que esta se completa. Por lo general es el método aproximado más rápido, sin embargo, a menudo regresan soluciones de calidad inferior en comparación con los algoritmos de búsqueda local.
  
- ∴ **Algoritmos de búsqueda local:** Parten de alguna solución inicial y de forma iterativa tratan de sustituir la solución actual por una mejor solución, en un vecindario apropiadamente definido de la solución actual.

Además de estos, los métodos aproximados se clasifican en otras categorías:

---

<sup>46</sup> NAKARIYAKUL, Songyot. A comparative study of suboptimal branch and bound algorithms. En: Information Sciences [base de datos en línea]. Vol. 278, No. 10 (Sep., 2014); p. 545-554, ISSN 0020-0255 [citado en 10 de agosto de 2015] Disponible en ScienceDirect Research Databases.

- ∴ **Algoritmos de Descomposición:** En estos, el problema original se descompone en sub problemas más sencillos de resolver, teniendo en cuenta que, de manera general, ambos pertenecen al mismo problema.
  
- ∴ **Métodos Inductivos:** Pretenden generalizar de versiones más pequeñas o más sencillas al caso completo. Algunas propiedades técnicas identificadas en estos casos (más fáciles de analizar) pueden ser aplicadas al problema completo.
  
- ∴ **Métodos de Reducción:** Identifican propiedades que se cumplen mayoritariamente en las buenas soluciones y las introducen como restricciones del problema. El objetivo es restringir el espacio de soluciones simplificando el problema. El riesgo obvio es dejar fuera las soluciones óptimas del problema original.

Dentro de los métodos aproximados se encuentran las heurísticas y metaheurísticas.<sup>47</sup>

### 6.3 HEURÍSTICA

Las heurísticas son algoritmos que encuentran soluciones de buena calidad para problemas combinatorios complejos; o sea, para problemas tipo NP. Los algoritmos heurísticos son fáciles de implementar y encuentran buenas soluciones con esfuerzos computacionales relativamente pequeños; sin embargo, renuncian (desde el punto de vista teórico) a encontrar la solución óptima global de un

---

<sup>47</sup> MARTÍ, Op.cit., p. 1- 60.

problema. En problemas de gran tamaño, rara vez un algoritmo heurístico encuentra la solución óptima global.

#### **6.4 METAHEURÍSTICA**

El término metaheurística fue introducido por Fred Glover en 1986. Los profesores Osman y Kelly (1995) lo definieron como:

“Los procedimientos metaheurísticos son una clase de métodos aproximados que están diseñados para resolver problemas difíciles de optimización combinatoria, en los que los heurísticos clásicos no son efectivos. Las metaheurísticas proporcionan un marco general para crear nuevos algoritmos híbridos, combinando diferentes conceptos derivados de la inteligencia artificial, la evolución biológica y los mecanismos estadísticos.”<sup>48</sup>

Existen ciertas propiedades fundamentales que caracterizan a este tipo de métodos:

- ∴ Las metaheurísticas son estrategias o plantillas generales que guían el proceso de búsqueda.
- ∴ El objetivo es una exploración eficiente del espacio de búsqueda para encontrar soluciones casi óptimas.
- ∴ Las metaheurísticas son algoritmos no exactos y generalmente son no deterministas.
- ∴ Pueden incorporar mecanismos para evitar regiones no prometedoras del espacio de búsqueda.

---

<sup>48</sup> Ibid., p. 1-60.

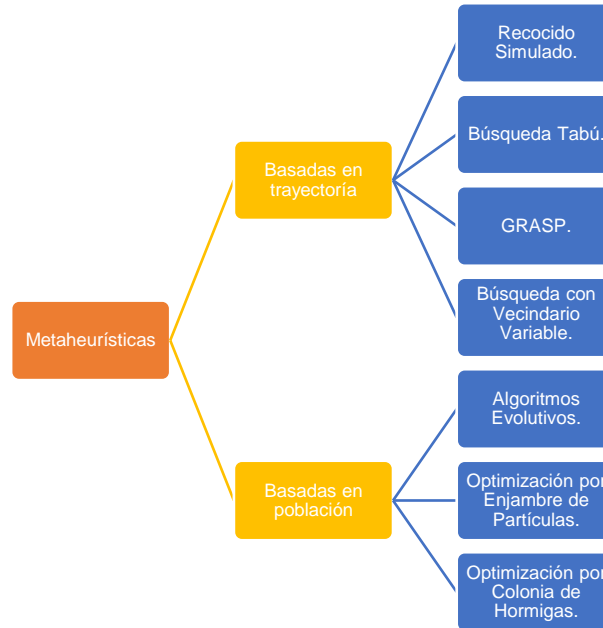
∴ El esquema básico de cualquier metaheurística tiene una estructura predefinida.

Resumiendo, se puede acordar que una metaheurística es una estrategia de alto nivel que usa diferentes métodos para explorar el espacio de búsqueda. En otras palabras, una metaheurística es una plantilla general no determinista, que debe ser rellenada con datos específicos del problema (representación de las soluciones, operadores para manipularlas, etc.) y que permite abordar problemas con espacios de búsqueda de gran tamaño. En este tipo de técnicas es especialmente importante el correcto equilibrio (generalmente dinámico) que haya entre diversificación e intensificación. El término diversificación se refiere a la evaluación de soluciones en regiones distantes del espacio de búsqueda (de acuerdo a una distancia previamente definida entre soluciones), también se conoce como exploración del espacio de búsqueda. El término intensificación, por otro lado, se refiere a la evaluación de soluciones en regiones acotadas y pequeñas con respecto al espacio de búsqueda, centradas en el vecindario de soluciones concretas (explotación del espacio de búsqueda). El equilibrio entre estos dos aspectos contrapuestos es de gran importancia, ya que por un lado deben identificarse rápidamente las regiones prometedoras del espacio de búsqueda global y por otro lado, no se debe malgastar tiempo en las regiones que ya han sido exploradas o que no contienen soluciones de alta calidad.

Hay diferentes formas de clasificar y describir las técnicas metaheurísticas. Dependiendo de las características que se seleccionen se pueden obtener diferentes taxonomías: basadas en la naturaleza y no basadas en la naturaleza, con memoria o sin ella, con una o varias estructuras de vecindario, etc. Una de las clasificaciones más populares las divide en metaheurísticas basadas en trayectoria y basadas en población. Las primeras manipulan en cada paso un único elemento

del espacio de búsqueda, mientras que las segundas trabajan sobre un conjunto de ellos (población).<sup>49</sup>

**Figura 2. Clasificación de las metaheurísticas**



Fuente. Metaheurísticas e Ingeniería del Software<sup>50</sup>

---

<sup>49</sup> CHICANO, J. Metaheurísticas e Ingeniería del Software. Málaga, 2007. Tesis doctoral. Universidad de Málaga.

<sup>50</sup> Ibid., p.45-69.

**6.4.1 Metaheurísticas basadas en la trayectoria:** La principal característica de estos métodos es que parten de una solución y mediante la exploración del vecindario van actualizando la solución actual, formando una trayectoria. La mayoría de estos algoritmos surgen como extensiones de los métodos de búsqueda local simples a los que se les añade algún mecanismo para escapar de los mínimos locales.

Esto implica la necesidad de una condición de parada más elaborada que la de encontrar un mínimo local. Normalmente se termina la búsqueda cuando se alcanza un número máximo predefinido de iteraciones, se encuentra una solución con una calidad aceptable o se detecta un estancamiento del proceso.

∴ **Recocido Simulado (SA):** Esta metaheurística fue propuesta por Kirkpatrick, Gelatt y Vecchi en 1983. Es un procedimiento de búsqueda local aleatorio en el que una modificación a la solución actual (que conduce a un aumento del costo de la solución), puede ser aceptada con cierta probabilidad.

Este algoritmo está inspirado en el proceso físico de templado de metales. Para conseguir que la estructura molecular del metal tenga las propiedades deseadas de resistencia o flexibilidad, es necesario controlar la velocidad del proceso de templado (enfriamiento). Si se hace adecuadamente, el estado final del metal es un estado de mínima energía.

En un contexto de optimización combinatoria, una solución corresponde a un estado del sistema físico y el costo de la solución, a la energía del sistema. En cada iteración, la solución actual es modificada por la selección aleatoria de un movimiento. Si la nueva solución ofrece una mejora, es automáticamente aceptada y se convierte en la nueva solución actual. De lo contrario, la nueva solución es aceptada de acuerdo con el criterio de Metrópolis, donde la probabilidad de aceptación está relacionada con la magnitud del aumento del costo y un parámetro llamado temperatura. Básicamente, es más probable que

sea aceptado un movimiento si la temperatura es alta y el aumento del costo es bajo. El parámetro de temperatura se reduce progresivamente, cuando la temperatura es lo suficientemente baja; sólo los movimientos que mejoran la función son aceptados y el método se detiene en un óptimo local. A diferencia de la mayoría de metaheurísticas, este método converge asintóticamente a un óptimo global (suponiendo un número infinito de iteraciones).<sup>51</sup>

- ∴ **Búsqueda Tabú (TS):** La búsqueda tabú usa mecanismos inspirados en la memoria humana como la memoria a corto y largo plazo para almacenar información del camino seguido al recorrer el espacio de búsqueda. La memoria a corto plazo es usada para prevenir que se visite la misma solución más de una vez (ciclo) y la memoria a largo plazo para buscar en diferentes áreas del espacio de solución. La búsqueda tabú empieza con una solución factible y luego se evalúan las soluciones del vecindario, de tal manera que la solución más prometedora es seleccionada. Esta estrategia permite la selección de los movimientos que deterioran la calidad de la solución con el fin de escapar de los óptimos locales.<sup>52</sup>

La memoria a corto plazo contiene la información acerca de los movimientos tabú, dichos movimientos son aquellos que no pueden ser llevados a cabo durante un cierto número de iteraciones y la memoria a largo plazo contiene información acerca del número de veces que un movimiento puede realizarse.<sup>53</sup>

- ∴ **Búsqueda con Vecindario Variable (VNS):** La búsqueda con vecindario variable fue introducida por Mladenović y Hansen en 1997. Es una

---

<sup>51</sup> GENDREAU, Op.cit., p. 189-213.

<sup>52</sup> CHICANO, Op.cit.

<sup>53</sup> BOUSSAÏD, Ilhem; LEPAGNOT, Julien y SIARRY, Patrick. A survey on optimization metaheuristics. En: Information Sciences [base de datos en línea]. Vol.237 (Jul., 2013); p.82-117, ISSN 0020-0255 [citado en 10 de Agosto de 2015] Disponible en ScienceDirect Research Databases.

metaheurística basada en la exploración sistemática de las diferentes estructuras de vecindario dentro de una rutina de búsqueda local. En términos generales, hay un cambio de vecindario cada vez que la búsqueda local se detiene alcanzando un óptimo local; esto evita que el algoritmo quede atrapado, dado que un óptimo local puede no permanecer como óptimo si se considera otra estructura de vecindad. VNS se ha aplicado con éxito a una variedad de problemas de optimización combinatoria y tiene una aplicación simple cuando se compara con otras metaheurísticas dado que generalmente no considera muchos parámetros.<sup>54</sup>

#### 6.4.2 Metaheurísticas basadas en población

- ∴ **Algoritmos Evolutivos (EA):** Los algoritmos evolutivos representan una gran clase de metodologías de resolución de problemas, como algoritmos genéticos (GA) (Holland, 1975). Estos algoritmos están motivados por la forma como las especies evolucionan y se adaptan a su medio ambiente, basado en el principio de selección natural de Darwin. Bajo este paradigma, una población de soluciones, a menudo codificada como una cadena de bits (números enteros, referidos como un cromosoma) evoluciona de una generación a la siguiente a través de la aplicación de operadores que imitan a los encontrados en la naturaleza: Selección de los más aptos, cruce y mutación. A través del proceso de selección, que está sesgado probabilísticamente hacia los mejores elementos de la población, las mejores soluciones se convierten en padres y se les permite generar descendencia.

---

<sup>54</sup> MACEDO, Rita; ALVES, Cláudio; HANAFI, Saïd; JARBOUi, Bassem; MLADENOVÍĆ, Nenad; RAMOS, Bruna y DE CARVALHO J.M. Valério. Skewed general variable neighborhood search for the location routing scheduling problem. En: Computers & Operations Research [base de datos en línea]. Vol. 61 (Sep.2015); p.143-152, ISSN 0305-0548 [citado en 11 de agosto de 2015] Disponible en ScienceDirect Research Databases.

El proceso de apareamiento, llamado cruce, toma dos soluciones de padres seleccionados y combina sus características más deseables para crear una o dos soluciones de descendencia. Esto se repite hasta que se crea una nueva población de soluciones descendientes. Antes de reemplazar la antigua población, cada miembro de la nueva población se somete (con una pequeña probabilidad) a pequeñas perturbaciones aleatorias a través del operador de mutación. A partir de una población inicial al azar o heurísticamente generada, este ciclo de renovación se repite para un número de iteraciones, y la mejor solución encontrada se devuelve al final. El rasgo distintivo del Algoritmo Genético permanece en la explotación de una población de soluciones y la creación de nuevas soluciones, a través de la recombinación de las buenas cualidades de dos soluciones de padres.<sup>55</sup>

- **Optimización basada en Colonias de Hormigas (ACO):** es un método de búsqueda local estocástico inspirado en el rastro de feromonas que dejan y siguen algunas especies de hormigas.<sup>56</sup> Esta técnica se inspira en la búsqueda de comida de las hormigas reales, el cual se describe como: Inicialmente, las hormigas exploran el área cercana a su nido de forma aleatoria; tan pronto como una hormiga encuentra comida, la lleva al nido. Mientras que realiza este camino, la hormiga va depositando una sustancia química denominada feromona. Esta sustancia ayudará al resto de las hormigas a encontrar la comida. La comunicación indirecta entre las hormigas mediante el rastro de feromonas las capacita para encontrar el camino más corto entre el nido y la comida. Este comportamiento es el que intenta simular este método para resolver problemas de optimización. La técnica se basa en dos pasos

---

<sup>55</sup> GENDREAU, Op.cit., p. 189-213.

<sup>56</sup> DORIGO, Marco y STÜTZLE, Thomas. Ant Colony Optimization. Estados Unidos: MIT Press, 2004. 321 p.

principales: la construcción de una solución basada en el comportamiento de una hormiga y la actualización de los rastros de feromonas artificiales. El algoritmo no fija ninguna planificación o sincronización a priori entre las fases, lo que permite que sean realizadas simultáneamente.

## 6.5 OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS

La Optimización por Enjambre de Partículas (PSO) se basa en la observación de grupos de individuos que trabajan juntos para mejorar su rendimiento colectivo e individual en alguna tarea. Esta técnica fue desarrollada por Kennedy y Eberhart en 1995, inspirada en un comportamiento de enjambre propio de la escolarización de los peces y las aves en la naturaleza. La idea inicial sobre el enjambre de partículas de Kennedy y Eberhart se orientó fundamentalmente a la producción de inteligencia computacional explotando análogos simples de interacción social, en lugar de las capacidades cognitivas individuales.<sup>57</sup> Cada individuo en el enjambre de partículas se compone de tres vectores con D-dimensiones (donde D es la dimensionalidad del espacio de búsqueda), los cuales son: la posición actual  $\vec{x}_i$ , la mejor solución encontrada hasta entonces  $\vec{P}_{best}$  y la velocidad  $\vec{v}_i$ .

La posición actual  $\vec{x}_i$ , puede ser considerada como un conjunto de coordenadas que describen un punto en el espacio. En cada iteración del algoritmo, la posición actual se evalúa como una solución del problema; si esa posición es mejor que cualquiera que se ha encontrado hasta el momento, entonces las coordenadas se almacenan en el segundo vector  $\vec{P}_{best}$  para la comparación en iteraciones posteriores.

---

<sup>57</sup> POLI, R., KENNEDY, J., & BLACKWELL, T. Particle swarm optimization. En: Swarm Intell [base de datos en línea] vol. 1 (Jun., 2007); p. 33-57. [citado en 19 de agosto de 2015] Disponible en SpringerLink Research Databases.

El objetivo, es mantener la búsqueda de las mejores posiciones y la actualización de  $\vec{P}_{best}$ . Los nuevos puntos se eligen a través de la adición de  $\vec{v}_i$  a las coordenadas de  $\vec{x}_i$ . El algoritmo opera mediante el ajuste de  $\vec{v}_i$ .<sup>58</sup>

En la Optimización por Enjambre de partículas, el sistema se inicializa con una población de soluciones y búsquedas aleatorias para óptimos, mediante la actualización de las generaciones en cada iteración; las soluciones posibles llamadas partículas, vuelan por el espacio del problema siguiendo las partículas óptimas actuales. En cada instante, cada partícula realiza un seguimiento de su coordenada en el espacio del problema, que se asocia con la mejor solución que ha logrado hasta entonces, conocido como  $P_{best}$ , y una mejor solución asociada a un valor obtenido por cualquier partícula del enjambre hasta ese momento, conocida como la mejor ubicación global, llamada  $P_{global}$ .<sup>59</sup>

Las partículas utilizan una combinación de conocimiento de sus mejores ubicaciones personales y la mejor ubicación global descubierta hasta el momento y una tendencia de moverse al azar, para determinar su curso en cada iteración. El objetivo es encontrar la mejor solución global entre todas las mejores soluciones actuales, hasta que el resultado ya no se mejore o después de un cierto número de iteraciones.<sup>60</sup>

La velocidad de los individuos en PSO puede cambiar debido a dos razones: la primera, porque las partículas recuerdan su mejor posición, lo que hace que cambien su velocidad para volver a esa posición (En PSO, una partícula viaja a

---

<sup>58</sup> Ibid., p. 33-57.

<sup>59</sup> SIMON, Dan. Evolutionary Optimization Algorithms. New Jersey, USA: John Wiley & Sons, 2013. 727 p.

<sup>60</sup> YANG, Xin She. Engineering Optimization: An Introduction with Metaheuristic Applications. New Jersey, USA: John Wiley & Sons, 2010. 343 p.

través del espacio de búsqueda, y su posición en el mismo, cambia de una iteración a la otra). La segunda, porque las partículas conocen la mejor posición de sus vecinos en la actual iteración, lo cual requiere de la definición de un tamaño de vecindario, y que todos los vecinos se comuniquen entre sí, acerca de su desempeño en el problema de optimización.

### 6.5.1 Algoritmo básico de PSO

**Tabla 1. Pseudocódigo PSO**

<b>OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS</b>
1: Inicializar una matriz población de partículas con posiciones aleatorias y velocidades en D- dimensiones en el espacio de búsqueda.
2: <b>Loop</b>
3: Para cada partícula, evaluar la función de optimización de la aptitud deseada en las variables D.
4: Comparar la evaluación aptitud para cada partícula con su <i><b>pbest<sub>i</sub></b></i> . Si el valor actual es mejor que <i><b>pbest<sub>i</sub></b></i> , a continuación, establezca <i><b>pbest<sub>i</sub></b></i> igual al valor actual, y <i><b>p<sub>i</sub></b></i> igual a la ubicación actual <i><b>x<sub>i</sub></b></i> en el espacio D-dimensional.
5: Identificar la partícula en el vecindario con el mayor éxito hasta el momento, y asignarle índice de variable <i><b>g</b></i> .
6: Cambiar la velocidad y la posición de la partícula de acuerdo con la siguiente ecuación:
$\vec{v}_i(t+1) = \vec{v}_i + (C_1 * Rand()) * (\vec{p}_i - \vec{x}_i) + (C_2 * Rand()) * (\vec{p}_g - \vec{x}_i) \quad (1)$ $\vec{x}_i(t+1) = \vec{x}_i + \vec{v}_i \quad (2)$
7: Si se cumple un criterio (mejor solución encontrada o un número máximo de iteraciones), exit Loop.
8: <b>end loop.</b>

**6.5.2 Parámetros :** El PSO básico descrito anteriormente tiene un número de parámetros fijo. Uno de los parámetros, es el tamaño de la población, que se establece empíricamente sobre la base de la dimensionalidad y la dificultad percibida del problema. Los valores en el rango de 20 a 50 son bastante comunes.

Los parámetros  $C_1$  y  $C_2$  en (1) determinan la magnitud de las fuerzas aleatorias en la dirección del mejor  $\vec{p}_i$  individual y  $\vec{p}_g$  global, llamados coeficientes de aceleración. El comportamiento del PSO cambia radicalmente con el valor de  $C_1$  y  $C_2$ . La interpretación establecida para los componentes  $(C_1 * Rand()) * (\vec{p}_i - \vec{x}_i)$  y  $(C_2 * Rand()) * (\vec{p}_g - \vec{x}_i)$  en (1) está dada como fuerzas de atracción producidas por fuentes de rigidez al azar, este movimiento se puede asumir como la integración de la segunda ley de Newton. Además, en esta interpretación,  $\frac{C_1}{2}$  y  $\frac{C_2}{2}$  representan la rigidez media de las fuerzas que tiran de una partícula, por lo cual si se llega a cambiar  $\phi_1$  y  $\phi_2$  se puede hacer que el PSO varíe su eficiencia e inclusive se vuelva inestable debido a velocidades de partículas crecientes sin control.

El valor  $C_1 = C_2$  que se adoptaba en las primeras investigaciones del PSO era dos, sin embargo, este valor a menudo perjudicaba la búsqueda y necesitaba ser controlado.

La técnica propuesta originalmente para controlar este valor, consiste en limitar las velocidades para que cada componente de  $\vec{v}_i$  se encuentre dentro del rango  $[-v_{max}, +v_{max}]$ . La elección del parámetro  $v_{max}$  requiere cierto cuidado ya que influye en el equilibrio entre la exploración y explotación. No obstante, el uso de límites en la velocidad, presenta algunos problemas, debido a que no hay una regla específica para establecer un valor óptimo de  $v_{max}$ .

**6.5.2.1 Peso Inercial:** Para controlar mejor el alcance de la búsqueda y reducir la importancia de  $v_{max}$ , o eliminarla por completo, Shi y Eberhart en 1998, presentan la siguiente modificación a la ecuación de velocidad:

$$\vec{v}_i(t+1) = \omega * \vec{v}_i + (C_1 * Rand()) * (\vec{p}_i - \vec{x}_i) + (C_2 * Rand()) * (\vec{p}_g - \vec{x}_i) \quad (3)$$

Donde el término  $\omega$  se define como el peso inercial.

Entonces, si se interpreta  $(C_1 * Rand()) * (\vec{p}_i - \vec{x}_i) + (C_2 * Rand()) * (\vec{p}_g - \vec{x}_i)$  como la fuerza externa  $\vec{f}_i$ , que actúa sobre una partícula, el cambio en la velocidad de una partícula se escribe como:

$$\Delta \vec{v}_i = \vec{f}_i - (1 - \omega) \vec{v}_i \quad (4)$$

La constante  $1 - \omega$  actúa como un coeficiente de fricción efectivo, por lo que  $\omega$  se puede interpretar como la fluidez del medio en el que una partícula se mueve. Al principio los investigadores observaban que el mejor rendimiento se encontraba mediante el establecimiento de un valor inicial para  $\omega$  relativamente alto (por ejemplo, 0.9), lo que corresponde a un sistema en el que las partículas se mueven en un medio de baja viscosidad y realizan extensa exploración, para después reducir gradualmente el valor de  $\omega$  a un valor mucho más bajo (por ejemplo, 0.4), donde el sistema sería más disipativo y explotador, profundizando la búsqueda de óptimos locales. El valor preciso de  $\omega$  que garantiza la estabilidad del sistema depende de los valores de los coeficientes de aceleración.

### 6.5.3 PSO con coeficientes de constricción

$$\vec{v}_i(t+1) = \chi(\vec{v}_i + (C_1 * Rand()) * (\vec{p}_i - \vec{x}_i) + (C_2 * Rand()) * (\vec{p}_g - \vec{x}_i)) \quad (5)$$

$$\vec{x}_i(t+1) = \vec{x}_i + \vec{v}_i \quad (6)$$

Donde  $C = C_1 + C_2 > 4$ .

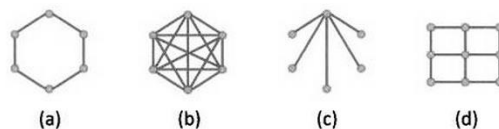
Y la forma de incorporarlo según Clerc y Kennedy (2002) es:

$$\chi = \frac{2}{C - 2 + \sqrt{C^2 - 4C}} \quad (7)$$

Estos coeficientes controlan la convergencia de la partícula y le brindan al sistema un método de prevención de la explosión de la velocidad, que garantiza la convergencia y elimina la arbitrariedad del parámetro  $v_{max}$ . El análisis también elimina las conjeturas de establecer los valores de  $C_1$  y  $C_2$ .<sup>61</sup>

**6.5.4 Topología Poblacional :** Los primeros enjambres de partículas (Kennedy y Eberhart 1995) evolucionaron a partir de simulaciones de aves descritas por Reynolds en 1987 y Heppner y Grenander en 1990, en los cuales la trayectoria de vuelo de cada ave se modifica mediante la aplicación de varias normas, como la proximidad de otras aves en el espacio físico. Las primeras topologías del PSO se basaron en la proximidad en el espacio de búsqueda, sin embargo, además de ser computacionalmente intensivo, este tipo de estructura de comunicación tenía propiedades de convergencia indeseables, por lo que pronto fue abandonada.

**Figura 3. Algunas Topologías del PSO**



Fuente. Engineering Optimization: An Introduction with Metaheuristic Applications.<sup>62</sup>

---

<sup>61</sup> Ibid., p.33-57.

<sup>62</sup> Ibid., 727 p.

La figura muestra en (a) la topología de anillo, en (b) la topología global, en (c) la topología de rueda y en (d) la topología en cuadrado. La topología en cuadrado se envuelve alrededor de la parte superior hacia la parte inferior, y desde la izquierda a la derecha, de manera que forma un toroide con cada partícula conectada a cuatro vecinos. Cada una de estas topologías puede ser estática o dinámica. A su vez, esta figura muestra que cada partícula es influenciada por sus vecinos más cercanos. La disposición de los vecinos que influyen en una partícula es lo que se conoce como topología del enjambre. El rendimiento del PSO puede variar fuertemente según la topología. Cuando la vecindad de cada partícula cambia en cada iteración, se le conoce como topología dinámica, de lo contrario se dice que es estática.

**Topología global:** Es donde el mejor vecino en toda la población influye en la partícula objetivo, puede ser conceptualizado como un gráfico completamente conectado, lo que en la práctica significa que el programa necesita no perder de vista el mejor resultado de la función encontrado y el índice de la partícula que lo encontró. El gbest es un ejemplo de topología estática, es decir, uno donde los vecinos y los vecindarios no cambian durante una iteración.

**Topología Thering:** Cada partícula está conectada con otras dos partículas.

**Topología de clúster:** Cada partícula está completamente conectada dentro de su propio clúster, mientras que unas pocas partículas en cada grupo también están conectadas a una partícula adicional en otro clúster.

**Topología de rueda:** Una partícula focal está conectada a todas las demás partículas, mientras que todas las otras partículas están conectadas solamente a la partícula focal.

**Topología cuadrada:** También llamada la topología de von Neumann, es en la que cada partícula está conectada a cuatro vecinos.<sup>63</sup>

## 6.6 GRASP

Greedy Randomized Adaptative Search Procedure (GRASP) es un método de búsqueda de dos fases iterativo, que se ha vuelto muy popular en la optimización combinatoria. Cada iteración consta de dos fases, una fase de construcción y una fase de búsqueda local. El resultado obtenido al final de estas dos fases es simplemente la mejor solución encontrada sobre todas las iteraciones.

En la fase de construcción, una función codiciosa aleatoria se utiliza para construir las soluciones iniciales; dicha técnica aleatoria proporciona una solución factible dentro de cada iteración. Esta fase puede ser descrita como un proceso por etapas, que añade un elemento a la vez a la solución parcial. La elección del siguiente elemento para añadir se determina ordenando todos los elementos en una lista de candidatos con respecto a una función codiciosa.

La estrategia es adaptativa porque los beneficios asociados con cada elemento se actualizan durante cada iteración de la fase de construcción, para reflejar los cambios provocados por la selección del elemento anterior. El componente probabilístico de un GRASP se caracteriza por la elección al azar de uno de los mejores candidatos en la lista, pero no necesariamente el mejor candidato.

En la segunda fase, una búsqueda local se inicializa a partir de los puntos arrojados en la fase de construcción, y el resultado final es simplemente la mejor solución encontrada sobre todas las búsquedas.

---

<sup>63</sup> Ibid., 727 p.

## 6.7 MULTIPLE PHASE NEIGHBORHOOD SEARCH - GRASP

El algoritmo GRASP ha demostrado ser muy eficaz para la solución del Problema del Agente Viajero. En la mayoría de sus implementaciones utiliza algún tipo de lista restringida de candidatos (RCL), en la que un parámetro RCL determina el nivel de codicia o aleatoriedad en la construcción. Sin embargo, en el caso del MPNS-GRASP, este parámetro no se utiliza y los mejores bordes candidatos son los seleccionados para formar la RCL. Posteriormente, uno de ellos es elegido al azar como próximo candidato para incluirse en la ruta.

El MPNS-GRASP introduce la flexibilidad de la aplicación de funciones codiciosas alternativas en cada iteración, en vez de una sola función codiciosa simple como en el enfoque clásico. El algoritmo comienza con una función codiciosa y si los resultados no mejoran o conducen a circunstancias patógenas (por ejemplo, en el caso de TSP una circunstancia patógena ocurre cuando un número de muy buenos arcos se quedan fuera de la solución), una función codiciosa alternativa se utiliza en su lugar.

La utilización de una búsqueda local simple en la segunda fase del algoritmo clásico, limita las posibilidades de obtener mejores soluciones. Por lo tanto, MPNS-GRASP se alía con la Expansión del Vecindario de Búsqueda (ENS) que es una estrategia de búsqueda local mucho más flexible.

En consecuencia, el MPNS-GRASP se convierte en un algoritmo muy flexible, potente y adaptable. Casi todas las implementaciones GRASP utilizan un criterio de terminación basado en el número máximo permitido de iteraciones.<sup>64</sup>

---

<sup>64</sup> MARINAKIS, Y; MIGDALAS, A y PARDALOS, P. Multiple phase neighborhood Search—GRASP based on Lagrangean relaxation, random backtracking Lin–Kernighan and path relinking for the TSP. En: Journal of Combinatorial Optimization [base de datos en línea]. Vol. 17, No. 2 (Feb. 2009); p.134-156. [citado en 20 de Agosto de 2015] Disponible en SpringerLink Research Databases.

### 6.7.1 Pasos del Algoritmo MPNS-GRASP

#### Inicialización

- (1) Seleccionar el conjunto de algoritmos codiciosos que se utilizará en la primera fase del MPNS-GRASP.
- (2) Seleccionar el conjunto de algoritmos de búsqueda local que se utilizará en la segunda fase del MPNS-GRASP.
- (3) Seleccionar el tamaño inicial de la RCL.

#### Algoritmo Principal

- (1) Establecer el número de iteraciones igual a cero.
- (2) *Hacer mientras* los criterios de detención no estén satisfechos (el número máximo de iteraciones no se ha alcanzado):

#### *Fase Principal 1*

- 2.1. Aumentar el contador de iteraciones.
- 2.2. Construir la RCL.
- 2.3. Seleccionar al azar de la RCL el elemento candidato para su inclusión a la solución parcial.
- 2.4. Llamar a un algoritmo codicioso.
- 2.5. Llamar al segundo algoritmo codicioso.

#### *Fase Principal 2*

- 2.6 Llamar a la ampliación del vecindario de búsqueda (ENS).
- (3) *Terminar Hacer*
- (4) Retornar la mejor solución.<sup>65</sup>

---

<sup>65</sup> Ibid., p. 134-156.

## 6.8 ENS (Expanding Neighborhood Search)

Expanding Neighborhood Search es un algoritmo metaheurístico que se puede usar para solucionar una serie de problemas de optimización combinatoria, con resultados notablemente buenos.

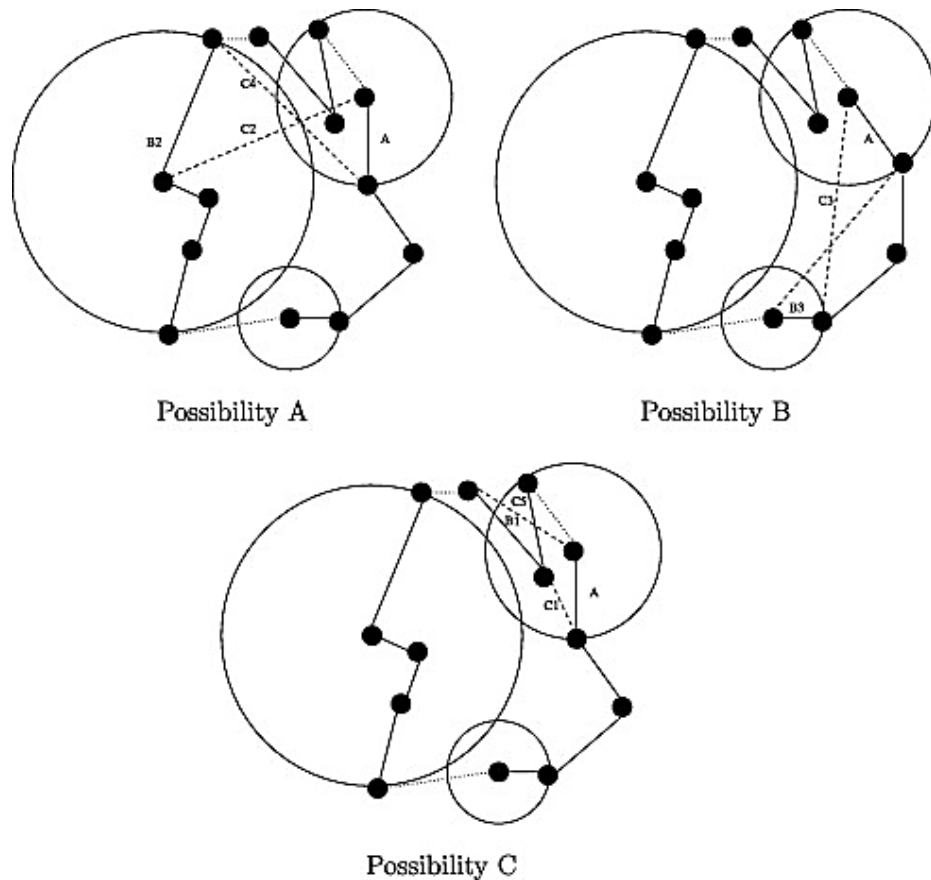
Las principales características de este algoritmo son:

- ∴ Uso de la estrategia de Movimientos de búsqueda local en círculos restringidos (CRLSM).
- ∴ Capacidad del algoritmo para cambiar entre las diferentes estrategias de búsqueda local.
- ∴ Uso de una estrategia de expansión.

La aplicación de la estrategia de Movimientos de búsqueda local en círculos restringidos (CRLSM), hace que el tiempo de cálculo se reduzca significativamente en comparación con otros algoritmos heurísticos y metaheurísticos, porque todas las aristas que no van a mejorar la solución, están excluidas del procedimiento de búsqueda. Esto se logra, restringiendo el espacio de búsqueda en círculos alrededor de los bordes candidatos a ser eliminados.

A continuación, una descripción de la estrategia CRLSM para un movimiento *2-opt* es presentada (Lin, 1965). En este caso, hay tres posibles intercambios, teniendo en cuenta los costos de los bordes.

**Figura 4. Estrategia de Movimientos de búsqueda local en círculos restringidos (CRLSM)**



Fuente. A hybrid particle swarm optimization algorithm for the vehicle routing problem.<sup>66</sup>

En la figura 4, las líneas discontinuas son los nuevos bordes, las líneas de puntos indican que hay una posibilidad de tener más nodos entre los dos nodos conectados

---

<sup>66</sup> MARINAKIS, Y., MARINAKI, M. y DOUNIAS, G. A hybrid particle swarm optimization algorithm for the vehicle routing problem. En: Engineering Applications of Artificial Intelligence [base de datos en línea]. Vol. 23, no. 4 (Jun. 2010); p. 463-472, ISSN 0952-1976 [citado en 22 de Agosto de 2015] Disponible en ScienceDirect Research Databases.

con estas líneas, y hay tres círculos en cada una de las sub figuras, con el fin de centrarse en las diferencias de las tres posibilidades.

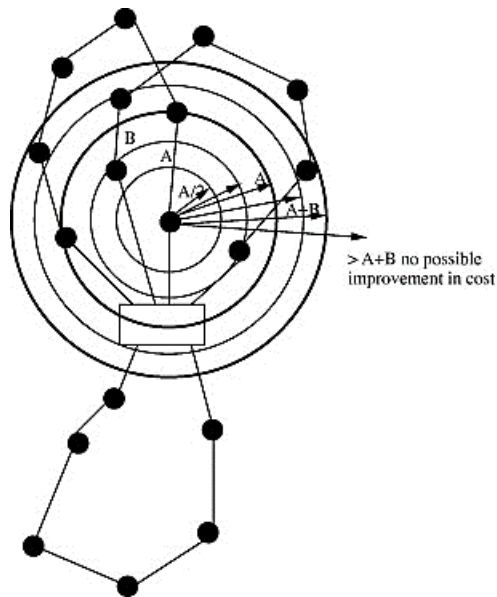
- ∴ Si ambos nuevos bordes aumentan el costo, un movimiento 2-opt no puede reducir el costo de la gira (por ejemplo, en la figura 4 (Posibilidad A), ambos nuevos bordes, tanto  $C2$  y  $C4$  son mayores que los costos  $B2$  y  $A$ ).
- ∴ Si uno de los dos nuevos bordes tiene un costo mayor que la suma de los costos de los dos bordes antiguos, un movimiento 2-opt trial no puede reducir el costo de la gira (por ejemplo, en la figura 4 (Posibilidad B) el costo del nuevo borde  $C3$  es mayor que la suma de los costos  $A$  y  $B3$ ).
- ∴ El único caso en el que un movimiento 2-opt puede reducir el costo de la ruta, es cuando al menos un borde nuevo representa un menor valor que el costo de uno de los dos bordes antiguos (por ejemplo, en la figura 4 (Posibilidad C), el costo  $C1$  del nuevo borde es menor que el costo del borde antiguo  $A$ ) y el otro borde tiene un costo menor que la suma de los costos de los dos bordes antiguos (por ejemplo,  $C5 < A + B1$  en la figura 4 (Posibilidad C)).

Teniendo en cuenta estas observaciones, la estrategia de Movimientos de Búsqueda Local en Círculos Restringidos, limita la búsqueda a los bordes donde uno de sus nodos finales está dentro de un círculo con un radio de longitud, como máximo, igual a la suma de las longitudes de los dos bordes candidatos a ser eliminados.

El procedimiento de ENS funciona de la siguiente manera: Inicialmente se selecciona un borde de la solución actual (por ejemplo, el borde con la peor longitud) y se aplica la estrategia de búsqueda local, con el objetivo de mejorar la solución. Si la solución mejora, se actualiza, se selecciona un nuevo borde y la nueva iteración de la Estrategia de Ampliación del vecindario de búsqueda comienza; por

otra parte, si no se encuentra una solución mejor, el círculo se expande y la estrategia de búsqueda local se aplica en el nuevo círculo hasta que se encuentre una mejor solución o el círculo alcance el máximo radio posible. Si se ha alcanzado el máximo radio posible, entonces un nuevo borde se selecciona (Figura 5).

**Figura 5. Estrategia de Ampliación del Vecindario de búsqueda**



Fuente. A hybrid particle swarm optimization algorithm for the vehicle routing problem.<sup>67</sup>

Inicialmente, el tamaño de la vecindad,  $s$ , se define con base en la estrategia del círculo restringido, por ejemplo,  $s = A / 2$ , donde  $A$  es el costo (longitud) de uno de los bordes candidatos a ser eliminados. Para el tamaño seleccionado de la vecindad, se aplica la estrategia de búsqueda local hasta que todos los posibles

---

<sup>67</sup> Ibid., p. 463-472.

movimientos de prueba se exploren. Si la solución no mejora en este vecindario, este se expande en un porcentaje  $\theta$  (por ejemplo,  $\theta = 10\%$ ). Si la longitud del radio de la estrategia CRLSM es igual a  $A$ , y ningún criterio de parada se ha activado, entonces el algoritmo termina con la solución actual.<sup>68</sup>

## 6.9 PATH RELINKING

La idea fundamental detrás de este método es que las buenas soluciones para un problema deberían compartir algunas características, mediante la generación de caminos (es decir, secuencias de soluciones intermedias) entre soluciones elite. Path Relinking puede ser interpretado como un método donde las soluciones son generadas por la combinación de elementos de otras soluciones.

Para generar los caminos deseados, una solución inicial y una solución guía son escogidas de un conjunto de soluciones elites de referencia, para representar los puntos de inicio y fin del camino. Los atributos de la solución guía son gradualmente introducidos a soluciones intermedias, de esta manera, estas soluciones contienen menos características de la solución inicial y más de la solución guía, con unos movimientos a lo largo del camino.<sup>69</sup>

Todas las implementaciones de Path Relinking giran alrededor de los siguientes 3 componentes, que son críticos en el diseño del algoritmo:

- ∴ Reglas para construir el conjunto de referencias.

---

<sup>68</sup> Ibid., p. 463-472.

<sup>69</sup> GENDREAU, M. y HO, S. Path Relinking for the vehicle routing problem. En: Journal of Heuristics [base de datos en línea]. Vol.12 (2006); p. 55-72. [citado en 22 de Agosto de 2015] Disponible en SpringerLink Research Databases.

∴ Reglas para escoger la solución inicial y la solución guía.

∴ Estructura de vecindario para moverse entre caminos.

A continuación, se detalla el pseudocódigo del procedimiento de Path Relinking a un par de soluciones  $X_s$  (solución inicial) y  $X_t$  (Solución Objetivo)<sup>70</sup>:

*Datos: Solución Inicial  $X_s$  y Solución Objetivo  $X_t$*

*Resultado: Mejor Solución  $X^*$  en el camino desde  $X_s$  hasta  $X_t$ .*

*Calcular diferencia simétrica  $\Delta(X_s, X_t)$ ;*

$f^* \leftarrow \min\{f(X_s), f(X_t)\};$

$X^* \leftarrow \operatorname{argmin}\{f(X_s), f(X_t)\};$

$X \leftarrow X_s;$

***Mientras  $\Delta(X, X_t) \neq 0$  Hacer***

$m^* \leftarrow \operatorname{argmin}\{f(X \oplus m) : m \in \Delta(X, X_t)\};$

$\Delta(X \oplus m^*, X_t) \leftarrow \Delta(X, X_t) \setminus \{m^*\};$

$X \leftarrow X \oplus m^*;$

***Si  $f(X) < f^*$  entonces***

$f^* \leftarrow f(X);$

$X^* \leftarrow X;$

*fin*

*fin*

---

<sup>70</sup> RESENDE, Mauricio y RIBEIRO, Celso. GRASP with Path Relinking: Recent advances and applications. En: Operations Research/Computer Science Interfaces Series [base de datos en línea]. Vol. 32 (2005); p. 29-63. [citado en 20 de Agosto de 2015] Disponible en SpringerLink Research Databases.

## 6.10 VEHICLE ROUTING PROBLEM (VRP)

El VRP es un problema de optimización combinatoria que pretende servir a un número determinado de clientes con una flota de vehículos; es un problema crítico en las áreas de transporte, distribución y logística.<sup>71</sup>

El fundamento del problema de ruteo de vehículos está en el Problema del Agente Viajero, TSP, por sus siglas en inglés. En este problema existe un agente de ventas que debe visitar a sus clientes ubicados en diferentes ciudades y luego volver a su ciudad de partida. Dicha actividad debe ser llevada a cabo con el menor costo posible.<sup>72</sup>

El VRP se puede definir como:

Sea  $G = (V, A)$  un grafo completo donde  $V = \{0, 1, \dots, n\}$  es el conjunto de vértices y  $A$  es el conjunto de arcos. Los vértices  $j = 1, \dots, n$  corresponden a los clientes, cada uno con una demanda no negativa  $d_j$  conocida y el vértice 0 corresponde al depósito. Un costo no negativo,  $C_{ij}$ , se asocia con cada arco  $(i, j)$  y representa el costo de viajar desde el vértice  $i$  al vértice  $j$ .

El VRP general o clásico consiste en diseñar un conjunto de  $K$  rutas para la entrega o recogida de bienes, de tal manera que cada ruta comience y termine en el depósito, cada cliente sea visitado exactamente una vez por exactamente un vehículo, la demanda total de cada ruta no exceda la capacidad  $C$  del vehículo y el

---

<sup>71</sup> MOGHADDAM, Op.cit., p. 306-317.

<sup>72</sup> GONZÁLEZ VARGAS, Guillermo, y GONZÁLEZ ARISTIZÁBAL, Felipe. Metaheurísticas aplicadas al ruteo de vehículos. Un caso de estudio: Parte 1: formulación del problema. En: Ingeniería e Investigación. Vol. 26, no. 3 (2006); p. 149-156.

costo total del ruteo se minimice. Partiendo de estos supuestos se formula el VRP como:<sup>73</sup>

$$\min \sum_{i \in V} \sum_{j \in V} C_{ij} X_{ij}$$

*Sujeto a:*

$$\sum_{j \in V} X_{ij} = 1 \quad \forall i \in V / \{0\}$$

$$\sum_{i \in V} X_{i0} = K$$

$$\sum_{j \in V} X_{0j} = K$$

$$\sum_{i \notin S} \sum_{j \in S} X_{ij} \geq r(S) \quad \forall S \subseteq V / \{0\}, S \neq \emptyset$$

$$X_{ij} = \{0,1\} \quad \forall i, j \in V$$

Donde:

$C_{ij}$  = Costo o Distancia de viaje desde  $i$  hasta  $j$ .

$r(S)$  = Mínimo número de vehículos necesarios para servir a todos los clientes.

$X_{ij} = 1$  si el arco  $(i, j) \in A$  a lo largo de la solución óptima, 0 de lo contrario.

$S \subseteq V$  = Conjunto de clientes.

La inclusión de diversos escenarios de la vida real al VRP dio lugar a muchas variantes que evolucionaron desde el modelo general, descrito anteriormente.

---

<sup>73</sup> DANESHZAND, Farzaneh. The Vehicle-Routing Problem, In Logistics Operations and Management. Londres: Reza Zanjirani FarahaniShabnam RezapourLaleh Kardar, (2011). p. 127-153.

Los componentes fundamentales del VRP son las coordenadas de los clientes, los clientes, los depósitos, los vehículos y los conductores. Para hacer diferentes versiones del VRP, diferentes limitaciones y situaciones pueden ser impuestas a cada componente, y cada uno de ellos se puede suponer para lograr objetivos particulares.<sup>74</sup>

### 6.10.1 Variantes básicas del VRP

- ∴ **CVRP (Capacited Vehicle Routing Problem):** La versión básica del VRP es CVRP. En este problema, cada vehículo tiene una capacidad conocida de antemano que no le permite al vehículo cargar más de su capacidad. Hay dos versiones de CVRP: ACVRP, cuando la matriz de costo es asimétrica, y SCVRP, cuando la matriz de costo es simétrica.
  
- ∴ **DCVRP (Distance Constrained and Capacited Vehicle Routing Problem):** El DCVRP es una variante del CVRP en el que se imponen limitaciones tanto en la capacidad del vehículo como en la distancia máxima. En este tipo de problemas, cada longitud de recorrido no debe ser superior a una cantidad previamente conocida.
  
- ∴ **VRPTW ( Vehicle Routing Problem with Time Windows):** El VRPTW es la extensión del CVRP donde el servicio a cada cliente debe comenzar dentro de una ventana de tiempo específica y el vehículo debe permanecer en la ubicación del cliente durante el servicio.

---

<sup>74</sup> TOTH, P. y VIGO, D. The vehicle routing problem. Philadelphia: SIAM Monographs on Discrete Mathematics and Applications, 2002. 363 p.

- ∴ **VRPB (Vehicle Routing Problem with Backhauls):** En el VRPB (VRP con viajes de regreso) los clientes pueden exigir o devolver algunos productos básicos. De hecho, es una extensión del CVRP en el que los clientes se dividen en dos subconjuntos: línea de recorrido y línea de regreso de recorrido. Habrán clientes que requieren que se les entregue una cantidad de producto, así como clientes que requieren que se les recoja cierta cantidad de producto.
  
- ∴ **VRPPD (Vehicle Routing Problem with Pickup and Delivery):** En el VRPPD, los vehículos tienen dos conjuntos de tareas, la entrega de bienes a los clientes, así como la recogida de bienes en las instalaciones del cliente.

En este, una flota de vehículos heterogéneos debe satisfacer un conjunto de solicitudes de transporte. Cada solicitud se define por un punto de recogida, un punto de entrega correspondiente, y una demanda a ser transportada entre estos lugares.<sup>75</sup>

### 6.10.2 Otras variantes del VRP

- ∴ **OVRP (Open Vehicle Routing Problem):** En el OVRP, los vehículos no están obligados a regresar a la estación. Este tipo de problema se presenta para las empresas que no son propietarias de una flota de vehículos o que la flota de vehículos es inadecuada para la demanda de todos los clientes. Por lo tanto, la empresa está obligada a contratar la totalidad o parte de la distribución de productos a operadores externos. Los vehículos de alquiler, serán asignados a las rutas y no tendrán que volver al centro de distribución de la compañía.

---

<sup>75</sup> DANESHZAND, Op.cit., p. 127-153.

La solución del problema proporcionará a la empresa, el número mínimo de vehículos que deben ser contratados con el fin de servir a los clientes y el conjunto de rutas que minimicen el costo de viaje. Además, en la situación en la que la empresa tenga su propia flota de vehículos y la demanda de los clientes varíe significativamente con el tiempo, la solución proporcionará la combinación adecuada de vehículos propios y contratados.

- ∴ **MDVRP (Multiple Depots Vehicle Routing Problem):** En el VRP clásico, sólo hay una estación y todos los vehículos inician y terminan sus rutas en ese depósito. En el MDVRP, existe más de un depósito. En este problema, cada cliente recibe la visita de un vehículo proveniente de alguno de los depósitos.

El MDVRP puede ser visto como un problema de agrupamiento, en el sentido de que la salida es un conjunto de horarios de vehículos agrupados por depósito. Por lo tanto, el MDVRP puede ser resuelto en dos etapas: en primer lugar, los clientes deben ser asignados a los depósitos y, en segundo lugar, los clientes asignados al mismo depósito deben ser unidos entre sí a través de rutas. Lo ideal es hacer frente a los dos pasos simultáneamente, sin embargo, un enfoque razonable sería dividir el problema en tantos sub problemas como número de depósitos.

- ∴ **MFVRP (Mix Fleet Vehicle Routing Problem):** El MFVRP difiere del VRP clásico, en que se ocupa de una flota heterogénea de vehículos, que tienen diversas capacidades, y costos fijos y variables. El costo del ruteo es la suma de los costos fijos y variables, en el que los costos variables son proporcionales a la distancia de viaje.
- ∴ **SDVRP (Split-Delivery Vehicle Routing Problem):** Es una relajación del VRP en el que el mismo cliente puede ser atendido por diferentes vehículos si esto

reduce los costos totales. Las entregas divididas resultan en ahorros, tanto en la distancia total recorrida y el número de vehículos utilizados.

- ∴ **PVRP (Periodic Vehicle Routing Problem):** En el PVRP, un conjunto de clientes tiene que ser visitado en un horizonte de tiempo dado, una o más veces. Diferentes clientes usualmente requieren diferentes números de visitas en cierto horizonte de tiempo. PVRP-SC es una variante del PVRP en el cual la frecuencia de visitas a los nodos, es una decisión variable del modelo. La solución del problema requiere la asignación de un régimen de visitas a cada cliente. Por cada día del horizonte temporal, las rutas de los vehículos deben definirse de tal manera que todos los clientes a los que en su cronograma se incluya ese día, sean servidos.<sup>76</sup>

**6.10.3 Stochastic Vehicle Routing Problem:** El ruteo de vehículos estocástico (SVRP) se caracteriza por tener algún elemento en el problema que es aleatorio. Los VRP estocásticos se diferencian de su versión determinista en varios aspectos fundamentales, el concepto de obtención de una solución es diferente, debido a que varias propiedades fundamentales del VRP determinista ya no se tienen en cuenta para el caso estocástico, y las metodologías de solución son considerablemente más complejas, ya que combinan las características de los programas estocásticos y enteros, clasificándolos como computacionalmente intratables. Las tres variantes más comunes para el SVRP son: VRP con demandas estocásticas (VRPSD), VRP con clientes estocásticos (VRPSC), y VRP con tiempos de viaje estocásticos (VRPST).<sup>77</sup>

---

<sup>76</sup> Ibid., p. 127-153.

<sup>77</sup> GENDREAU, Michel; LAPORTE, Gilbert y SÉGUIN, René. Stochastic vehicle routing. En: European Journal of Operational Research [base de datos en línea]. Vol. 88, no. 1 (Ene. 1996); p. 3-

**6.10.4 VRP with Stochastic Travel Times and Soft Time Windows:** El problema de ruteo de vehículos con tiempos de viaje estocásticos y ventanas de tiempo suave pertenece a la clase de problemas de optimización combinatoria NP-hard. En este problema se considera el tiempo de viaje del vehículo como una variable aleatoria, con distribución de probabilidad conocida. El tiempo solo se conoce en el momento en el que el vehículo llega al cliente; adicional a esto el cliente proporciona al distribuidor un intervalo de tiempo para ser servido.<sup>78</sup>

Esta formulación, divide los componentes del costo en: Costos de transporte y costos del servicio. La distinción entre estos dos costos es de gran importancia en aplicaciones de la vida real, ya que las empresas de transporte muchas veces centran sus esfuerzos en aspectos que no son relevantes para sus clientes.

En la práctica, los proveedores de servicios tienen como objetivo servir a todos sus clientes mediante el empleo de la menor cantidad de rutas y vehículos posibles (costos operativos) y además brindarles confiabilidad al ser servidos por estas rutas (dentro de las ventanas de tiempo preestablecidas), de manera que se sientan satisfechos con el servicio recibido. El componente del costo de transporte afecta directamente en los costos reales pagados por las empresas de transporte, y además está directamente relacionado con la eficiencia de las operaciones. El costo de servicio se obtiene mediante la confrontación de los tiempos de llegada reales con los establecidos en las ventanas de tiempo y por lo tanto directamente asociados con la fiabilidad en las operaciones.<sup>79</sup>

---

12, ISSN 0377-2217. [citado en 23 de agosto de 2015] Disponible en ScienceDirect Research Databases.

<sup>78</sup> TAŞ, Op.cit., p. 789-799.

<sup>79</sup> TAŞ, Op.cit., p. 789-799.

Es por esto, que esta consideración permite obtener rutas que sean tanto eficientes como confiables.<sup>80</sup>

**6.10.4.1 Objetivo del VRPSTTW:** En esta formulación, el objetivo es minimizar la suma de los costos de transporte y los costos del servicio. Los costos de transporte son el resultado de dos elementos que son, la distancia total recorrida y el número de vehículos. El costo de servicio se causa por llegadas tempranas y tardías, que son resultado de las violaciones a las ventanas de tiempo establecidas por los clientes.<sup>81</sup>

**6.10.4.2 Formulación del Problema:** Sea  $G = (N, A)$  un dígrafo conectado, que denota la red donde  $N = \{0, 1, \dots, n\}$  es el conjunto de nodos y  $A = \{(i, j) | i, j \in N, i \neq j\}$  es el conjunto de arcos. El depósito está representado por el nodo 0 y cada nodo en  $N \setminus \{0\}$  corresponde a un cliente distinto. Cada cliente tiene una demanda conocida ( $q_i \geq 0$ ), una duración establecida del servicio ( $s_i \geq 0$ ), y una ventana de tiempo suave ( $[l_i, u_i], l_i \geq 0, u_i \geq 0$ ). La ventana de tiempo en el depósito,  $[l_0, u_0]$ , corresponde al horizonte de programación. Se supone que el servicio brindado a los clientes puede comenzar antes o después de las ventanas de tiempo, considerando que:

- ∴ Si un vehículo llega temprano a un cliente, no se considera la opción de esperar hasta que la ventana de tiempo se abra.
- ∴ Si un cliente se sirve fuera de su ventana de tiempo, entonces la empresa incurre en sanciones por el servicio temprano o tarde.

---

<sup>80</sup> TAŞ, Op.cit., p. 214-224.

<sup>81</sup> TAŞ, Op.cit., p. 66-83.

Cada arco  $(i, j) \in A$  tiene un peso  $d_{ij}$  que corresponde a la distancia de ese arco. Además, se supone que la función de distribución de probabilidad del tiempo de viaje para cada arco  $(i, j)$  es conocida. La ubicación base de los vehículos es el depósito y cada vehículo  $v \in V$  posee la misma capacidad  $Q$ .

El objetivo es la construcción de un conjunto de rutas que operen a un costo total mínimo, mediante el cumplimiento de los siguientes requisitos:

- ∴ Cada ruta es operada por un solo vehículo y cada cliente es atendido una sola vez por un solo vehículo.
- ∴ Cada ruta debe comenzar y terminar en el depósito.
- ∴ La demanda total de los clientes asignados a una ruta no puede exceder la capacidad del vehículo.

Las notaciones utilizadas en la formulación matemática del problema son: La variable de decisión  $x_{ijv}$  toma el valor 1 si el arco  $(i, j)$  está cubierto por vehículo  $v$ , y toma el valor de cero en caso contrario. El vector  $\mathbf{x}$ , donde  $\mathbf{x} = \{x_{ijv} | ij \in N, v \in V\}$  se utiliza para indicar las asignaciones de los vehículos y las secuencias de los clientes en estas asignaciones (rutas de vehículos). Se tienen dos funciones en el componente del costo de servicio  $D_{jv}(\mathbf{x})$  y  $E_{jv}(\mathbf{x})$ , la demora y la antelación esperada respectivamente en el nodo  $j$  en caso de ser visitado por el vehículo  $v$ .  $O_v(\mathbf{x})$  es el tiempo extra esperado (en horas) del conductor que trabaja en la ruta del vehículo  $v$ . Los cálculos de los valores esperados están directamente relacionados con las decisiones de enrutamiento.

Los coeficientes  $C_d$  y  $C_e$ , se pueden considerar como los costos en los que la empresa incurre por una unidad de demora y una unidad de antelación,

respectivamente, estos coeficientes son necesarios para balancear el servicio tarde y temprano. Los parámetros  $C_t$  y  $C_o$ , son los costos que la empresa transportista tiene que pagar por una unidad de distancia recorrida y de horas extras, respectivamente. Por otra parte, el costo fijo  $C_f$  se refiere al costo que la empresa paga por cada vehículo usado.

La formulación matemática es:

$$\begin{aligned} \min \quad & \left[ \rho \frac{1}{C_1} (C_d \sum_{j \in N} \sum_{v \in V} D_{jv}(\mathbf{x}) + C_e \sum_{j \in N} \sum_{v \in V} E_{jv}(\mathbf{x})) + \right. \\ & \left. (1 - \rho) \frac{1}{C_2} (C_t \sum_{i \in N} \sum_{j \in N} \sum_{v \in V} d_{ij} x_{ijv} + C_f \sum_{j \in N \setminus \{0\}} \sum_{v \in V} x_{0jv} + C_o \sum_{v \in V} O_v(\mathbf{x})) \right] \quad (8) \end{aligned}$$

Sujeto a:

$$\sum_{i \in N} x_{ikv} - \sum_{i \in N} x_{kiv} = 0, \quad k \in N \setminus \{0\}, \quad v \in V \quad (9)$$

$$\sum_{j \in N} \sum_{v \in V} x_{ijv} = 1, \quad i \in N \setminus \{0\} \quad (10)$$

$$\sum_{j \in N} x_{0jv} = 1, \quad v \in V, \quad (11)$$

$$\sum_{i \in N} x_{i0v} = 1, \quad v \in V, \quad (12)$$

$$\sum_{i \in N \setminus \{0\}} q_i \sum_{j \in N} x_{ijv} \leq Q, \quad v \in V \quad (13)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijv} \leq |S| - 1, \quad S \subseteq N \setminus \{0\}, \quad v \in V \quad (14)$$

$$x_{ijv} \in \{0,1\}, \quad i \in N, \quad j \in N, \quad v \in V. \quad (15)$$

El objetivo en (8) es minimizar el costo ponderado total. El conjunto de restricciones (9) aseguran la conservación de flujo para cada vehículo en cada

cliente. La restricción (10) establece que cada cliente es visitado una sola vez. Los conjuntos de restricciones (11) y (12) aseguran que cada vehículo inicie y termine su ruta en el depósito. En (13) se establece que la demanda total de los clientes atendidos por un vehículo no puede exceder la capacidad del vehículo. En (14) se eliminan subtours y la restricción (15) es necesaria debido a que no se puede contar con servicios parciales en los clientes. En este modelo, el parámetro  $\rho$  se utiliza para obtener combinaciones alternativas del costo del servicio y el costo de transporte. <sup>82</sup>

---

<sup>82</sup> TAŞ, Op.cit., p. 214-224.

## 7. INSTANCIAS

Una instancia o experimento se obtiene otorgando valores específicos a todos los parámetros de un problema.<sup>83</sup>

En 1987, Solomon propone un conjunto de instancias, compuestas de 6 tipos de problemas: R1, R2, C1, C2, RC1 y RC2. Dichos problemas consideran varios factores que pueden influir en el comportamiento de las heurísticas de programación y enrutamiento. Estos factores incluyen: datos geográficos, número de clientes atendidos por un vehículo y características de la ventana de tiempo, como el porcentaje de clientes con limitaciones de tiempo, amplitud y posicionamiento de las ventanas de tiempo.<sup>84</sup>

Cada conjunto de prueba contiene entre 8 y 12 instancias. En R1 y R2, los datos geográficos son aleatoriamente generados por una distribución uniforme. En los experimentos C1 y C2, los clientes son ubicados en clústeres y finalmente en RC1 y RC2, algunos clientes son ubicados en clústeres mientras otros son ubicados aleatoriamente.<sup>85</sup> Ver figura 6.

Todas las instancias de la misma clase tienen las mismas coordenadas y la misma capacidad del vehículo, pero las ventanas de tiempo son diferentes. Para las clases R1, C1 y RC1, las ventanas de tiempo son más estrechas y tienen la capacidad del vehículo en 200 unidades, dando lugar a trayectos cortos que permiten servir entre 5 y 10 clientes por ruta; mientras que las clases R2, C2 y RC2 tienen ventanas de

---

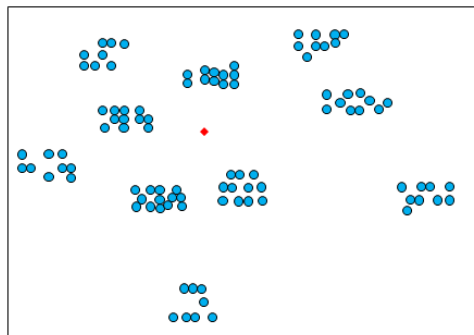
<sup>83</sup> TAŞ, Op.cit., p. 214-224.

<sup>84</sup> SOLOMON, Marius. Algorithms for the vehicle routing and scheduling problems with time window constraints. En:Operations Research Society of America [base de datos en línea].Vol. 35, no. 2(mar. – abr. 1987); p. 258-259 [citado en 15 de junio de 2016] Disponible en JSTOR Research Databases.

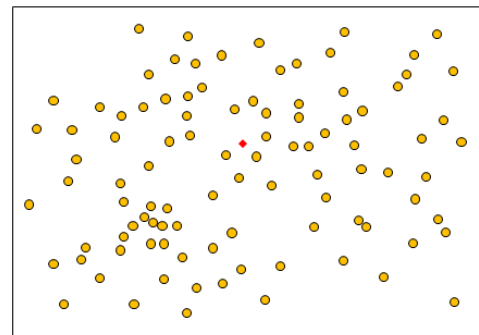
<sup>85</sup> KALLEHAUGE, Brian, *et al.* Vehicle Routing Problem with Time Windows. En: GERAD.COLUMN GENERATION. United States of America: Springer, 2005. 331p.

tiempo más anchas y mayor capacidad del vehículo (100 a 1000 unidades), lo cual permite servir a un gran número de clientes (alrededor de 30) con el mismo vehículo, es decir, recorrer grandes distancias.<sup>86</sup>

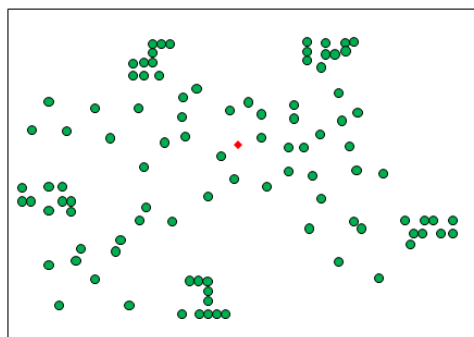
**Figura 6. Distribución de los clientes en el área geográfica para las instancias de Solomon (1987)**



a) Tipo C1 con 100 clientes.



b) Tipo R1 y R2 con 100 clientes.



c) Tipo RC1 y RC2 con 100 clientes.

La eficacia del algoritmo expuesto en este trabajo se prueba en las instancias C101, C102, C103, R101, R102, R103, RC101, RC102, RC103, C201, C201, C203, R201, R202, R203, RC201, RC202 y RC203 de Solomon (*Ver Anexo B*), con el fin de conocer el funcionamiento del mismo en los 3 tipos de experimentos posibles. Se

---

<sup>86</sup> OCHOA, Carlos Alberto, Op. cit., p.223.

considera que las instancias de Solomon constituyen un buen mecanismo de evaluación del algoritmo, dado que recoge los parámetros principales que aborda el problema que se pretende resolver con la investigación, es decir, el problema de ruteo de vehículos con tiempos de viaje estocásticos y ventanas de tiempo suave.

## 8. AJUSTE DE PARÁMETROS DEL ALGORITMO

### 8.1 Parámetros del MPNS-GRASP

La metaheurística MPNS-GRASP considera como único parámetro, el tamaño de la lista restringida de candidatos (RCL), cuyo ajuste cobra importancia dado que prioriza que los mejores arcos (menor distancia) sean los primeros en considerarse en la construcción de las rutas iniciales. Razón por la cual, una lista demasiado grande aumentaría la probabilidad de incluir arcos que no son muy promisorios en la solución inicial, así como que arcos que están muy abajo en cola para ingresar, tomen lugares privilegiados. Ahora bien, un tamaño de lista pequeño, limita la variabilidad entre las rutas iniciales, debido a que la probabilidad de que un mismo arco ocupe la misma posición en todas las rutas generadas, aumenta.

Gran parte de la literatura relacionada con el MPNS-GRASP aplicado al Problema de Ruteo de Vehículos, se debe a la investigación llevada a cabo por Yannis Marinakis<sup>87</sup>, quien en varias de sus publicaciones utiliza un tamaño de lista de 50 bordes candidatos, obteniendo buenos resultados. Es por esto que para el presente trabajo se define el parámetro con el mismo valor.

### 8.2 Parámetros del ENS

Expanding Neighborhood Search considera dos parámetros dentro de su formulación, el número de iteraciones (*RunLim*), que se define como el número de veces que se traza un nuevo círculo producto de una mejora y el factor de ampliación del radio, que determina en qué proporción aumenta el radio cada vez que no se encuentra una mejora al realizar un intercambio.

---

<sup>87</sup> MARINAKIS, Y. et al. Op. cit., pp.463-472.

En el desarrollo de este algoritmo se define *RunLim* como un número muy grande, cuyo valor numérico no es significativo, debido a que el algoritmo cuenta con un criterio de parada que hace que se rompa este ciclo, una vez la solución no muestre tendencia a mejorar, evitando así desperdicios en tiempo computacional.

En cuanto al factor de ampliación del radio se establece un aumento progresivo del 25%, con el ánimo de favorecer tanto la exploración como la explotación en el espacio de búsqueda.

### 8.3 Parámetros del PSO

Los resultados arrojados por un algoritmo de PSO están fuertemente influenciados por los parámetros establecidos. La correcta sincronización de los parámetros puede proveer mejores resultados que aplicar otros algoritmos de optimización.<sup>88</sup>

Los parámetros a sintonizar incluyen, el peso inercial  $w$  y las constantes de aceleración  $C_1$  y  $C_2$ .

El concepto de peso inercial se introduce en la versión original del PSO para balancear la búsqueda local y global durante el proceso de optimización.<sup>89</sup>

Un peso inercial alto reduce la influencia de la nostalgia y de la cooperación sobre el movimiento de las partículas, favoreciendo la exploración. Mientras que un peso

---

<sup>88</sup> Ghosh, A. y Gorai, A. Hue-Preserving Color Image Enhancement Using Particle Swarm Optimization. En: Dehuri, Satchidananda, *et al.* Integration of Swarm Intelligence and Artificial Neural Network. Singapore: World Scientific Publishing, 2011. p. 157-168.

<sup>89</sup> EBERHART, R. C. y SHI, Y. Particle swarm Optimization: Developments, applications and resources. En: Proc. IEEE Int. Conf. Evolutionary Computation, Vol. 9, (2001); p. 81–86.

inercial muy bajo promueve la convergencia hacia la solución propuesta por la propia partícula, y por la mejor solución de conjunto.<sup>90</sup>

En diferentes investigaciones se han propuesto valores específicos para este parámetro, que van desde considerar el peso inercial como una constante, hasta hacerlo variar durante cada iteración de manera decreciente o creciente; incluso, se ha propuesto un modelo que genera el peso inercial de manera aleatoria para evitar la dependencia de este con el número de iteraciones. Ver Tabla 2.

Eberhart y Shi<sup>91</sup> observaron que la solución óptima para la mayoría de los problemas puede ser mejorada variando el valor del peso inercial desde 0.9 al principio de la búsqueda hasta 0.4 al final de la misma.

**Tabla 2. Parámetros PSO, configuraciones típicas**

Peso Inercial	Parámetros Básicos del Algoritmo
Linealmente Decreciente	$w = 0.9 - 0.4; C_1 = C_2 = 2.0$
Linealmente Decreciente	$w = 0.9 - 0.4; C_1 = C_2 = 1.49445$
Aleatorio	$w = 0.5 + rnd/2.0; C_1 = C_2 = 1.49445$
Linealmente Decreciente	$w = 0.95 - 0.2; C_1 = C_2 = 2.0$
Constante	$w = 0.4; C_1 = C_2 = 2.0$

Ahora bien, la elección del peso inercial va ligada a la elección de las constantes de aceleración. Las constantes de aceleración  $C_1$  y  $C_2$  son llamadas coeficiente

---

<sup>90</sup> PÉREZ LÓPEZ, Jesús Ramón. Contribución a los métodos de optimización basados en procesos naturales y su aplicación a la medida de antenas en campo próximo. Santander, 2005, 68 p. Trabajo de Grado (Doctor). Universidad de Cantabria. Departamento de Ingeniería de Comunicaciones.

<sup>91</sup> EBERHART, R. C. y SHI, Y., Op. cit., p.81-86.

cognitivo y coeficiente social, respectivamente. El coeficiente cognitivo muestra la tendencia de las partículas de regresar a sus propias mejores posiciones, mientras que el coeficiente social identifica la propensión de las partículas de avanzar hacia la mejor posición de todo el enjambre.<sup>92</sup>

Diferentes autores han investigado el efecto de estos coeficientes en las trayectorias de las partículas y en las propiedades de convergencia del algoritmo, mostrando que cuando las constantes de aceleración aumentan, la frecuencia de oscilación de las partículas alrededor del óptimo incrementa, mientras que la disminución en las constantes de aceleración implica la aparición de patrones sinusoidales.<sup>93</sup> En general, los investigadores acostumbran a usar las constantes de aceleración con valor de 2 ( $C_1 = C_2 = 2$ ), para de esta manera garantizar que la media sea 1 y que las partículas sobrevuelen el objetivo la mitad del tiempo.<sup>94</sup>

En esta investigación se tomaron los parámetros del PSO más usados en la literatura y que han tenido un efecto favorable en los resultados de otros trabajos, es decir, el modelo linealmente decreciente:  $C_1 = C_2 = 2$ ;  $w = 0.9 - 0.4$ . Para la variación del peso inercial durante cada iteración se aplicó la fórmula mostrada a continuación:

$$w = w_{max} - \frac{w_{max} - w_{min}}{K} k \quad (16)$$

Donde  $K$  es el número máximo de iteraciones y  $k$  es la iteración actual.

---

<sup>92</sup> MARINI, Federico y WALCZAK, Beata. Particle Swarm Optimization (PSO): A Tutorial. En: Chemometrics and Intelligent Laboratory Systems [base de datos en línea]. Vol. 149 parte B (Dic., 2015); p. 153-165 [citado en 19 de junio de 2016] Disponible en ScienceDirect Research Databases.

<sup>93</sup> Ibid., p. 155.

<sup>94</sup> KENNEDY, J. y EBERHART, R. Particle Swarm Optimization. En: Neural Networks [base de datos en línea]. Vol. 4(Nov.-Dic., 1995); p. 1942-1948 [citado en 21 de Junio de 2016] Disponible en IEEE Xplore Digital Library.

## 9. LÍMITES DEL ESPACIO DE BÚSQUEDA

Uno de los problemas que surgen al aplicar la metaheurística de Optimización por enjambre de partículas, es que las partículas se salen de los límites del espacio de búsqueda dentro de los cuales, generalmente, se encuentra la solución.<sup>95</sup> Cuando esto ocurre es necesario traer de regreso a las partículas dentro de dichos límites, para no desperdiciar esfuerzos en encontrar soluciones que no están disponibles dentro del conjunto de soluciones factibles.

En la literatura existen diferentes estrategias para tratar esta cuestión, dichas estrategias controlan la posición de destino de la partícula y en caso de exceder los límites modifican los vectores de posición y/o velocidad de la misma.<sup>96</sup>

Se han investigado cuatro de tipos de barreras<sup>97</sup>:

- ∴ Pared Absorbente: lleva a la partícula al límite de la dimensión y anula su velocidad.
- ∴ Pared Reflectante: refleja la componente del vector posición en la dimensión excedida e invierte el signo de la velocidad en esta dimensión.
- ∴ Pared Invisible: no restringe el movimiento de las partículas, pero el fitness de aquellas que exceden los límites no se calcula.

---

<sup>95</sup> KAUR, Avneet & KAUR, Mandeep. Dealing with Boundary Constraint Violations in Particle Swarm optimization with Aging Leader and Challengers (ALC-PSO). En: International Journal of Computer Applications [base de datos en línea]. Vol. 121, no. 11 (jul. 2015); p.13-19 [citado en 2 de junio de 2016] Disponible en ProQuest Research Databases.

<sup>96</sup> PÉREZ LÓPEZ, Jesús Ramón, Op. Cit., p. 149.

<sup>97</sup> Ibid., p.149.

∴ Pared Frontera: lleva a la partícula al límite de la dimensión sin modificar el vector velocidad.

En el algoritmo propuesto se usa la barrera de contención denominada pared absorbente, así:

$$\text{Si } X_{i,t+1} > X_{i,max} \text{ entonces } X'_{i,t+1} = X_{i,max}$$

$$\text{Si } X_{i,t+1} < X_{i,min} \text{ entonces } X'_{i,t+1} = X_{i,min}$$

Donde  $X_{i,max}$  y  $X_{i,min}$  son los límites máximo y mínimo para la dimensión  $i$ , respectivamente.

## 10. REPRESENTACIÓN DE LA SOLUCIÓN

Una de las cuestiones clave para el diseño de un PSO exitoso en la implementación al problema de ruteo de vehículos, es la correspondencia adecuada entre las soluciones del VRP y las partículas en PSO.<sup>98</sup> Teniendo en cuenta que el VRP es un problema de optimización combinatoria con soluciones en el dominio de los enteros y la Optimización por Enjambre de Partículas fue creada para dar solución a problemas con dominio en el espacio continuo, se hace necesario que la relación entre la posición de la partícula y las rutas de vehículos estén claramente definidas.<sup>99</sup>

La definición de las partículas como una solución codificada generalmente se llama representación de la solución y el procedimiento para convertirlo en la solución específica del problema es conocido como método de decodificación.<sup>100</sup>

En este proyecto se hace uso de una representación, en la cual, el número de dimensiones de la posición de la partícula es igual al número de clientes en el problema. Es decir, para  $n$  clientes, cada partícula es codificada como un vector de números reales con  $n$  dimensiones, donde cada elemento del arreglo es un número continuo entre  $[0, n]$ .<sup>101</sup> De acuerdo con esto, la representación de la solución para

---

<sup>98</sup> MARINAKIS, Y., Op. Cit., p. 463-472.

<sup>99</sup> PORNISING, CHOOSAK. A Particle Swarm Optimization for the Vehicle Routing Problem. Estados Unidos, 2014. Tesis Doctoral (Ingeniería Industrial). Universidad de Rhode Island. Disponible en línea en < [http://digitalcommons.uri.edu/oa\\_diss](http://digitalcommons.uri.edu/oa_diss)>

<sup>100</sup> Geetha, S *et al.* A Hybrid Particle Swarm Optimization with Genetic Operators for Vehicle Routing Problem. En: Journal of advances in information technology [base de datos en línea]. Vol.1, No. 4 (nov. 2010); p. 181-188. [Citado en 16 de junio de 2016] Disponible en SpringerLink Research Databases.

<sup>101</sup> WANG, W., WU, B., ZHAO, Y. y FENG, D. Particle Swarm Optimization for Open Vehicle Routing Problem En: Kang Li *Et al.* Computational Intelligence. Kunming, China: Springer, 2006. p. 999-1007.

un VRP con  $n$  clientes, es una partícula  $n$ -dimensional en el espacio del problema  $s_i = (s_{i1}, s_{i2}, \dots, s_{id}) \ i = 1, 2, \dots, n$ .

## 10.1 MÉTODO DE CODIFICACIÓN Y DECODIFICACIÓN

Inicialmente, todas las soluciones (partículas) son almacenadas a través de la representación de un tour, es decir, a través de la secuencia específica de los nodos. Por ejemplo, para un problema con cinco clientes una posible representación del tour es:

**Figura 7. Ruta inicial**

1	3	5	2	4
---	---	---	---	---

De acuerdo con esto, la solución inicial debe ser codificada en un arreglo (matriz) con el objetivo de aplicar el algoritmo PSO. Esta conversión se realiza utilizando la siguiente ecuación:

$$x_{ij} = x_{min} + \left( \frac{x_{max} - x_{min}}{n} \right) * [(y_{ij} - 1) + rand] \quad (17)$$

Donde:

- ∴  $y_{ij}$  es la  $j^{th}$  dimensión de la  $i^{th}$  solución.
- ∴  $x_{ij}$  es la  $j^{th}$  dimensión de la  $i^{th}$  partícula.
- ∴  $rand$  es un numero uniformemente distribuido entre  $[0,1]$ .
- ∴  $n$  representa el número de clientes.
- ∴  $x_{min}$  y  $x_{max}$  son los valores límite de la posición de la partícula.

Para el ejemplo anterior, cada  $y_{ij}$  es convertido en un número real entre 0 y 5 usando la ecuación (17), de manera que el valor de posición de la partícula  $x_{ij}$  es:

**Tabla 3. Codificación de la ruta**

<b>Solución</b>	1	3	5	2	4
<b>Valor de la posición (<math>x_{ij}</math>)</b>	1,36	3,85	4,36	2,2	3,8

Luego del cálculo de la velocidad y la actualización de la posición, estos valores reales de posición de las partículas se convierten nuevamente en soluciones del VRP (valores enteros) utilizando el método del ROV. El ROV utiliza el valor de la posición más pequeño (SPV) de una partícula y le asigna a este, el menor valor del rango  $n$ ; del mismo modo, la posición en la que se ubica el siguiente SVP toma el valor del siguiente entero en  $n$  y así sucesivamente. Por lo tanto, la nueva solución al VRP para un vector actualizado de la posición es:

**Tabla 4. Decodificación de la ruta**

<b>Valor de la Posición</b>	1,02	4,33	2,85	1,9	3,8
<b>Solución</b>	1	5	3	2	4


Una vez realizado el proceso de decodificación de la solución se procede a considerar las demandas de los clientes con el fin de generar las rutas. Cada ruta es formada cumpliendo la restricción de capacidad del vehículo, es decir, un cliente

solo puede ser añadido a una gira si su demanda es menor a la capacidad del vehículo, de lo contrario, el cliente debe ser asignado a una nueva ruta. La figura 8 muestra el proceso de generación de rutas y la representación final de la solución teniendo en cuenta el ejemplo considerado anteriormente, para una flota de dos (2) vehículos con capacidad de veinte (20) cada uno. <sup>102</sup>

Todas las rutas deben empezar y terminar en el depósito. La asignación se realiza en orden de izquierda a derecha comprobando en cada nodo si la demanda del cliente actual es menor a la capacidad del vehículo en un instante  $t$ . Para el ejemplo, el primer cliente a ser asignado en la gira es el 1, como su demanda no viola la restricción de capacidad, este es asignado satisfactoriamente a la primera ruta. El segundo cliente en el tour es el 5, dado que la sumatoria de las demandas es menor a su capacidad este también es asignado a la primera ruta; en ese instante de tiempo la capacidad sobrante del vehículo es de 2, razón por lo cual el cliente 3 no puede ser asignado a esta ruta y debe abrirse una nueva gira para poder satisfacer su demanda. Este procedimiento se repite hasta que todos los clientes en el tour sean servidos.

**Figura 8. Representación de la solución**

Solución	1	5	3	2	4
Demanda	5	13	10	2	5



Rutas		$\sum d$
1	Depósito – 1 – 5 – 2 – Depósito	20
2	Depósito – 3 – 5 – Depósito	15

<sup>102</sup> MARINAKIS, Y., Op. Cit., p. 463-472.

## 11. ETAPAS DEL ALGORITMO PARA EL VRPSTTW

### MPNS-GRASP

1. Inicialización de los parámetros.
2. Construcción de la matriz de distancias.
3. Creación de la lista RCL.
4. Construcción de las rutas iniciales.
5. Evaluación del costo de las rutas generadas.

### ENS

1. Inicialización de los parámetros.
2. Aplicación de la metaheurística *Circle Restricted Local Search Moves Strategy en conjunto con la heurística de mejora 2-opt.*
3. Evaluación del costo de cada ruta.

### PSO

1. Inicialización de los parámetros.
2. Cálculo de la velocidad de las partículas y actualización de la posición.
3. Decodificación de las nuevas rutas.
4. Evaluación del costo de las rutas.
5. Actualización de los mejores locales y el mejor global.
6. Aplicación de la metaheurística de mejora Path Relinking.

La solución del problema VRPSTTW es la ruta representada en el Mejor Global (Gbest) al finalizar las iteraciones.

## **MPNS-GRASP**

### **1. Inicialización de los parámetros**

Se establecen los parámetros requeridos por la metaheurística MPNS-GRASP los cuales son: número de iteraciones ( $n_I$ ) y tamaño de la lista RCL ( $tm\_list$ ).

Así mismo, se ingresan los parámetros generales del modelo de optimización, que corresponden a: coordenadas de los clientes, número de clientes, número de vehículos, capacidad del vehículo, costo por demora, costo de antelación, costo por unidad de distancia recorrida, costo por usar el vehículo y tiempo de servicio.

### **2. Construcción de la matriz de distancias**

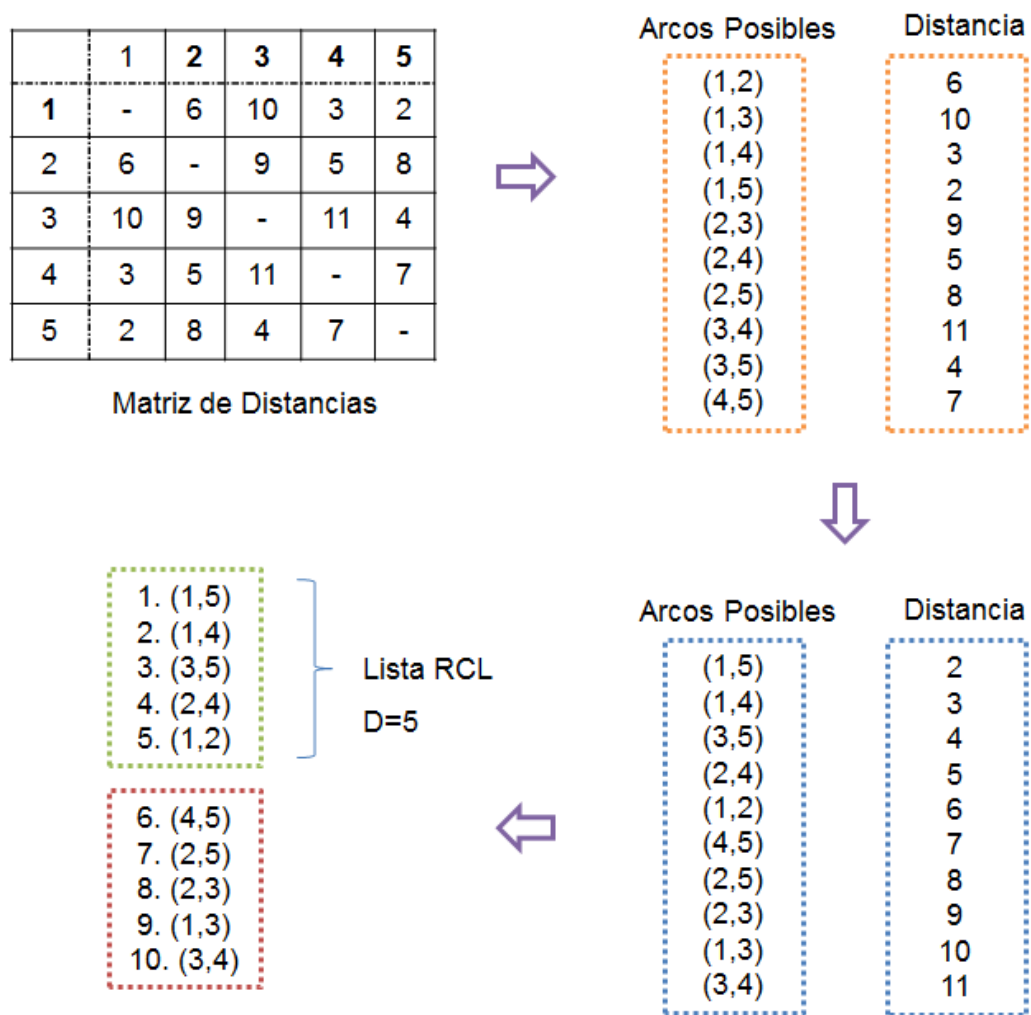
A partir de las coordenadas suministradas por las instancias se construye una matriz cuadrada que contiene las distancias euclidianas entre cada uno de los clientes.

### **3. Creación de la lista RCL**

En esta etapa se procede a generar una lista restringida de candidatos (RCL) a partir de todas las posibles combinaciones entre pares de clientes (bordes), donde dichos bordes se encuentran ordenados de menor a mayor distancia. De esta lista los primeros  $D$  bordes, son seleccionados con el fin de formar la lista restringida de candidatos final. El borde candidato para ser incluido en la ruta se selecciona al azar de la RCL utilizando un generador de números aleatorios. Ver Figura 9.

Por último, la lista se reajusta en cada iteración mediante la sustitución del borde que ha sido incluido en la ruta por otra arista que no pertenece a la RCL, la cual se encuentra en la posición  $D + m$  (donde  $m$  es el número de la iteración actual).

**Figura 9. Creación de la lista RCL**



#### 4. Construcción de las rutas iniciales

En lugar de utilizar una población inicial generada aleatoriamente, que puede o no contener buenas soluciones candidatas, se hace uso de una versión modificada del GRASP, conocida como MPNS-GRASP, que permite crear las rutas haciendo uso de algoritmos voraces que se alternan cada  $n$  iteraciones. Las funciones codiciosas utilizadas en esta etapa son el algoritmo de PRIM, el cual evalúa la inserción de un nuevo borde a la ruta considerando la distancia mínima entre el cliente inicial y final de la ruta ya existente y los dos nuevos clientes a ser insertados, y el algoritmo del

vecino más cercano el cual sólo toma a consideración las distancias entre la nueva dupla y el cliente final de la ruta.

El resultado de esta etapa es una ruta por cada iteración.

## **5. Evaluación del costo de las rutas generadas**

Una vez generadas las soluciones iniciales se hace necesario evaluar su calidad, mediante una función de costo que aborda tanto el costo de servicio como el costo de transporte. El costo de servicio está relacionado con el cumplimiento de las ventanas de tiempo para cada cliente y el costo de transporte con la distancia recorrida.

El aspecto estocástico se introduce en el código mediante la generación de escenarios para la evaluación del costo, lo que significa generar distribuciones de probabilidad para la variable tiempo de viaje y hallar el valor esperado del costo de la misma.

## **ENS**

### **1. Inicialización de parámetros**

Se definen los parámetros de la metaheurística ENS, los cuales son: número de iteraciones (*runlim*) y factor de ampliación del radio.

El número de iteraciones determina el criterio de parada de la heurística de mejora utilizada y el factor de ampliación del radio entrega la proporción en que aumentará el radio, cada vez que no se cumplan las condiciones requeridas para realizar un intercambio o cuando la modificación llevada a cabo en la ruta no mejore la calidad de la solución.

## **2. Aplicación de la estrategia Circle Restricted Local Search Moves en conjunto con la heurística de mejora 2-opt.**

Expanding Neighborhood Search se basa en la estrategia de búsqueda local CRLSM, en esta estrategia el tiempo de cálculo se reduce significativamente en comparación con otros algoritmos heurísticos y metaheurísticos, debido a que se excluye del procedimiento de búsqueda a todos los bordes que no conllevan a una mejora en la solución.

Este método comienza con un tamaño de vecindario dado (el radio del círculo), el cual corresponde al borde más largo en la ruta actual. Una vez delimitada la zona, se evalúa si se cumplen las condiciones necesarias para aplicar el algoritmo de mejora, las cuales son:

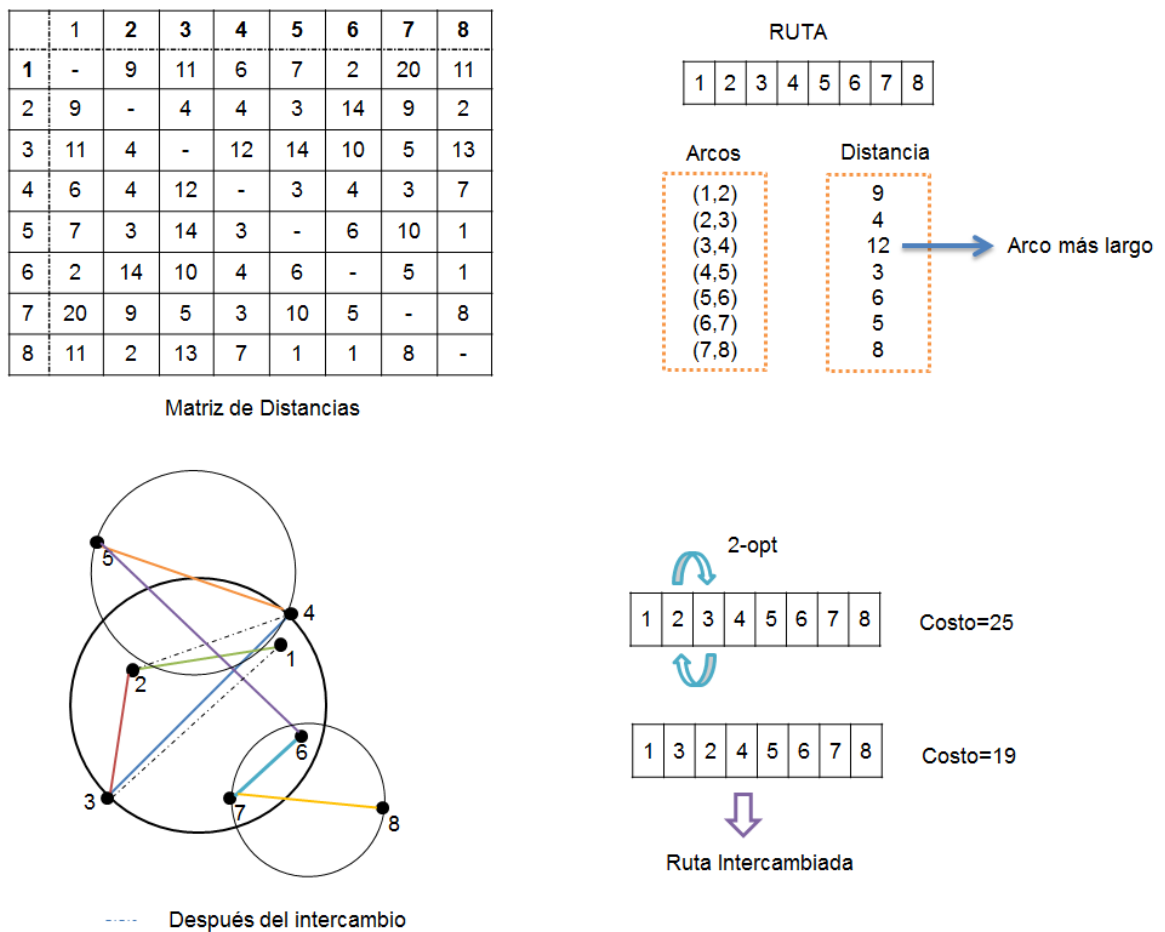
- ∴ El vecindario debe contener al menos dos bordes.
- ∴ Los bordes candidatos al intercambio deben ser adyacentes.
- ∴ Los bordes candidatos al intercambio no deben compartir un nodo.

El algoritmo de mejora usado al interior del círculo es la búsqueda local 2-opt, la cual a partir de una ruta  $X$  elimina dos arcos adyacentes y reconecta los dos nodos libres de forma que se cree una nueva ruta  $X'$ ; si el costo de esta nueva ruta es menor que el costo de la ruta inicial se reemplaza la ruta  $X$  por la ruta  $X'$  y se procede a trazar un nuevo círculo, cuyo radio será el arco con la mayor longitud en esta nueva ruta. En caso de que el costo de la ruta intercambiada no mejore la longitud del radio del círculo, se incrementa de acuerdo al parámetro de ampliación del radio, con el propósito de incluir más bordes candidatos a ser intercambiados en el vecindario. Este procedimiento se repite hasta que uno de los criterios de parada se active.

Un aspecto clave a considerar en esta etapa del algoritmo es que el tiempo computacional requerido se incrementa, debido a la gran cantidad de

combinaciones que pueden resultar al aplicar la heurística 2-opt, por lo cual se hace necesario implementar la siguiente estrategia a fin de mitigar este efecto: La búsqueda local 2-opt se detendrá en el momento en que encuentre una ruta con menor costo que la inicial. De lo contrario, el criterio de parada se dará a partir de un número de iteraciones establecido.

**Figura 10. Aplicación de la estrategia CRLSM**



### 3. Evaluación del costo de cada ruta

El procedimiento de Expansión de Búsqueda Local pretende mejorar en cada iteración una solución de la matriz de rutas proveniente de la metaheurística anterior (MPNS-GRASP). Lo que significa que cada partícula recibe tratamiento en tiempos diferentes y no de manera simultánea.

Por tanto, la generación de escenarios para la evaluación del costo se debe realizar en cada iteración para cada ruta.

## PSO

### 1. Inicialización de los parámetros

Se establecen los parámetros de la metaheurística principal (PSO) los cuales son: número de enjambres ( $numswarm$ ), factor de inercia mínima y máximo ( $W_{min}$  y  $W_{max}$ ), constante de aceleración cognitiva y social ( $C_1$  y  $C_2$ ), número máximo de iteraciones ( $MaxIter$ ) y número mínimo y máximo de clientes ( $X_{min}$  y  $X_{max}$ ).

Una cuestión clave en el diseño exitoso del PSO para el problema de ruteo de vehículos, es encontrar la correspondencia adecuada entre las soluciones del VRP y las partículas del PSO. Por tal razón, cada solución es representada como un camino o tour que cubre la totalidad de los clientes, mediante una secuencia de nodos y se transforma al dominio continuo del problema, convirtiéndose en partículas.

Esta transformación se hace a partir de un método de codificación que genera un arreglo, cuyos elementos son números reales entre 0 y  $n$ , siendo  $n$  el número de

clientes. La conversión se lleva a cabo a través de la fórmula propuesta por Geetha et al<sup>103</sup>:

$$X_{ij} = X_{min} + \frac{(X_{max}-X_{min})}{n} [Y_{ij} - 1 + rand()] \quad (18)$$

Dónde:

- ∴  $Y_{ij}$  es la  $j^{th}$  dimensión de la  $i^{th}$  solución.
- ∴  $X_{ij}$  es la  $j^{th}$  dimensión de la  $i^{th}$  partícula.
- ∴  $rand()$  es un número distribuido uniformemente [0,1].
- ∴  $n$  representa el número de clientes.
- ∴  $X_{min}$  y  $X_{max}$  son los valores límites de la posición de una partícula.

Inicialmente, la matriz de soluciones arrojada por la metaheurística ENS se transforma por medio de la fórmula (18) y el resultado (arreglo de elementos reales) se considera la posición inicial de las partículas en PSO.

Así mismo, se inicializa la velocidad en cero para cada  $j^{th}$  dimensión de la  $i^{th}$  partícula. A pesar de que este enfoque ha sido reiteradamente debatido, se sigue considerando una buena estrategia para la inicialización de la velocidad de las partículas pues evita que se violen los límites del espacio de búsqueda y a su vez la divergencia del algoritmo.

## **2. Cálculo de la velocidad de las partículas y actualización de la posición**

Las partículas se mueven en el espacio de búsqueda de acuerdo a las ecuaciones (19) y (20):

---

<sup>103</sup> Geetha., Op. cit; p. 181-188.

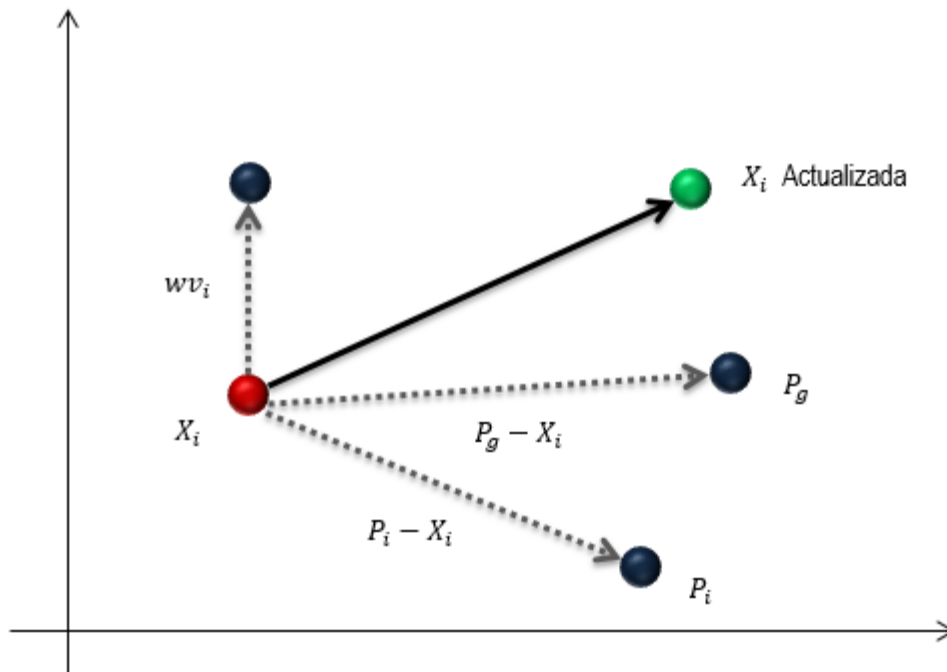
$$\vec{v}_i(t+1) = w * \vec{v}_i + (C_1 * Rand()) * (\vec{p}_i - \vec{x}_i) + (C_2 * Rand()) * (\vec{p}_g - \vec{x}_i) \quad (19)$$

$$\vec{x}_i(t+1) = \vec{x}_i + \vec{v}_i \quad (20)$$

En la primera ecuación se calcula la velocidad de las partículas, la cual gobierna la forma en que las partículas se mueven a través del espacio de búsqueda. La velocidad tiene en cuenta 3 términos: el primero, la inercia, que evita que la partícula cambie drásticamente de dirección, el segundo, el componente cognitivo, que representa la tendencia de las partículas a regresar a sus propias mejores posiciones y el último, el componente social, que identifica la tendencia de una partícula a avanzar hacia la mejor posición de todo el enjambre.

Posteriormente, se aplica la ecuación (20) que actualiza la posición de la partícula.

**Figura 11. Actualización de la posición y la velocidad en PSO**



### **3. Decodificación de las nuevas rutas**

Dado que en la etapa anterior se obtiene un arreglo de elementos continuos que no permite ver claramente la orientación de la solución, es decir la ruta, se hace necesario aplicar un procedimiento que convierta la partícula en solución, pasando del espacio de solución del PSO al espacio discreto que define los arcos  $i, j$ .

El procedimiento usado es ROV (Rank Order Value), el cual asigna el valor de 1 al elemento más pequeño del vector, el 2 al siguiente más pequeño y así sucesivamente hasta completar todos los elementos de la partícula.

### **4. Evaluación del costo de las rutas**

Una vez obtenidas las rutas provenientes de las partículas que se movieron en el espacio de búsqueda de PSO, se procede a evaluar la calidad de estas soluciones mediante la estrategia usada a lo largo del algoritmo, la generación de escenarios que consideran el comportamiento estocástico de la variable tiempo de viaje.

### **5. Actualización de los mejores locales y el mejor global**

Dado que el principio clave del funcionamiento del PSO es que las partículas se muevan en la dirección de los mejores resultados encontrados hasta el momento, la actualización de los mejores locales y el mejor global se convierte en una parte fundamental del procedimiento metaheurístico, esta se lleva a cabo mediante la comparación de la calidad de las soluciones actuales de cada partícula y sus mejores locales ( $P_{best}$ ), así como el mejor global ( $G_{best}$ ). En caso de que el costo sea menor en la posición actual de la partícula, esta pasaría a ser el mejor local. De la misma manera, después de que se actualicen los mejores locales se comparan con el mejor global y se hace la respectiva actualización si el costo de alguno de los  $P_{best}$  es menor.

### **6. Aplicación de la metaheurística de mejora Path Relinking**

Se plantea la metaheurística Path Relinking con el fin de mejorar las rutas obtenidas hasta el momento. Es así como se selecciona el mejor local  $P_{best}$  de cada partícula

como solución inicial y el mejor global como solución guía. Gradualmente, se agregan atributos de la solución guía a la solución inicial y se evalúa la calidad de la nueva ruta, en caso de que el costo disminuya, esta ruta pasa a ser la solución inicial y se continúa el procedimiento hasta que la solución inicial sea igual a la solución guía.

## 12. APLICACIÓN DE LA METAHEURÍSTICA PROPUESTA A UN PROBLEMA DEL TSP

Una vez realizado el estudio del algoritmo híbrido propuesto y con el fin de afianzar los conocimientos acerca del procedimiento y funcionamiento del mismo, se procede a dar solución a un problema propuesto en la literatura con el objeto de mostrar el procedimiento ejecutado y facilitar la comprensión en la aplicación al problema VRP con ventanas de tiempo suave y tiempos de viajes estocásticos. El ejemplo seleccionado es un problema del agente viajero (Traveling Salesman Problem, TSP) propuesto en el libro “Introducción a la Investigación de Operaciones” de Frederick Hillier.<sup>104</sup>

Un modelo TSP se define mediante el número de ciudades  $n$  y la matriz de distancias  $d_{ij}$ . Para el ejemplo considere la siguiente matriz TSP:

**Figura 12. Matriz de distancias TSP**

$$||d_{ij}|| = \begin{pmatrix} 0 & 8 & 4 & 8 & 11 \\ 8 & 0 & 5 & 6 & 3 \\ 4 & 5 & 0 & 4 & 7 \\ 8 & 6 & 4 & 0 & 6 \\ 11 & 3 & 7 & 6 & 0 \end{pmatrix}$$

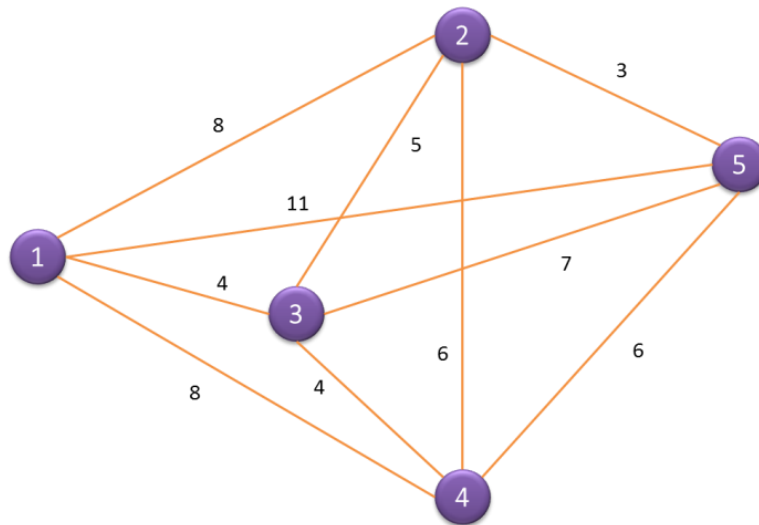
---

<sup>104</sup> HILLIER, Frederick S.; LIEBERMAN, Gerald J. Introducción a la investigación de operaciones. 9ª Ed. Mc Graw Hill. 2010, pp. 570-580. (ISBN: 978-607-15-0308-4).

El problema se compone de cinco ciudades como lo indica el diagrama, de las cuales, la número uno, corresponde a la residencia del viajero. La esencia del problema consiste en encontrar el recorrido más corto (cerrado), donde cada ciudad es visitada exactamente una vez antes de regresar al punto de partida. La definición de un recorrido prohíbe conectar una ciudad a sí misma al asignar una penalización muy alta a los elementos diagonales de la matriz de distancias.<sup>105</sup>

Los valores de las aristas que unen a cada par de ciudades corresponden a la distancia entre cada una de ellas. La figura 13 presenta el problema correspondiente.

**Figura 13. Grafo del problema del TSP**



---

<sup>105</sup> TAHA, Hamdy A. Investigación de operaciones. 9° Ed. México: Pearson, 2012. pp. 397-402. (ISBN: 978-607-32-0796-6).

Teniendo en cuenta las etapas del algoritmo propuesto, el primer paso es la construcción de las soluciones iniciales (rutas), haciendo uso de la metaheurística MPNS- GRASP. Para ello se debe:

1. Ordenar todos los arcos presentes en el problema de forma ascendente según la distancia.
2. Definir un tamaño de lista  $m$  en el cual estarán incluidos los  $n$  primeros arcos de la lista ordenada (lista RCL).
3. Extraer un arco de la lista aleatoriamente con el objeto de iniciar el tour. Cada vez que un elemento sale de la lista RCL, el siguiente elemento en espera de la lista ordenada pasa a ocupar este lugar, esto se repite hasta que ya no queden más elementos en la lista ordenada.
4. Repetir el paso anterior y aplicar un algoritmo codicioso (algoritmo del vecino más cercano para este caso) con el fin de construir la solución inicial.
5. Volver al paso 3 hasta que todos los nodos estén incluidos en el tour.
6. Evaluar el fitness de la solución.

El procedimiento anterior se repite cuantas soluciones iniciales se quiera generar. Es importante resaltar que cada cierto número de soluciones iniciales construidas (iteraciones), el algoritmo codicioso empleado cambia como alternativa hacia una nueva búsqueda, lo cual otorga variedad en las soluciones generadas.

Los algoritmos codiciosos usados en el algoritmo propuesto en esta investigación para la construcción de las rutas iniciales son: el algoritmo del vecino más cercano y el algoritmo de Prim.

∴ **Algoritmo de Prim**

El algoritmo de Prim pertenece a la categoría de algoritmos voraces. Encuentra el árbol de expansión de costo mínimo, partiendo de un borde (raíz) al cual se le van añadiendo bordes de costo mínimo en cada paso, es decir, elige la arista de menor peso (costo) que une un vértice en el conjunto del árbol con un vértice que no está en el árbol, sin crear ciclos.<sup>106</sup>

∴ **Algoritmo del vecino más cercano**

El algoritmo del vecino más cercano es un procedimiento heurístico constructivo sencillo que da solución al problema del agente viajero (TSP). Este algoritmo trata de construir un ciclo Hamiltoniano de bajo costo basándose en el vértice más cercano a un nodo de referencia.<sup>107</sup>

Es decir, el procedimiento parte de elegir un vértice inicial  $j_1$ ; una vez seleccionado, mide la distancia que hay desde ese vértice a los restantes y elige aquel cuya distancia al vértice inicial sea la mínima (el vecino más cercano  $j_2$ ). Siguiendo este procedimiento construye una trayectoria  $j_1, j_2, j_3, \dots, j_k, j_{k+1}, \dots, j_n$ , donde el vértice  $j_{k+1}$  se elige tomando la mínima distancia que hay desde  $j_k$  hasta cada uno de los vértices que sean distintos de los ya elegidos  $j_1, j_2, j_3, \dots, j_k$ . Al terminar, se agrega el arco que va del vértice  $j_n$ , hasta el vértice  $j_1$ , con el fin de cerrar el tour.<sup>108</sup>

Para el ejemplo tenemos:

---

<sup>106</sup> JOYANES, Luis, et al. Estructura de datos en C. Madrid: McGraw-Hill España, 2005. 440 p.

<sup>107</sup> Martí, Op. cit., pp. 1-60.


<sup>108</sup> Universidad Autónoma de Nuevo León. Investigación de Operaciones en Acción: Heurísticas para la Solución del TSP [En línea]. <http://yalma.fime.uanl.mx/~roger/work/Papers/article/article-inge-2000.pdf> [citado en 1 de julio de 2016].

**Tabla 5. Parámetros iniciales para el MPNS- GRASP**

Parámetros	
Tamaño Lista RCL	5

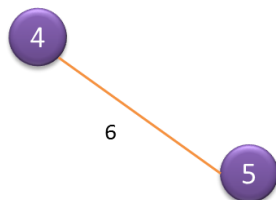
**Tabla 6. Lista Ordenada de todos los arcos que componen el problema**

Lista Ordenada		
(2,5)	3	} Lista RCL
(1,3)	4	
(3,4)	4	
(2,3)	5	
(2,4)	6	
(4,5)	6	
(3,5)	7	
(1,2)	8	
(1,4)	8	
(1,5)	11	

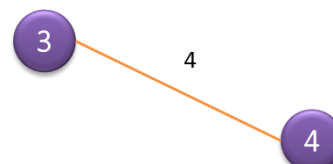


**Figura 14. Duplas seleccionadas**

Segunda dupla seleccionada



Primera dupla seleccionada



Como la primera y segunda dupla seleccionadas comparten un nodo, el camino se une de la forma 4-3-5. Una vez llevado a cabo todo el procedimiento, se obtiene la primera solución inicial representada por la siguiente secuencia de nodos: 1-4-3-5-2-1. Para el ejemplo se generan dos soluciones iniciales.

**Tabla 7. Soluciones Iniciales**

Solución	Tour	$\Sigma dij$
Solución 1	1-4-3-5-2-1	30
Solución 2	1-3-2-4-5-1	32

Una vez obtenidas las soluciones iniciales, el siguiente paso es utilizar un método de búsqueda local que permita encontrar mejoras en estas soluciones. ENS utiliza una estrategia conocida como *Circle Restricted Local Search Moves* la cual reduce significativamente el tiempo de cálculo en comparación con otros algoritmos, ya que excluye del procedimiento de búsqueda a todos los arcos que no tienden a mejorar la solución, por tanto, la estrategia restringe el espacio de búsqueda en círculos alrededor de los bordes que son candidatos a ser eliminados (aquellos que presentan mayor longitud en el tour) para luego aplicar una estrategia de intercambio 2-opt.<sup>109</sup>

Teniendo en cuenta esto, se debe:

---

<sup>109</sup> MARINAKIS, Yannis. Multiple Phase Neighborhood Search-GRASP for the Capacitated Vehicle Routing Problem. En: Expert Systems with Applications [base de datos en línea]. Vol. 39, no. 8 (Jun. 2012); p. 6807-6815. [Citado en 22 de agosto de 2015]. Disponible en Science Direct Research Databases.

1. Seleccionar el borde más largo del tour.
2. Ubicarse en un nodo de este borde y trazar un círculo con radio igual a la longitud del arco.
3. Realizar movimientos 2-opt con aquellos bordes que se encuentren al interior del círculo, siempre y cuando se cumplan las condiciones para hacerlo.
4. Si al interior del círculo no existen bordes para realizar intercambios, ampliar la longitud del radio en un porcentaje  $\alpha$  hasta que se cumplan las condiciones necesarias.
5. Una vez se encuentre una mejor solución (nueva ruta), repetir el procedimiento. El criterio de parada para este algoritmo se da por número de iteraciones.

Aplicando este procedimiento al ejemplo, tenemos que:

**Tabla 8. Arcos que componen la primera ruta con sus respectivas distancias**

Arcos	$D_{ij}$
(1,4)	8
(4,3)	4
(3,5)	7
(5,2)	3
(2,1)	8

→ Arco más largo

Los resultados arrojados al aplicar la estrategia ENS al ejemplo son:

**Tabla 9. Soluciones mejoradas por ENS.**

Solución	Tour	$\Sigma dij$
Solución 1	1-3-4-5-2-1	25
Solución 2	1-3-2-5-4-1	26

Una vez la estrategia ENS entrega los resultados mejorados, se da paso a la aplicación del algoritmo central: Optimización por Enjambre de Partículas, PSO.

Para dar inicio a este algoritmo se debe:

1. Inicializar los parámetros del modelo: Constante de aceleración cognitiva y social, factor de inercia máximo y mínimo y número de iteraciones.
2. Inicializar los mejores personales ( $P_{best}$ ) de las partículas y el mejor global ( $g_{best}$ ). Para la primera iteración el  $P_{best}$  se hace igual a la partícula inicial ( $P_{best_i} = Y_i$ ) y el  $g_{best}$  es la mejor solución inicial encontrada (Partícula 1 para el ejemplo).
3. Teniendo en cuenta que en esta fase del algoritmo se hace necesario encontrar la correspondencia adecuada entre la solución del TSP y el PSO, cada solución se debe codificar al inicio, para poder llevar a cabo el procedimiento. Esencialmente lo que se realiza es la conversión del tour de clientes (valores discretos) en valores de posición reales dentro del espacio de solución. La conversión se realiza usando la ecuación 21:

$$x_{ij} = x_{min} + \left( \frac{x_{max} - x_{min}}{n} \right) * [(y_{ij} - 1) + rand] \quad (21)$$

**Tabla 10. Parámetros iniciales del modelo PSO**

Parámetros	
$X_{max}$	5
$X_{min}$	1
n	5
$C_1$	2
$C_2$	2
$W_{max}$	0.9
$W_{min}$	0.4
$K$	5

Cada solución inicial que entra a esta etapa del proceso es denominada partícula. Para el ejemplo, se tienen dos partículas cuyos vectores  $X_i$  obtenidos de aplicar la ecuación 21, se presentan en la tabla 11:

**Tabla 11. Codificación de las partículas**

Partícula 1:

Dimensión j	1	2	3	4	5
Solución ( $Y_i$ )	1	3	4	5	2
Vector Posición ( $X_i$ )	1,01	2,78	3,79	4,56	2,59

Partícula 2:

Dimensión j	1	2	3	4	5
Solución ( $Y_i$ )	1	3	2	5	4
Vector Posición ( $X_i$ )	1,12	2,8	1,99	4,47	3,58

Una vez codificada la solución inicial se procede a:

4. Calcular la velocidad de cada partícula.
5. Calcular la nueva posición.
6. En caso de que la partícula se salga de la frontera, regresarla al espacio de búsqueda.
7. Decodificar la partícula con el fin de volver el vector posición, una solución del TSP.
8. Evaluar el costo de cada partícula.
9. Aplicar la estrategia Path Relinking como método de mejora de la solución.
10. Actualizar los mejores personales y el mejor global.
11. Codificar las partículas.

El proceso se repite hasta un número de iteraciones predeterminado (cinco para el ejemplo). Continuando con el ejemplo y aplicando las ecuaciones (22) y (23), tenemos para la partícula 2:

$$\vec{v}_i(t+1) = w * \vec{v}_i + (C_1 * Rand()) * (\vec{p}_i - \vec{x}_i) + (C_2 * Rand()) * (\vec{p}_g - \vec{x}_i) \quad (22)$$

$$\vec{x}_i(t+1) = \vec{x}_i + \vec{v}_i \quad (23)$$

La tabla 12 muestra los resultados de velocidad y posición para la primera iteración.

**Tabla 12. Velocidad y posición de la partícula 2 en la primera iteración**

Vector Posición ( $X_i$ )	1,12	2,8	1,99	4,47	3,58
$v_i(t + 1)$	-0,19	-0,03	3,06	0,153	-1,68
$x_i(t + 1)$	0,933	2,766	5,05	4,623	1,897

Con el fin de convertir la solución arrojada por el PSO a una solución del problema, se hace necesario regresar los valores reales de posición del vector  $X_i$  en valores discretos del vector solución  $Y_i$ . Para esto, se utiliza la regla ROV (Rank Order Value), creada por Bo, Wang y Jing<sup>110</sup>, en la cual se asigna la valoración de 1 al menor valor dentro de cada partícula (Smallest Position Value SPV), posteriormente se asigna la valoración de 2, al segundo menor valor de posición y así sucesivamente para los demás valores de la partícula.

La tabla 13 muestra el vector  $Y_i$  al aplicar la regla ROV para la solución arrojada anteriormente:

**Tabla 13. Decodificación del vector posición  $X_i$ .**

Dimensión j	1	2	3	4	5
Vector Posición ( $X_i$ )	0,933	2,766	5,05	4,623	1,897
Solución ( $Y_i$ )	1	3	5	4	2

---

<sup>110</sup> LIU, Bo, WANG, Ling y JIN, Yi-hui. An effective hybrid PSO-based algorithm for flow shop scheduling with limited buffers. En: Computers & Operations Research. 2008. Vol. 35, pp. 2791–2806. (DOI 10.1016/j.cor.2006.12.013).

La distancia total recorrida asociada a esta nueva solución es de 31. Como se puede observar en esta primera iteración la partícula no encontró una mejor posición por lo cual no se actualiza su  $P_{best}$ .

Finalmente, se hace uso de la estrategia Path Relinking como método de mejora para la solución.

Path Relinking es una estrategia de post optimización que busca generar nuevas soluciones a partir de la exploración de trayectorias que conectan soluciones de alta calidad (mejores soluciones). El procedimiento inicia partiendo de una solución conocida como solución inicial y genera una trayectoria explorando el vecindario que conduce hacia la otra solución, conocida como solución objetivo.<sup>111</sup>

Para el ejemplo, se considera al mejor global como la solución guía y al  $P_{best}$  de cada partícula como las soluciones iniciales. El procedimiento a seguir es:

1. Determinar qué posición en la solución inicial difiere con la solución objetivo.
2. Ubicar en esta posición el valor que se encuentra en esa dimensión de la solución objetivo y trasladar el valor que se encontraba allí hacia el lugar vacío.
3. Evaluar el costo de esa nueva solución.

---

<sup>111</sup> F. Glover, M. Laguna, R. Marti. Scatter search and path relinking: advances and applications. F. Glover, G.A. Kochenberger (Eds.), Handbook of Metaheuristics, Kluwer Academic Publishers, Boston (2003), pp. 1–36.

4. En caso de que la solución mejore, se considera a esta como la solución inicial y se repite el proceso hasta considerar todas las dimensiones diferentes entre las soluciones.
5. La mejor solución encontrada se actualiza como el  $P_{best}$  de esa partícula.
6. Comparar la nueva solución encontrada con el  $G_{best}$  de la iteración, si el costo asociado a la misma es menor, actualizar el  $G_{best}$ .

Este procedimiento se aplicará para todos los  $P_{best}$  de las partículas en cada iteración. La tabla 14 muestra la solución encontrada luego de aplicar este método:

**Tabla 14. Solución objetivo, inicial y nueva solución encontrada con la estrategia Path Relinking**

Solución Objetivo	1	3	4	5	2
Solución inicial	1	3	2	5	4
Nueva solución	1	3	4	5	2

La distancia total recorrida asociada a esta solución es de 25, hecho que evidencia que la distancia para esta ruta disminuyó, por lo cual se actualiza el respectivo  $P_{best}$  de la partícula número 2. El  $G_{best}$  no se actualiza, debido a que la nueva ruta no mejoró en comparación con esta. Una vez ejecutadas todas las iteraciones del algoritmo central, se encuentra que la ruta óptima es: 1-2-5-4-3-1 la cual minimiza la distancia recorrida a 25 y corresponde a la ruta óptima del problema propuesto.

### 13. CÁLCULO DE LA FUNCIÓN OBJETIVO

Debido a que el cálculo del costo total de una ruta está directamente relacionado con la naturaleza estocástica del SVRPTTW, el cual considera los tiempos de viaje entre clientes como aleatorios, se hace necesario evaluar el costo promedio de una ruta a priori en varios escenarios, donde los tiempos de llegada entre nodos, varían según una distribución de probabilidad dada.

En este problema, las ventanas de tiempo suaves permiten brindar un servicio temprano o tarde a los clientes generando una penalización. El objetivo es minimizar la suma de los costos de transporte y de servicio. Los costos de transporte son el resultado de dos elementos: la distancia total recorrida y el número de vehículos usados, mientras que, el costo de servicio se causa solo si se incurre en llegadas tempranas o tardías, dado que esto corresponde a una violación en la ventana de tiempo del cliente.<sup>112</sup>

Con el objetivo de dar claridad al procedimiento que se lleva a cabo para el cálculo de la función objetivo, se desarrolla un ejemplo sencillo que considera los siguientes aspectos:

- ∴ Las coordenadas de los clientes son:  $Depósito = [0,0]$ ,  $A = [1, 3]$ ,  $B = [3, 4]$ ,  $C = [5, 3]$ ,  $D = [4, 1]$ .
- ∴ Las ventanas de tiempo asociadas a cada cliente son:  $A = [16 - 18]$ ,  $B = [30 - 32]$ ,  $C = [28 - 30]$ ,  $D = [42 - 44]$ .
- ∴ El tiempo de atención a un cliente es de una unidad.

---

<sup>112</sup> TAŞ, D. Op. cit., pp. 789-799.

- ∴ Las demandas asociadas a los clientes son:  $d_A = 10, d_B = 5, d_C = 8, d_D = 6$ .
- ∴ No hay limitación en la flota de vehículos.
- ∴ El costo por usar un vehículo es de \$2.
- ∴ El costo por unidad de distancia recorrida es de \$1.
- ∴ El costo por demora o antelación es de \$1.
- ∴ Cada vehículo tiene una capacidad constante de:  $Q = 15$ .
- ∴ La ruta a priori es  $R = Depósito - A - B - C - D - Depósito$ .
- ∴ El conjunto de tiempos de viaje entre clientes es:  $t_{Depósito-A} = [14, 17, 18.2]$ ,  
 $t_{A-B} = [13, 15, 13.8]$ ,  $t_{Depósito-C} = [29, 28.7, 27]$ ,  $t_{C-D} = [11.2, 13, 12]$ .
- ∴ El número de escenarios  $E = 3$ .

Una vez definidos estos aspectos, se procede a crear los escenarios para el problema a solucionar.

$$E1 = \{A = 17, B = 15, C = 28.7, D = 13\}$$

$$E2 = \{A = 14, B = 13, C = 27, D = 12\}$$

$$E3 = \{A = 18.2, B = 13.8, C = 29, D = 11.2\}$$

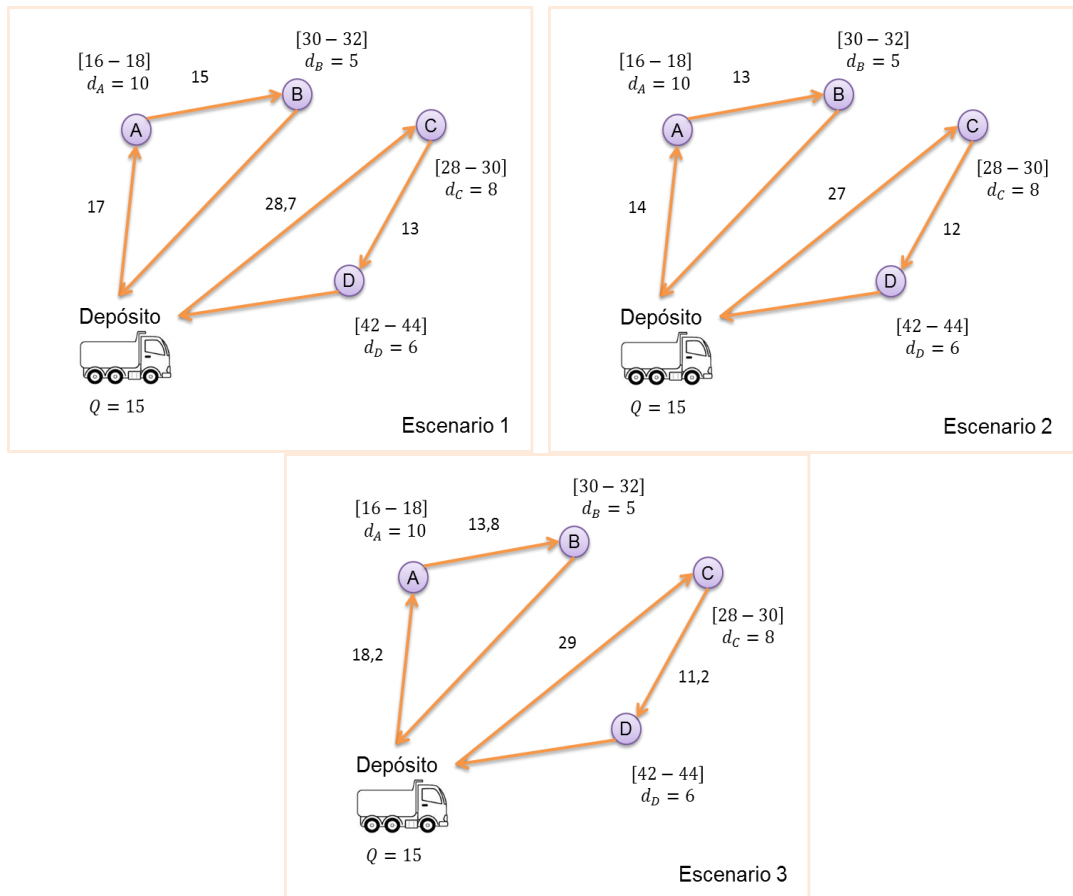
La figura 13 ilustra el recorrido de los vehículos para satisfacer las demandas de los clientes, teniendo en cuenta la ventana de atención al proveedor en cada uno de los escenarios.

Dado que los costos de transporte están asociados a la distancia recorrida por cada vehículo y a la cantidad de flota utilizada, estos no varían dependiendo del escenario ya que se evalúan siempre para la misma ruta a priori y la demanda de cada cliente es constante.

1. En el escenario uno es posible observar como el primer vehículo no logra cumplir con la ventana establecida por el cliente B, dado que el recorrido entre A y B es

mayor al límite superior de la ventana de este cliente, por lo cual se cobra una penalización de una unidad por retraso. El costo total de este escenario es de 27,59 unidades.

**Figura 15. Escenarios para el cálculo de la función objetivo**



- En el escenario 2 solo el cliente B es atendido dentro de su ventana. Para los demás clientes: A, C y D, el vehículo llega a entregar la mercancía antes de que el espacio de atención de cada uno de estos se abra, es decir, cuando el tiempo de viaje es menor al límite inferior de la ventana predeterminada por cada uno de ellos, por lo que los clientes son servidos en este momento, penalizando el costo por llegada temprana. El costo total de este escenario es de: 30,59 unidades.

3. En el escenario 3, ocurre una combinación de violaciones a las ventanas de tiempo. Tanto para el cliente A como para el Cliente B, los tiempos de llegada son mayores que los límites superiores de las ventanas establecidas por cada uno de ellos, razón por la cual, se cobra una penalización por demora. Por otro lado, el tiempo de llegada al cliente D es menor al límite inferior de su ventana causando así una penalización por antelación. El costo total de este escenario es de 28,59 unidades.

El costo total de la ruta a priori, es igual al promedio de los costos en cada uno de los escenarios evaluados, es así como el resultado es:

$$C_t = \frac{27,59 + 30,59 + 28,59}{3} = 28,92 \text{ unidades}$$

## 14. DEFINICIÓN DE LOS COSTOS PARA LA FUNCIÓN OBJETIVO

En este proyecto se considera la formulación matemática para el problema de ruteo de vehículos con tiempos de viaje estocásticos y ventanas de tiempo suave, empleada por los autores Duygu Tas et al <sup>113</sup>, Jan Fabian Ehmke et al <sup>114</sup> y D. Tas et al<sup>115</sup>, en donde se tienen en cuenta dos componentes para el cálculo del costo, los cuales son: los costos de operación y los costos de servicio.

Los costos de operación comprenden la cantidad de vehículos requeridos para cubrir la demanda de los clientes y la distancia total recorrida por cada uno de ellos. Y los costos de servicio comprenden el incumplimiento de las ventanas de tiempo de los clientes (llegadas tarde o temprano).

Los costos usados por los autores anteriormente mencionados se muestran en la Tabla 15:

**Tabla 15. Definición de los Costos del modelo**

Definición del Costo	Valor
Cd: Costo por demora	1
Ca: Costo por antelación	0.1
Cpv: Costo por usar el vehículo	400
Cp_ud: Costo por unidad de distancia recorrida	1

Para la presente investigación se utilizan los mismos valores de costo.

---

<sup>113</sup> TAS, D. et al. Op. cit., pp. 66-83.

<sup>114</sup> EHMKE, J. et al. Op. cit., pp-539-550-

<sup>115</sup>TAS, D. et al. Op.cit., pp. 789-799.

## 15. DEFINICIÓN DEL NÚMERO DE ESCENARIOS

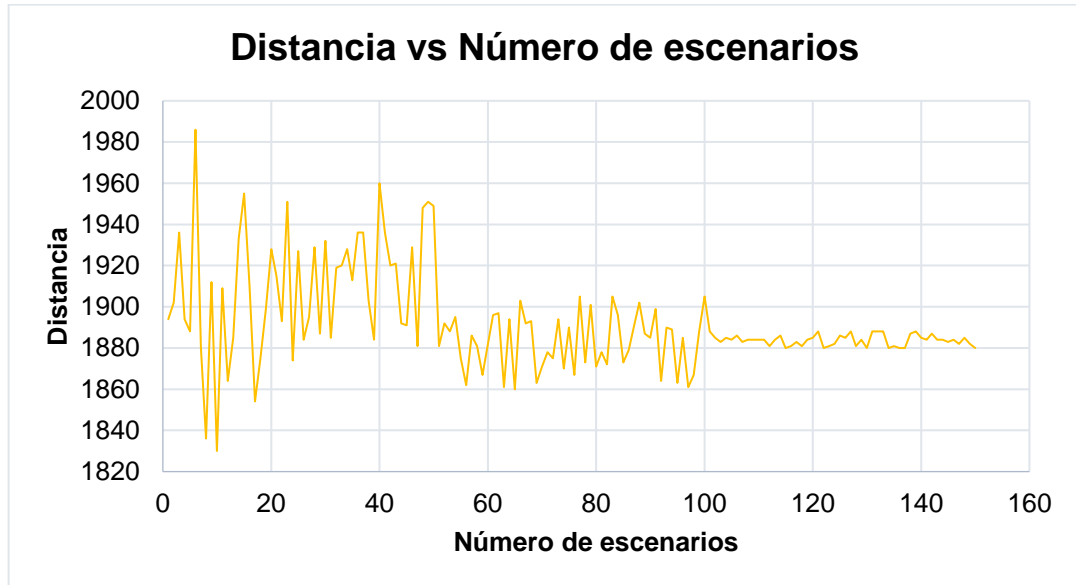
La variante del Problema de Ruteo de Vehículos que se trata en esta investigación considera tiempos de viaje estocásticos, por lo cual los resultados que se presentan están ligados a cierta variabilidad. Como se mencionó anteriormente, la estrategia que se usa para introducir la naturaleza estocástica del problema al algoritmo, es la definición de cierta cantidad de escenarios, en los cuales se evalúa la calidad de la solución y se concluye con base en el valor esperado de esta.

Lo ideal en toda simulación es que las variables alcancen el estado estacionario para que se reduzca el “ruido” en la toma de decisiones. Es por ello que la definición del número de escenarios cobra suma importancia en la implementación del algoritmo. Al considerar un número pequeño de escenarios, el costo se verá altamente afectado por la variabilidad, debido a la presencia de picos (valores máximos y mínimos de costo) que distorsionan la calidad de la solución, no obstante, el algoritmo mejorará en términos de eficiencia (menor tiempo computacional); mientras que al considerar un número grande de escenarios se disminuye la variabilidad y el tiempo computacional crecerá significativamente. Por tanto, se hace necesario realizar un análisis que permita determinar de forma experimental, la cantidad adecuada de escenarios a fin de encontrar el equilibrio entre estabilidad en el costo total esperado y el tiempo computacional.

Para determinar el efecto del número de escenarios en el costo de la solución y en el tiempo computacional, se llevaron a cabo varias corridas para 50 y 100 escenarios, considerando 150 iteraciones para ambos casos (Ver Tabla 16 y 17). La selección de estos valores para el análisis (50 y 100 escenarios) es soportada en la figura 16, donde se observa que con un número de escenarios mayor a 100, las fluctuaciones en el costo son insignificantes, mientras que entre 50 y 100 escenarios, existen variaciones que valen la pena considerar.

El análisis que se lleva a cabo se ejecuta en la instancia R101 de Solomon.

**Figura 16. Gráfica Distancia Vs Número de escenarios**



Para 50 escenarios:

**Tabla 16. Resultados del costo y tiempo computacional para 50 escenarios**

Corrida	Costo	Tiempo
1	1954,643581	11667,56977
2	1926,81224	11994,0823
3	2020,54789	12015,9452
4	1941,69852	11496,1624
5	1966,36842	11121,4821
<b>Promedio</b>	1962,01413	11659,04835
<b>Desviación Estándar</b>	35,88656769	372,5604077

Para 100 escenarios:

**Tabla 17. Resultados del costo y tiempo computacional para 100 escenarios**

<b>Corrida</b>	<b>Costo</b>	<b>Tiempo</b>
1	1900,452369	20402,3659
2	1864,96358	21122,3659
3	1902,36548	21063,65874
4	1869,58742	20869,95874
5	1878,02547	20614,2598
<b>Promedio</b>	1883,078864	20814,52182
<b>Desviación Estándar</b>	17,38916041	304,0920147

Al analizar los datos obtenidos en las corridas, se evidencia que para 50 escenarios el promedio de la distancia recorrida es de 1962.01413 con una desviación estándar de 35.88656 y para 100 escenarios, estos datos corresponden a 1883.078864 y 17.38916041, respectivamente. A partir de esto se concluye que a medida que aumenta el número de escenarios mejora la calidad de la solución en un 4.023% y disminuye la variabilidad en la distancia recorrida en un 51.54%.

Ahora bien, el tiempo computacional para 50 escenarios es de 11659.04835 con desviación estándar de 372.5604077 segundos y de 20814.52182 y 304.0920147 segundos, para 100 escenarios. Con base en estos datos se concluye que el tiempo computacional aumenta en un 78%, al evaluar el costo de las soluciones con 100 escenarios, causando solo una disminución en la variabilidad de 18.37% con respecto a 50 escenarios.

Teniendo en cuenta lo anterior, se decide tomar 50 escenarios para la evaluación del costo, debido a que el aumento en el número de escenarios altera significativamente la eficiencia (tiempo computacional) del algoritmo de manera negativa, mientras que la eficacia del algoritmo (distancia recorrida) no mejora de

forma relevante. Por lo cual se evita un desperdicio de recursos, al adoptar esta consideración.

## 16.RESULTADOS COMPUTACIONALES

El algoritmo propuesto en esta investigación para la solución al problema de ruteo de vehículos con tiempos de viaje estocásticos y ventanas de tiempo suave, es probado en las instancias introducidas por Solomon<sup>116</sup> en 1987 para el VRPTW.

Se seleccionan los tres primeros problemas de cada clase de instancia (RC1, RC2, R1, R2, C1, C2) para un total de 18 tests de prueba.

Cada problema se corre 5 veces para un valor fijo de 150 iteraciones. Los tiempos de viaje entre clientes son modelados a partir de una distribución uniforme. El algoritmo es implementado en Matlab versión R2014a, en un equipo con procesador Intel ® Core ™ I5 con 6 Gb de memoria RAM instalada.

En la tabla 18 se muestra el resultado (distancia recorrida) obtenido en las 5 corridas realizadas para cada instancia, el promedio, la mejor solución conocida y la brecha entre esta y la solución arrojada por el algoritmo implementado

De los 18 problemas tratados en la presente investigación, el algoritmo implementado logra resultados óptimos en los problemas R101 y RC102, alcanzando el mejor valor conocido en la literatura.

Para los problemas R102, R201, RC101, RC201, RC202 y RC203, los resultados arrojados por el algoritmo son valores muy cercanos a los conocidos en la literatura con un diferencial (gap) menor al 10%.

---

<sup>116</sup> SOLOMON., Op. cit; p.254-265.

Para los demás problemas, la implementación del algoritmo muestra resultados poco efectivos (de baja calidad), en comparación con los mejores conocidos en la literatura, puesto que se alejan de estos en porcentajes mayores a 10.

En los resultados obtenidos para las instancias seleccionadas, se puede observar que la desviación entre los valores de distancia de las soluciones arrojadas por el algoritmo implementado y los mejores valores de distancia conocidos en la literatura no supera el 22.426%.

También es posible concluir que las peores respuestas se encuentran en las clases C1 y C2, en las cuales los clientes se encuentran ubicados en clústeres. Esto sugiere que el algoritmo no se desempeña de forma eficaz cuando los clientes están agrupados.

Para las instancias con ventanas de tiempo estrechas y capacidad del vehículo de 200 unidades (R1 y RC1) el algoritmo muestra buenos resultados computacionales, esto permite inferir que la presencia de aleatoriedad (total o parcial) en la distribución geográfica de los clientes, favorece la consecución de buenas soluciones. Del mismo modo, el algoritmo alcanza mejores resultados, cuando en la programación de las rutas se asignan menor cantidad de clientes para ser servidos por un vehículo.

**Tabla 18. Resultados Computacionales del algoritmo propuesto**

Problema	Corridas					Media		Mejor Conocida		Gap
	1	2	3	4	5					
C101	1005,50014	1002,84755	1007,99524	995,055543	1007,65538	1003,81077	11	828,93664	10	21,0962
C102	1008,8905	1011,65914	1003,01582	1006,7021	1006,24619	1007,30275	12	828,93664	10	21,51746
C103	1002,76972	1013,29442	1017,30225	1018,35389	1017,1118	1013,76642	12	828,06464	10	22,426
R101	1651,19931	1649,9975	1650,53219	1652,29031	1649,98251	1650,80036	19	1650,79864	19	0,00010443
R102	1605,68833	1605,82215	1606,41689	1603,293	1606,11952	1605,46798	17	1486,85859	17	7,97718
R103	1487,63616	1486,08495	1499,94243	1489,17445	1499,65804	1492,49921	15	1292,67493	13	15,4582
RC101	1792,3342	1799,5632	1783,45916	1788,53303	1795,84688	1791,94729	14	1696,94871	14	5,5982
RC102	1554,9704	1555,01301	1554,82046	1555,00732	1553,9902	1554,76028	12	1554,74684	12	0,00086432
RC103	1422,69855	1437,69983	1422,44622	1426,8873	1426,206	1427,18758	13	1261,67143	11	13,1188
C201	727,661524	722,118642	724,532192	722,213291	722,189629	723,743056	4	591,55626	3	22,3456
C202	698,089627	695,900869	686,755409	692,599985	682,845222	691,238222	2	591,55626	3	16,8508
C203	667,670956	662,013429	660,163057	667,552721	665,122999	664,504632	4	591,17315	3	12,4044
R201	1332,73161	1326,8367	1325,44783	1332,62516	1336,3948	1330,80722	4	1252,37074	4	6,26304
R202	1359,67299	1358,37403	1369,64754	1358,8388	1354,02313	1360,1113	4	1191,70252	3	14,13178
R203	1076,69846	1069,54885	1071,3339	1076,86757	1062,51197	1071,39215	2	939,50286	3	14,0382
RC201	1467,73347	1463,00334	1459,43168	1455,88141	1466,16051	1462,44208	4	1406,93961	4	3,944908
RC202	1405,01615	1402,17698	1399,73794	1406,64264	1408,05608	1404,32596	3	1365,64462	3	2,83246
RC203	1147,76933	1155,37343	1160,04426	1152,84384	1154,1947	1154,04511	2	1049,62383	3	9,948448

En los resultados obtenidos para las instancias seleccionadas, se puede observar que la desviación entre los valores de distancia de las soluciones arrojadas por el algoritmo implementado y los mejores valores de distancia conocidos en la literatura no supera el 22.426%.

También es posible concluir que las peores respuestas se encuentran en las clases C1 y C2, en las cuales los clientes se encuentran ubicados en clústeres. Esto sugiere que el algoritmo no se desempeña de forma eficaz cuando los clientes están agrupados.

Para las instancias con ventanas de tiempo estrechas y capacidad del vehículo de 200 unidades (R1 y RC1) el algoritmo muestra buenos resultados computacionales, esto permite inferir que la presencia de aleatoriedad (total o parcial) en la distribución geográfica de los clientes, favorece la consecución de buenas soluciones. Del mismo modo, el algoritmo alcanza mejores resultados, cuando en la programación de las rutas se asignan menor cantidad de clientes para ser servidos por un vehículo.

## 17. DEFINICIÓN DEL NÚMERO DE ITERACIONES

El proceso iterativo aplicado a los algoritmos de búsqueda pretende asegurar la convergencia del algoritmo a una solución cercana a la óptima. Por lo tanto, una interrupción temprana de la estrategia aplicada puede ocasionar soluciones de baja calidad.

Con el propósito de definir el número de iteraciones que garantice la convergencia del algoritmo a una buena solución para el VRPSTTW, se propone un análisis de varianza unifactorial, que estudie el efecto de los diferentes niveles del factor (número de iteraciones) sobre la variable de respuesta (distancia recorrida).

Los niveles seleccionados en el desarrollo de este análisis, se obtienen mediante la observación del comportamiento de la variable respuesta en cada uno de los problemas elegidos en la presente investigación, para un número fijo de iteraciones (150).

La figura 17 muestra los resultados obtenidos para la variable distancia recorrida a medida que aumentan las iteraciones en el problema R101 de las instancias de Solomon. Este análisis se llevó a cabo para el primer problema de cada grupo de instancias (aleatoria, clústeres, aleatoria-clústeres), completando 6 experimentos de prueba. De los cuales se infiere que, los cambios más relevantes en la variable respuesta se presentan en el rango de 30 a 100 iteraciones, razón por la cual el análisis se enfoca en valores fijos comprendidos en este rango. Los valores seleccionados son:

- ∴ Nivel 1: 30.
- ∴ Nivel 2: 50.
- ∴ Nivel 3: 70.
- ∴ Nivel 4: 90.

**Figura 17. Resultados Distancia Vs Número de Iteraciones para la instancia R101**



### 17.1 Análisis de la varianza para el número de iteraciones

El procedimiento adecuado para probar la igualdad de varias medias de una variable de salida sometida a diferentes tratamientos es el análisis de varianza<sup>117</sup>. Para el desarrollo de este experimento se consideran 4 tratamientos (30, 50, 70 y 90 iteraciones) de un solo factor aplicados al primer problema de cada grupo en las instancias de Solomon. Dado que los niveles son seleccionados a criterio de los autores, el análisis realizado es un análisis del modelo con efectos fijos. La respuesta observada de cada uno de los 4 tratamientos, es una variable aleatoria que para el presente caso es la distancia recorrida.

Con el propósito de garantizar que el ambiente en el que se apliquen estos tratamientos sea lo más uniforme posible, el experimento es llevado a cabo en un

---

<sup>117</sup> MONTGOMERY, Douglas. Diseño y Análisis de Experimentos. Segunda Edición. México: Limusa Wiley, 2004. 686 p. ISBN: 968-18-6156 – 6.

orden completamente aleatorio. El número de réplicas para cada nivel es 5, lo que totaliza 20 corridas para cada experimento. La tabla 19 muestra la secuencia en la que se llevaron a cabo las corridas.

**Tabla 19. Orden aleatorio de las corridas**

Corrida	Nivel del factor
1	1
2	4
3	2
4	3
5	2
6	4
7	1
8	1
9	4
10	2
11	3
12	1
13	3
14	3
15	4
16	3
17	2
18	4
19	2
20	1

La ecuación del modelo de efectos fijos que describe las observaciones es:

$$Y_{ij} = \mu + \tau_i + \varepsilon_{ij} \quad \left\{ \begin{array}{l} i = 1, 2, 3, 4. \\ j = 1, 2, 3, 4, 5. \end{array} \right.$$

Donde:

- $Y_{ij}$  es la observación  $ij$  – ésima.
- $\mu$  es la media global.
- $\tau_i$  es el efecto del tratamiento  $i$  – ésimo.
- $\varepsilon_{ij}$  es un componente del error aleatorio.

En este experimento se quiere comprobar si existe algún efecto significativo del factor número de iteraciones en la variable de salida: Distancia recorrida. La prueba de hipótesis que se plantea para ello es:

$$H_0: \mu_1 = \mu_2 = \mu_3 = \mu_4$$

$$H_1: \mu_i \neq \mu_j \text{ para el menos un par } (i, j)$$

Como se puede observar la hipótesis nula supone que la igualdad de las medias en los diferentes grupos es cierta, es decir, que todas las observaciones provienen de un único grupo cuya media y variabilidad es la misma que la de cualquiera de los grupos por separado<sup>118</sup>.

El análisis de varianza provee la información necesaria para probar la anterior hipótesis, puesto que considera la división de la variabilidad total en sus partes componentes. Las fuentes de variabilidad se deben a las diferencias entre los promedios de los tratamientos y el promedio global y a las diferencias de las observaciones dentro de los tratamientos y el promedio de los tratamientos.

Si al realizar la prueba ANOVA se obtiene un nivel de significancia bajo, la hipótesis nula que considera que en todos los grupos las medias son iguales, se rechaza. Esto no implica que cada una de las medias es significativamente diferente de cada

---

<sup>118</sup> BARÓN LOPEZ, F. & TELLEZ MONTIEL, F. Apuntes de Bioestadística [en línea]. <http://www.bioestadistica.uma.es/baron/apuntes/ficheros/cap05.pdf> [citado en 2 de julio de 2016].

una de las otras; simplemente implica que no todas las medias son iguales. Para conocer cuáles medias difieren de otras, se requieren pruebas adicionales.

### Instancia C101

**Tabla 20. Resultados Instancia C101**

<b>C101</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Promedio</b>
<b>30</b>	1029,30924	1028,7654	1032,90732	1026,76093	1030,76532	1029,70164
<b>50</b>	1005,50014	1002,84755	1007,99524	996,155543	1007,85538	1004,07077
<b>70</b>	1005,60981	1002,54067	1006,89982	999,88324	1004,7209	1003,93089
<b>90</b>	1005,88478	1003,79329	1004,57097	1002,46356	1003,34231	1004,01098

**Tabla 21. Análisis de Varianza Instancia C101**

<b>Tabla ANOVA para Distancia Recorrida por Número de iteraciones C101</b>					
<b>Fuente</b>	<b>GL</b>	<b>SC</b>	<b>MC</b>	<b>F</b>	<b>P</b>
<b>Iteración</b>	3	2476,39	825,46	85,6	0
<b>Error</b>	16	154,29	9,64		
<b>Total</b>	19	2630,68			

$s=3,105$ ;  $R\text{-cuad}=94,14\%$ ;  $R\text{-cuad. (ajustado)}=93.04\%$ .

### Instancia R101

**Tabla 22. Resultados Instancia R101**

<b>R101</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Promedio</b>
<b>30</b>	1900,92834	1907,98232	1918,00523	1902,21314	1921,00452	1910,02671
<b>50</b>	1876,13131	1872,48723	1874,00125	1889,64532	1857,14221	1873,88146
<b>70</b>	1780,45931	1761,21953	1732,45821	1713,21222	1726,54982	1742,77982
<b>90</b>	1651,19931	1649,9975	1650,53219	1652,29031	1649,98251	1650,80036

**Tabla 23. Análisis de Varianza Instancia R101**

Tabla ANOVA para Distancia Recorrida por Número de iteraciones R101					
Fuente	GL	SC	MC	F	P
Iteración	3	214862	71621	295,68	0
Error	16	3876	242		
Total	19	218737			

s=15,16; R-cuad.=98,23%; R-cuad. (ajustado)=97,90%.

### Instancia RC101

**Tabla 24. Resultados Instancia RC101**

RC101	1	2	3	4	5	Promedio
30	1887,52613	1875,4289	1877,02931	1882,15169	1870,91587	1878,61038
50	1792,3342	1799,5632	1783,45916	1788,53303	1795,84688	1791,94729
70	1785,44087	1790,56228	1798,9759	1794,8527	1789,80367	1791,92708
90	1793,7529	1796,74569	1790,74267	1789,54313	1789,95332	1792,14754

**Tabla 25. Análisis de Varianza Instancia RC101**

Tabla ANOVA para Distancia Recorrida por Número de iteraciones RC101					
Fuente	GL	SC	MC	F	P
Iteración	3	28125,5	9375,2	322,65	0
Error	16	464,9	29,1		
Total	19	28950,4			

s=5,390; R-cuad.=98,37%; R-cuad.(ajustado)=98,07%.

## Instancia C201

**Tabla 26. Resultados Instancia C201**

<b>C201</b>		<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Prom</b>
<b>30</b>		727,66152	722,11864	724,53219	722,21329	722,18963	723,74
<b>50</b>		725,12647	727,40429	726,07508	725,12558	720,23083	724,79
<b>70</b>		725,72641	727,85071	726,34584	721,32432	722,31983	724,71
<b>90</b>		721,96360	722,24878	723,90940	723,91838	723,73839	723,15

**Tabla 27. Análisis de Varianza Instancia C201**

<b>Tabla ANOVA para Distancia Recorrida por Número de iteraciones C201</b>						
<b>Fuente</b>	<b>GL</b>	<b>SC</b>	<b>MC</b>	<b>F</b>	<b>P</b>	
<b>Iteración</b>	3	9,37	3,12	0,57	0,641	
<b>Error</b>	16	87,35	5,46			
<b>Total</b>	19	96,72				

s=2,337; R-cuad.=9,69%; R-cuad. (ajustado)=0,00%.

## Instancia R201

**Tabla 28. Resultados Instancia R201**

<b>R201</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Promedio</b>
<b>30</b>	1496,23094	1487,02398	1485,05687	1492,16161	1494,42436	1490,97955
<b>50</b>	1376,00988	1358,09654	1372,15233	1376,57430	1365,20101	1369,60681
<b>70</b>	1332,73161	1326,8367	1325,44783	1332,62516	1336,3948	1330,80722
<b>90</b>	1329,98333	1330,92645	1327,98794	1331,85265	1332,89871	1330,72982

**Tabla 29. Análisis de Varianza Instancia R201**

Tabla ANOVA para Distancia Recorrida por Número de iteraciones R201					
Fuente	GL	SC	MC	F	P
Iteración	3	86354,2	28784,7	1055,63	0
Error	16	463,3	27,3		
Total	19	86790,4			

s=5,222; R-cuad.=99,50%; R-cuad. (ajustado)=99,40%.

### Instancia RC201

**Tabla 30. Resultados Instancia RC201**

RC201	1	2	3	4	5	Promedio
30	1601,28943	1598,72395	1594,63487	1603,45823	1605,08467	1600,63823
50	1543,21109	1557,75437	1548,01147	1552,62693	1558,01132	1551,92304
70	1501,24376	1509,99461	1499,8736	1499,72356	1500,50134	1502,26737
90	1467,73347	1463,00334	1459,43168	1455,88141	1466,16051	1462,44208

**Tabla 31. Análisis de Varianza Instancia RC201**

Tabla ANOVA para Distancia Recorrida por Número de iteraciones RC201					
Fuente	GL	SC	MC	F	P
Iteración	3	54008,4	18002,8	718,42	0
Error	16	400,9	25,1		
Total	19	54409,4			

s=5,066; R-cuad.=99,26%; R-cuad. (ajustado)=99,12%.

Las tablas 20, 22, 24, 26, 28 y 30 muestran los resultados del algoritmo para los diferentes niveles del factor en cada una de las corridas y las tablas 21, 23, 25, 27, 31 y 33 muestran su respectivo análisis de varianza.

La regla de decisión seleccionada para rechazar o aceptar la hipótesis nula es:

*Si Valor  $P < \alpha$ , se rechaza  $H_0$ .*

*Si Valor  $P > \alpha$ , no se rechaza  $H_0$ .*

De acuerdo con lo anterior, se concluye que existe una diferencia estadísticamente significativa entre las medias de cada uno de los problemas seleccionados, es decir, se rechazan las hipótesis nulas de todos los problemas tratados, a un nivel de confianza del 95%; a excepción del experimento C201, en el que el *valor  $p = 0.641$*  es mayor al nivel de significancia  $\alpha = 0.05$ , por lo que no se rechaza  $H_0$ .

Para determinar qué medias son significativamente diferentes de las otras se lleva a cabo una prueba de comparaciones múltiples de medias en cada uno de los experimentos.

### **Instancia C101**

**Tabla 32. Prueba Tukey Instancia C101**

#### **Comparaciones en parejas de Tukey**

Agrupar información utilizando el método de Tukey y una confianza de 95%

Iteración	N	Media	Agrupación
1	5	1029,70	A
2	5	1004,07	B
4	5	1004,01	B
3	5	1003,93	B

Las medias que no comparten una letra son significativamente diferentes.

Los resultados de la prueba Tukey para este problema muestran que para los niveles 2, 3 y 4 que corresponden a 50, 70 y 90 iteraciones no hay diferencias significativas entre las medias ya que comparten la agrupación B.

Por lo cual se establece que para este problema el número de iteraciones apropiado es de 50, ya que al aumentar el factor (número de iteraciones) no se evidencian cambios significativos en la variable de salida (distancia recorrida), por lo que no se justifica la utilización de recurso computacional.

### **Instancia R101**

**Tabla 33. Prueba Tukey Instancia R101**

**Comparaciones en parejas de Tukey**

Agrupar información utilizando el método de Tukey y una confianza de 95%

Iteración	N	Media	Agrupación
1	5	1910,03	A
2	5	1873,88	B
3	5	1742,8	C
4	5	1650,80	D

Las medias que no comparten una letra son significativamente diferentes.

Al analizar los resultados arrojados por la prueba se puede concluir que existen diferencias significativas entre todos los pares de medias, debido a que la alineación de las columnas difiere para todos los grupos. Sin embargo, al observar la media de cada uno de los tratamientos se evidencia que la menor distancia se encuentra en el nivel 4 que corresponde a 90 iteraciones, el cual coincide con el mejor valor conocido en la literatura, razón por la cual es posible concluir que en este nivel ya es posible detener la búsqueda y de este modo ahorrar recurso computacional.

## Instancia RC101

**Tabla 34. Prueba Tukey Instancia RC101**

### **Comparaciones en parejas de Tukey**

Agrupar información utilizando el método de Tukey y una confianza de 95%

Iteración	N	Media	Agrupación
1	5	1878,61	A
4	5	1792,15	B
2	5	1791,95	B
3	5	1791,93	B

Las medias que no comparten una letra son significativamente diferentes.

Para la instancia RC101, la prueba Tukey muestra que en los niveles 2, 3 y 4 que corresponden a 50, 70 y 90 iteraciones, no hay diferencias significativas entre las medias, debido a que los grupos se encuentran alineados.

Por lo cual es posible concluir que el algoritmo alcanza el estado estacionario con 50 iteraciones.

## Instancia R201

**Tabla 35. Prueba Tukey Instancia R201**

### **Comparaciones en parejas de Tukey**

Agrupar información utilizando el método de Tukey y una confianza de 95%

Iteración	N	Media	Agrupación
1	5	1600,64	A
2	5	1551,92	B
3	5	1502,27	C
4	5	1462,44	D

Las medias que no comparten una letra son significativamente diferentes.

La comparación entre pares de medias para esta instancia muestra que existen diferencias significativas entre todos los niveles, debido a que la alineación de las columnas difiere para todos los grupos. Sin embargo, al observar la media de cada uno de los tratamientos se evidencia que la menor distancia se encuentra en el nivel 4 que corresponde a 90 iteraciones. Para efecto de este experimento, se establece este nivel como el apropiado para obtener soluciones de bajo costo (menor distancia recorrida), no obstante, el hecho de que no exista alineación en ninguno de los niveles, permite suponer que no se ha alcanzado el estado estacionario y por tanto da lugar a la posibilidad de encontrar soluciones de mejor calidad considerando un mayor número de iteraciones.

### **Instancia RC201**

**Tabla 36. Prueba Tukey Instancia RC201**

**Comparaciones en parejas de Tukey**

Agrupar información utilizando el método de Tukey y una confianza de 95%

Iteración	N	Media	Agrupación
1	5	1490,98	A
2	5	1369,61	B
3	5	1330,81	C
4	5	1330,73	C

Las medias que no comparten una letra son significativamente diferentes.

Al observar los resultados arrojados por la prueba Tukey se concluye que el nivel apropiado en el número de iteraciones, para obtener soluciones de buena calidad sin desperdiciar recurso computacional es de 70, puesto que en la tabla se evidencia que los niveles 3 y 4 se encuentran alineados, lo que indica que no existen diferencias significativas entre las medias de estos tratamientos y son las que arrojan la menor distancia (menor costo) para el problema planteado.

## 18. CONCLUSIONES

- ∴ La implementación de un algoritmo que contiene dos metaheurísticas enfocadas en generar soluciones iniciales da lugar a que la metaheurística principal (Optimización por Enjambre de Partículas) ahorre esfuerzos en la búsqueda de soluciones de calidad, puesto que sus entradas son soluciones que han sido previamente mejoradas. Esto significa que el algoritmo requiere menor cantidad de iteraciones para converger a una solución y por lo tanto menor tiempo computacional.
- ∴ El algoritmo propuesto muestra un desempeño intermedio en la resolución del problema de ruteo de vehículos con tiempos de viaje estocásticos y ventanas de tiempo suave, puesto que solo alcanza resultados óptimos en 8 de las 18 instancias en las que fue evaluado.
- ∴ La estructura del algoritmo propuesto para resolver el VRPSTTW favorece la capacidad de explotación del algoritmo, pero limita su capacidad de exploración, este hecho está fuertemente influenciado por la relación existente entre la metaheurística principal (PSO) y la metaheurística de post optimización (Path Relinking), debido a que ambos procedimientos centran sus esfuerzos en acercarse a la mejor solución encontrada hasta el momento (Gbest) parcializando la búsqueda en el espacio de soluciones factibles.
- ∴ La implementación de Circle Restricted Local Search Moves favorece la eficiencia del algoritmo, dado que encamina su estrategia a mejorar los puntos críticos de la solución, es decir, a aquellos arcos cuya eliminación impacta en el costo de forma positiva.

- ∴ La utilización de una metaheurística que combina un nivel de codicia (utilización de algoritmos voraces) y aleatoriedad en la construcción de las rutas iniciales, permite que el algoritmo inicie su proceso de búsqueda con soluciones de calidad, provocando que se reduzcan los procedimientos de búsqueda posteriores.
- ∴ El algoritmo propuesto presenta un mejor desempeño en la programación de rutas cortas con clientes que mantienen la aleatoriedad en su distribución geográfica, es decir, con distribución totalmente aleatoria o parcialmente aleatoria (clústeres-aleatorios).
- ∴ La selección apropiada del número de iteraciones es un factor clave para la convergencia del algoritmo a soluciones de buena calidad sin desperdiciar recurso computacional. Del análisis estadístico realizado se observa que no todos los problemas requieren el mismo número de iteraciones para converger, y aquellos cuya convergencia se da de forma prematura no logran los resultados óptimos esperados.
- ∴ El tratamiento que se da al comportamiento estocástico en el problema de ruteo de vehículos influye directamente en la efectividad del método de solución empleado. En esta investigación se simulan escenarios donde se genera una distribución de probabilidad uniforme para los tiempos de viaje entre clientes en cada escenario, lo que permite evaluar el comportamiento aleatorio del problema de forma práctica y lógica.

## 19.RECOMENDACIONES

Para futuras investigaciones se recomienda:

- ∴ Establecer criterios de parada más rigurosos en la metaheurística Expanding Neighborhood Search, más específicamente en 2-opt Exchange con el fin de evitar el despilfarro de recurso computacional en modificaciones que no aportan a una mejora significativa en el costo.
- ∴ Considerar otras distribuciones de probabilidad para la generación de los tiempos de viaje entre clientes que se ajusten mejor a la realidad como la distribución Gamma.
- ∴ Llevar a cabo un análisis experimental que estudie el efecto de los parámetros sobre la variable respuesta con el fin de encontrar un arreglo que tenga en cuenta interacciones significativas, ajuste de parámetros en un rango determinado, entre otros.
- ∴ Incluir una condición de cumplimiento de las ventanas de tiempo en la generación de las subrutas que restrinja la excesiva penalización en el costo por incumplimiento, con el propósito de brindar mayor atención a los costos de servicio del problema; debido a que actualmente la generación de subrutas se lleva a cabo teniendo en cuenta únicamente las restricciones de capacidad del vehículo.
- ∴ Ampliar el campo de aplicación del híbrido desarrollado en la presente investigación, con el fin de estudiar la efectividad de este en otros problemas de la Ingeniería Industrial.

## BIBLIOGRAFÍA

ACOSTA NIÑO, Francia del Pilar; OSORIO OSORIO, Daniela Angélica. Un método evolutivo de colonia de hormigas para la solución del problema de ruteo de vehículos con demandas estocásticas. Tesis de pregrado de Ingeniería Industrial. Bucaramanga: Universidad Industrial de Santander, Facultad de Ingenierías Físico-Mecánicas, Escuela de Estudios Industriales y Empresariales, 2013.

BALSEIRO, S.R., LOISEAU, I., RAMONET, J. An Ant Colony algorithm hybridized with insertion heuristics for the Time Dependent Vehicle Routing Problem with Time Windows. *Computers & Operations Research* [en línea]. Junio, 2011. Vol. 38, no.6; p. 954-966 [citado en 1 agosto de 2015]. Disponible en ScienceDirect Research Databases.

BAÑOS NAVARRO, Raúl. Metaheurísticas Híbridas para Optimización Mono-objetivo y Multi-Objetivo: Paralelización y Aplicaciones. Trabajo de Grado (Doctor en Informática). Almería: Universidad de Almería. Departamento de Arquitectura de computadores y electrónica, 2006. 100p.

BAÑOS, Raúl; ORTEGA, Julio; GIL, Consolación; MÁRQUEZ, Antonio L. y DE TORO, Francisco. A hybrid meta-heuristic for multi-objective vehicle routing problems with time windows. *Computers & Industrial Engineering* [en línea]. Junio, 2013. Vol. 65, No. 2; p. 286-296, ISSN 0360-8352 [citado en 7 de agosto de 2015] Disponible en ScienceDirect Research Databases.

BARÓN LOPEZ, F. & TELLEZ MONTIEL, F. Apuntes de Bioestadística [en línea]. <http://www.bioestadistica.uma.es/baron/apuntes/ficheros/cap05.pdf> [citado en 2 de julio de 2016].

BASU, S.K. DESIGN Methods and Analysis of Algorithms. 2ed. New Delhi: PHI Learning Private Limite, 2013. 369 p.

BELHAIZA, Slim; HANSEN, Pierre y LAPORTE, Gilbert. A hybrid variable neighborhood tabu search heuristic for the vehicle routing problem with multiple time windows. *Computers & Operations Research*. [en línea]. Diciembre, 2014. Vol. 52, Part B; p. 269-281, ISSN 0305-0548. [citado en 9 de Agosto de 2015] Disponible en ScienceDirect Research Databases.

BLUM, C. y ROLI, A. Metaheuristics in combinatorial optimization: Overview and conceptual Comparison. *ACM Computing Surveys (CSUR)*. Septiembre 2003. Vol. 35, No. 3; p. 268-308.

BOUSSAÏD, Ilhem; LEPAGNOT, Julien y SIARRY, Patrick. A survey on optimization metaheuristics. *Information Sciences* [en línea]. Julio, 2013. Vol.237; p.82-117, ISSN 0020-0255 [citado en 10 de Agosto de 2015] Disponible en ScienceDirect Research Databases.

CHICANO, J. Metaheurísticas e Ingeniería del Software. Tesis doctoral. Málaga: Universidad de Málaga, 2007.

CHRISTOFIDES, Nicos. The Vehicle Routing Problem. R.A.I.R.O. *Recherche Opérationnelle*. 1976. Vol.10, No. 5. p.55-70.

CLARKE, G y WRIGHT, J. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points, Citado por LAPORTE, Gilbert. The vehicle routing problem. An overview of exact and approximate algorithms. *European Journal of Operational Research*. Junio 1992. Vol 59, no 3; p. 345-358, ISSN 0377-2217.

CONNORS, R y SUMALEE, A. A network equilibrium model with travellers' perception of stochastic travel times, Citado por ZHANG, Tao; CHAOVALITWONGSE, W.A y ZHANG, Yuejie. Scatter search for the stochastic travel-time vehicle routing problem with simultaneous pick-ups and deliveries. *Computers & Operations Research* [en línea]. Octubre, 2012. Vol. 39, No. 10; p. 2277-2290, ISSN 0305-0548 [citado en 5 de agosto de 2015] Disponible en ScienceDirect Research Databases.

CONSEJO PRIVADO DE COMPETITIVIDAD. Infraestructura, transporte y logística [en línea] <<http://www.compite.com.co/site/wp-content/uploads/2012/10/6-Infraestructura-Transporte-y-Logistica.pdf>> [citado en 10 de agosto de 2015].

CRUZ JIMÉNEZ, Camilo; GONZÁLEZ VILLAMIZAR, Ludy Juliana. Desarrollo de un algoritmo híbrido para la resolución del problema de ruteo de vehículos con entrega y recogida simultáneas (VRPSPD). Tesis de Pregrado de Ingeniería Industrial. Bucaramanga: Universidad Industrial de Santander, Facultad de Ingenierías Físico-Mecánicas, Escuela de Estudios Industriales y Empresariales, 2013.

DANESHZAND, Farzaneh. The Vehicle-Routing Problem, In *Logistics Operations and Management*. Londres: Reza Zanjirani FarahaniShabnam RezapourLaleh Kardar, (2011). p. 127-153.

DANTZIG, G. y RAMSER, J. The Truck Dispatching Problem. *Management Science*. Octubre, 1959. Vol. 6, no. 1; p. 80-91.

DORIGO, Marco y STÜTZLE, Thomas. *Ant Colony Optimization*. Estados Unidos: MIT Press, 2004. 321 p.

DUARTE MUÑOZ, Abraham. *Metaheurística*. España: S.L.-DYKINSON, 2007. 220p.

EBERHART, R. C. y SHI, Y. Particle swarm Optimization: Developments, applications and resources. *Proc. IEEE Int. Conf. Evolutionary Computation*. 2001. Vol. 9; p. 81–86.

ERICKSON, Jeff. Algorithms: NP-Hardness [en línea]. <<http://www.cs.uiuc.edu/~jeffe/teaching/algorithms>> [citado en 15 de junio de 2016].

Geetha, S et al. A Hybrid Particle Swarm Optimization with Genetic Operators for Vehicle Routing Problem. *Journal of advances in information technology* [en línea]. Noviembre, 2010. Vol.1, No. 4; p. 181-188. [Citado en 16 de junio de 2016] Disponible en SpringerLink Research Databases.

GELVES TELLO, Natalia Alejandra; MORA MORENO, Ricardo Andrés. Solución del problema de ruteo de vehículos con demandas estocásticas (VRPSD) mediante la metaheurística optimización por espiral. Tesis de pregrado de Ingeniería Industrial. Bucaramanga: Universidad Industrial de Santander, 2015.

GENDREAU, M. y HO, S. Path Relinking for the vehicle routing problem. *Journal of Heuristics* [en línea]. 2006. Vol.12; p. 55-72. [citado en 22 de Agosto de 2015] Disponible en SpringerLink Research Databases.

GENDREAU, Michel; LAPORTE, Gilbert y SÉGUIN, René. Stochastic vehicle routing. *European Journal of Operational Research* [en línea]. Enero, 1996. Vol. 88, no. 1 (Ene. 1996); p. 3-12, ISSN 0377-2217. [citado en 23 de Agosto de 2015] Disponible en ScienceDirect Research Databases.

Ghosh, A. y Gorai, A. Hue-Preserving Color Image Enhancement Using Particle Swarm Optimization. En: Dehuri, Satchidananda, et al. *Integration of Swarm Intelligence and Artificial Neural Network*. Singapore: World Scientific Publishing, 2011. p. 157-168.

GONZÁLEZ VARGAS, Guillermo, y GONZÁLEZ ARISTIZÁBAL, Felipe. Metaheurísticas aplicadas al ruteo de vehículos. Un caso de estudio: Parte 1: formulación del problema. *Ingeniería e Investigación*. 2006. Vol. 26, no. 3; p. 149-156.

HILLIER, Frederick S.; LIEBERMAN, Gerald J. Introducción a la investigación de operaciones. 9° Ed. Mc Graw Hill. 2010, pp. 570-580. (ISBN: 978-607-15-0308-4).

IGLESIAS MARTÍNEZ, José, et al. Complejidad Computacional. Grado Ingeniería informática. Teoría de Autómatas y Lenguas formales. [en línea] <<http://ocw.uc3m.es/ingenieria-informatica/teoria-de-automatasy-lenguajes-formales/material-de-clase-1/tema-8-complejidad-computacional>> En: Universidad Carlos III de Madrid. [Citado el 12 de Julio del 2014].

JIANG, Dengying.; JIAN, Wenxia. y HUANG, Zhangcan. A Hybrid Algorithm for Vehicle Routing Problem with Time Windows. *Springer Berlin Heidelberg* [en línea]. Diciembre, 2008. Vol. 5370; p. 198-205. [Citado en 1 de agosto de 2015] Disponible en SpringerLink Research Databases.

JOYANES, Luis, et al. Estructura de datos en C. Madrid: McGraw-Hill España, 2005. 440 p.

KALLEHAUGE, Brian, et al. Vehicle Routing Problem with Time Windows. *Gerad column generation*. 2005. United States of America: Springer, 331p.

KAUR, Avneet & KAUR, Mandeep. Dealing with Boundary Constraint Violations in Particle Swarm optimization with Aging Leader and Challengers (ALC-PSO). *International Journal of Computer Applications* [en línea]. Julio, 2015. Vol. 121, no. 11; p.13-19 [citado en 2 de junio de 2016] Disponible en ProQuest Research Databases.

KENNEDY, J. y EBERHART, R. Particle Swarm Optimization. *Neural Networks* [datos en línea]. Diciembre, 1995. Vol. 4; p. 1942-1948 [citado en 21 de Junio de 2016] Disponible en IEEE Xplore Digital Library.

LAMBERT, V., LAPORTE, G; LOUVEAUX, F. Designing collection routes through bank branches. *Computers & Operations Research*. 1993. Vol. 20, No. 7; p. 783–791.

LAPORTE, G; LOUVEAUX, F; MERCURE, H. y JAILLET. VRP homogéneo, Citado por, ROCHA, L.; GONZÁLEZ, C. y ORJUELA, J. Una revisión al estado del arte del problema de ruteo de vehículos: Evolución histórica y métodos de solución. *Ingeniería*. 2011. Vol.16, no. 2; p. 35-55.

LAPORTE, G; LOUVEAUX, F; Y MERCURE, H. The vehicle routing problem with stochastic travel times, Citado por LI, Xiangyong; TIAN, Peng y LEUNG, Stephen C.H. Vehicle routing problems with time windows and stochastic travel and service times: Models and algorithm. *International Journal of Production Economics* [en línea]. Mayo, 2010. Vol. 125, No. 1; p.137-145, ISSN 0925-5273. [citado en 3 de agosto de 2015] Disponible en ScienceDirect Research Databases.

LECLUYSE, C; WOSENSEL, T. Van y PEREMANS, H. Vehicle routing with stochastic time-dependent travel times. *4OR: A Quarterly Journal of Operations Research* [en línea]. Noviembre, 2009. Vol. 7, No. 4; p. 363–377 [citado en 5 de Agosto de 2015] Disponible en SpringerLink Research Databases.

LI, Jin. Vehicle routing problem with time windows for reducing fuel consumption. *Journal of Computers*. Diciembre, 2012. Vol. 7, No. 12; p. 3020-3027.

LIU, Bo, WANG, Ling y JIN, Yi-hui. An effective hybrid PSO-based algorithm for flow shop scheduling with limited buffers. *Computers & Operations Research*. 2008. Vol. 35, pp. 2791–2806. (DOI 10.1016/j.cor.2006.12.013).

LOZADA DÍAZ, Adriana y CADENA GONZÁLEZ, Ricardo Andrés. Solución del problema de ruteo de vehículos con ventanas de tiempo (VRPTW) mediante métodos heurísticos. Tesis de pregrado de Ingeniería Industrial. Bucaramanga: Universidad Industrial de Santander, Facultad de Ingenierías Fisico-Mecánicas, Escuela de Estudios Industriales y Empresariales, 2012.

MACEDO, Rita; ALVES, Cláudio; HANAFI, Saïd; JARBOUJ, Bassem; MLADENović, Nenad; RAMOS, Bruna y DE CARVALHO J.M. Valério. Skewed general variable neighborhood search for the location routing scheduling problem. *Computers & Operations Research* [en línea]. Septiembre, 2015. Vol. 61; p.143-152, ISSN 0305-0548 [citado en 11 de Agosto de 2015] Disponible en ScienceDirect Research Databases.

MARINAKIS, Y., MARINAKI, M. y DOUNIAS, G. A hybrid particle swarm optimization algorithm for the vehicle routing problema. *Engineering Applications of Artificial Intelligence* [en línea]. Junio, 2010. Vol. 23, no. 4; p. 463-472, ISSN 0952-1976 [citado en 22 de Agosto de 2015] Disponible en ScienceDirect Research Databases.

MARINAKIS, Y; MIGDALAS, A y PARDALOS, P. Multiple phase neighborhood Search—GRASP based on Lagrangean relaxation, random backtracking Lin–Kernighan and path relinking for the TSP. *Journal of Combinatorial Optimization* [en línea].Febrero, 2009. Vol. 17, No. 2; p.134-156. [citado en 20 de Agosto de 2015] Disponible en SpringerLink Research Databases.

MARINAKIS, Yannis. Multiple Phase Neighborhood Search-GRASP for the Capacitated Vehicle Routing Problem. *Expert Systems with Applications* [en línea].

Junio, 2012. Vol. 39, no. 8; p. 6807-6815. [Citado en 22 de Agosto de 2015]. Disponible en Science Direct Research Databases.

MARINI, Federico y WALCZAK, Beata. Particle Swarm Optimization (PSO): A Tutorial. *Chemometrics and Intelligent Laboratory Systems* [en línea]. Diciembre, 2015. Vol. 149 parte B; p. 153-165 [citado en 19 de junio de 2016] Disponible en ScienceDirect Research Databases.

MARTÍ, R. Procedimientos metaheurísticos en optimización combinatoria. *Departament d'Estadística i Investigació Operativa*. Facultat de Matemàtiques. Universitat de València. 2003. Vol. 1, No. 1; p. 1-60.

MOGHADDAM, Babak Farhang; RUIZ, Rubén y SADJADI, Seyed Jafar. Vehicle routing problem with uncertain demands: An advanced particle swarm algorithm. *Computers & Industrial Engineering* [en línea]. Febrero, 2012. Vol. 62, no. 1; p. 306-317, ISSN 0360-8352. [citado en 23 de Agosto de 2015] Disponible en ScienceDirect Research Databases.

MONTGOMERY, Douglas. Diseño y Análisis de Experimentos. Segunda Edición. México: Limusa Wiley, 2004. 686 p. ISBN: 968-18-6156 – 6.

NAKARIYAKUL, Songyot. A comparative study of suboptimal branch and bound algorithms. *Information Sciences* [en línea]. Septiembre, 2014. Vol. 278, No. 10; p. 545-554, ISSN 0020-0255 [citado en 10 de Agosto de 2015] Disponible en ScienceDirect Research Databases.

OCHOA, Carlos Alberto, et al. Logistics Management and Optimization through Hybrid Artificial Intelligence Systems. United States of America: IGI Global, 2012. 511p.

PARK ,Yang-Byung y SONG, Sung-Hu. Vehicle scheduling problems with time-varying speed. *Computers & Industrial Engineering* [en línea]. Diciembre, 1997. Vol. 33, No. 3–4; p. 853–856 [citado en 2 de Agosto de 2015] Disponible en ScienceDirect Research Databases.

PÉREZ LÓPEZ, Jesús Ramón. Contribución a los métodos de optimización basados en procesos naturales y su aplicación a la medida de antenas en campo próximo. Trabajo de Grado (Doctor). Santander: Universidad de Cantabria. Departamento de Ingeniería de Comunicaciones. 2005, 68 p.

POLI, R., KENNEDY, J., & BLACKWELL, T. Particle swarm optimization. *Swarm Intell* [en línea]. Junio, 2007. vol. 1; p. 33-57. [citado en 19 de agosto de 2015] Disponible en SpringerLink Research Databases.

PORNSING, CHOOSAK. A Particle Swarm Optimization for the Vehicle Routing Problem. Tesis Doctoral (Ingeniería Industrial). Estados Unidos: Universidad de Rhode Island. 2014. Disponible en línea en <[http://digitalcommons.uri.edu/oa\\_diss](http://digitalcommons.uri.edu/oa_diss)>.

PULLEN, H Y WEBB, M. A Computer Application to a Transport Scheduling Problem, Citado por, ROCHA, L.; GONZÁLEZ, C. y ORJUELA, J. Una revisión al estado del arte del problema de ruteo de vehículos: Evolución histórica y métodos de solución. *Ingeniería*. 2011. Vol.16, no. 2; p. 35-55.

RESENDE, Mauricio y RIBEIRO, Celso. GRASP with Path Relinking: Recent advances and applications. *Operations Research/Computer Science Interfaces Series* [en línea]. 2005. Vol. 32; p. 29-63. [citado en 20 de agosto de 2015] Disponible en SpringerLink Research Databases.

SAVELSBERGH, M.W.P. Local Search in Routing Problems with Time Windows. *Annals of Operations Research* [en línea]. Julio, 1985. Vol. 4, No. 1; p. 285-305 [citado en 10 de Agosto de 2015] Disponible en SpringerLink Research DataBases.

SHAO, Z; GAO, S y WANG, S. A hybrid particle swarm optimization algorithm for vehicle routing problem with stochastic travel time. *Fuzzy information and engineering* [en línea]. 2009. Vol. 54; p. 566–74 [citado en 4 de Agosto de 2015] Disponible en SpringerLink Research Databases.

SIMON, Dan. *Evolutionary Optimization Algorithms*. New Jersey, USA: John Wiley & Sons, 2013. 727 p.

SOLOMON, M. Algorithms for the vehicle routing and scheduling problems with time window Constraints, Citado por CHENG, Chi-Bin, WANG, Keng-Pin. Solving a vehicle routing problem with time windows by a decomposition technique and a genetic algorithm. *Expert Systems with Applications* [en línea]. Mayo, 2009. Vol. 36, no. 4; p. 7758-7763, ISSN 0957-4174. [citado en 5 de agosto de 2015] Disponible en ScienceDirect Research Databases.

SOLOMON, Marius. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research Society of America* [en línea]. Abril, 1987. Vol. 35, no. 2; p. 258-259 [citado en 15 de junio de 2016] Disponible en JSTOR Research Databases.

STEWART, W y GOLDEN, B. Stochastic vehicle routing: a comprehensive approach, Citado por TAŞ, Duygu; DELLAERT, Nico; WOENSEL, Tom van y DE KOK, Ton. Vehicle routing problem with stochastic travel times including soft time windows and service costs. *Computers & Operations Research* [en línea]. Enero, 2013. Vol. 40, No. 1; p. 214-224, ISSN 0305-0548 [citado en 6 de agosto de 2015] Disponible en ScienceDirect Research Databases.

TAHA, Hamdy A. Investigación de operaciones. 9° Ed. México: Pearson, 2012. pp. 397-402. (ISBN: 978-607-32-0796-6).

TAŞ, D; GENDREAU, M; DELLAERT, N; WOENSEL, T. van y DE KOK, A.G. Vehicle routing with soft time windows and stochastic travel times: A column generation and branch-and-price solution approach. *European Journal of Operational Research* [en línea]. Agosto, 2014. Vol. 236, No. 3; p.789-799, ISSN 0377-2217 [citado en 6 de Agosto de 2015] Disponible en ScienceDirect Research Databases.

TAŞ, D; GENDREAU, M; DELLAERT, N; WOENSEL, T. van y DE KOK, A.G. Vehicle routing with soft time windows and stochastic travel times: A column generation and branch-and-price solution approach. *European Journal of Operational Research* [en línea]. Agosto, 2014. Vol. 236, No. 3; p.789-799, ISSN 0377-2217 [citado en 6 de Agosto de 2015] Disponible en ScienceDirect Research Databases.

TAŞ, Duygu; DELLAERT, Nico; WOENSEL, Tom van y DE KOK, Ton. The time-dependent vehicle routing problem with soft time windows and stochastic travel times. *Transportation Research Part C: Emerging Technologies* [en línea]. Noviembre, 2014. Vol. 48; p.66-83, ISSN 0968-090X [citado en 2 de agosto de 2015] Disponible en ScienceDirect Research Databases.

TAŞ, Duygu; DELLAERT, Nico; WOENSEL, Tom van y DE KOK, Ton. Vehicle routing problem with stochastic travel times including soft time windows and service costs. *Computers & Operations Research* [en línea]. Enero, 2013. Vol. 40, no. 1; p. 214-224, ISSN 0305-0548. [citado en 23 de agosto de 2015] Disponible en ScienceDirect Research Databases.

TOTH, P. y VIGO, D. The vehicle routing problema. Philadelphia: SIAM Monographs on Discrete Mathematics and Applications, 2002. 363 p.

Universidad Autónoma de Nuevo León. Investigación de Operaciones en Acción: Heurísticas para la Solución del TSP [En línea]. <http://yalma.fime.uanl.mx/~roger/work/Papers/article/article-inge-2000.pdf> [citado en 1 de julio de 2016].

UNIVERSIDAD DEL ROSARIO. Elevados costos de transporte [en línea]. <<http://www.urosario.edu.co/Universidad-Ciencia-Desarrollo/ur/Fasciculos-Anteriores/Tomo-II-2007/Fasciculo-15/ur/Elevados-costos-de-transporte/>> [citado en 10 de Agosto de 2015].

VAN LANDEGHEM, H. A bi-criteria heuristic for the vehicle routing problem with time windows. Citado por, ROCHA, L.; GONZÁLEZ, C. y ORJUELA, J. Una revisión al estado del arte del problema de ruteo de vehículos: Evolución histórica y métodos de solución. *Ingeniería*. 2011. Vol.16, no. 2; p. 35-55.

WANG, W., WU, B., ZHAO, Y. y FENG, D. Particle Swarm Optimization for Open Vehicle Routing Problem En: Kang Li Et al. Computational Intelligence. Kunming, China: Springer, 2006. p. 999-1007.

WOENSEL, Tom Van; KERBACHE, Laoucine; PEREMANS, Herbert y VANDAELE, Nico. A queueing framework for routing problems with time-dependent travel times. *Journal of Mathematical Modelling and Algorithms* [en línea]. Marzo, 2007. Vol. 6, No. 1; p. 151–173 [citado en 4 de agosto de 2015] Disponible en SpringerLink Research Databases.

XIE, B. Research on stochastic vehicle routing problems. China, 2003, Ph.D. thesis. Xinan Jiaotong University.

YANG, X. Introduction to Mathematical Optimization: From Linear Programming to Metaheuristics. Reino Unido: Cambridge, GBR: Cambridge International Science Publishing, 2008. 150 p.

YANG, Xin She. Engineering Optimization: An Introduction with Metaheuristic Applications. New Jersey, USA: John Wiley & Sons, 2010. 343 p.