

**“DESARROLLO DE UN ALGORITMO HÍBRIDO PARA LA RESOLUCIÓN DEL  
PROBLEMA DE RUTEO DE VEHÍCULOS CON ENTREGA Y RECOGIDA  
SIMULTÁNEAS (VRPSPD)”**

**CAMILO CRUZ JIMÉNEZ  
LUDY JULIANA GONZÁLEZ VILLAMIZAR**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICOMECÁNICAS  
ESCUELA DE ESTUDIOS INDUSTRIALES Y EMPRESARIALES  
BUCARAMANGA**

**2013**

**“DESARROLLO DE UN ALGORITMO HÍBRIDO PARA LA RESOLUCIÓN DEL  
PROBLEMA DE RUTEO DE VEHÍCULOS CON ENTREGA Y RECOGIDA  
SIMULTÁNEAS (VRPSPD)”**

**CAMILO CRUZ JIMÉNEZ (2080374)  
LUDY JULIANA GONZÁLEZ VILLAMIZAR (2081924)**

**Tesis de Grado en modalidad “Trabajo de investigación” presentado como  
requisito parcial para optar al título de Ingeniero Industrial**

**Director  
HENRY LAMOS DÍAZ  
MATEMÁTICO  
Ph. D. en Física-Matemática**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICOMECÁNICAS  
ESCUELA DE ESTUDIOS INDUSTRIALES Y EMPRESARIALES  
BUCARAMANGA**

**2013**

## **AGRADECIMIENTOS**

*Especiales agradecimientos al profesor Henry Lamos, por toda la enseñanza que recibimos de su parte.*

*A Silvia Galván y a Cesar Álvarez quienes nos compartieron su tiempo y su conocimiento.*

*A todos nuestros familiares y amigos que nos apoyaron durante todo este proceso.*

## TABLA DE CONTENIDO

INTRODUCCIÓN .....	17
1. ANTECEDENTES DEL PROYECTO.....	20
1.1 JUSTIFICACIÓN.....	20
1.2 PLANTEAMIENTO DEL PROBLEMA.....	21
1.3 OBJETIVOS.....	22
1.3.1 Objetivo General.....	22
1.3.2 Objetivos específicos.....	22
1.4 ALCANCE DEL PROYECTO.....	23
2. GENERALIDADES .....	24
2.1 SISTEMAS LOGÍSTICOS.....	24
2.2 LOGÍSTICA DE DISTRIBUCIÓN .....	26
2.3 LOGÍSTICA INVERSA .....	27
3. MARCO TEÓRICO .....	31
3.1 OPTIMIZACIÓN COMBINATORIA .....	31
3.1.1 Problemas Combinatorios.....	32
3.1.2 Complejidad Computacional .....	33
3.1.3 Optimización Matemática.....	35
3.2 PROBLEMA DE RUTEO DE VEHÍCULOS.....	36
3.2.1 Definición .....	36
3.3 PROBLEMA DE RUTEO DE VEHÍCULOS CON ENTREGA Y RECOLECCIÓN SIMULTÁNEAS .....	40
3.3.1 Formulación Matemática.....	41
3.4 MÉTODOS DE SOLUCIÓN PARA EL VRPSD .....	44
3.4.1 Métodos Exactos .....	45
3.4.2 Métodos Heurísticos .....	46

3.4.3	Métodos Metaheurísticos.....	48
3.4.4	Heurísticas de mejoramiento .....	51
3.4.5	Inteligencia Computacional.....	53
3.5	OPTIMIZACIÓN DE ENJAMBRE DE PARTÍCULAS (PSO) .....	58
3.5.1	Local Best PSO.....	62
3.5.2	Aspectos relacionados con la implementación del algoritmo PSO ....	64
3.5.3	Peso inercial y constantes de aceleración .....	69
3.5.4	Otros parámetros del algoritmo.....	72
3.5.5	Optimización de Enjambre de partículas con múltiples estructuras sociales: GLNPSO .....	73
4.	DISEÑO DE LA SOLUCIÓN .....	77
4.1	REPRESENTACIÓN DE LA SOLUCIÓN.....	78
4.2	DESCRIPCIÓN DE LA METAHEURÍSTICA GLNPSO .....	80
4.2.1	Heurística de Inserción más barata (Cheapest insertion heuristic) ....	84
4.2.2	Heurística 2-opt.....	94
4.3	BÚSQUEDA DE NÚMERO APROPIADO DE VEHÍCULOS .....	100
5.	RESULTADOS COMPUTACIONALES.....	102
5.1	ANÁLISIS DE VARIANZA DEL FACTOR NÚMERO DE ITERACIONES .....	109
5.2	ANÁLISIS FACTORIAL DE EFECTOS FIJOS.....	122
	CONCLUSIONES .....	132
	RECOMENDACIONES.....	135
	REFERENCIAS BIBLIOGRÁFICAS.....	137
	ANEXOS .....	144

## LISTA DE TABLAS

Tabla 4.	Matriz de distancias ejemplo problema .....	87
Tabla 5.	Matriz de distancias ejemplo para 2-opt .....	94
Tabla 6.	Instancias.....	103
Tabla 7.	Parámetros del GLNPSO.....	104
Tabla 8.	Problemas de 100 clientes.....	105
Tabla 9.	Problemas para 200 clientes .....	106
Tabla 10.	Problemas para 400 clientes .....	106
Tabla 11.	Comparación de Resultados para el VRPSPD con distintos enfoques 107	
Tabla 12.	Aleatorización de las corridas .....	110
Tabla 13.	Instancia c101.....	111
Tabla 14.	ANOVA para Desv_opt por Nro de iteraciones (c101).....	111
Tabla 15.	Grupos homogéneos c101.....	112
Tabla 16.	Contrastes entre medias C101 .....	112
Tabla 17.	Tabla ANOVA para Dev_opt por Nro_iteraciones (r201) .....	114
Tabla 18.	Grupos homogéneos r201 .....	114
Tabla 19.	Contrastes entre medias r201.....	115
Tabla 20.	ANOVA para Desv_opt por Nro de iteraciones (R121) .....	116
Tabla 21.	ANOVA para Desv_opt por Nro de iteraciones (C221) .....	117
Tabla 22.	Grupos homogéneos C221 .....	118
Tabla 23.	Contrastes entre medias C221 .....	118
Tabla 24.	Instancia R241 .....	119
Tabla 25.	ANOVA para Desv_opt por Nro de iteraciones (R241) .....	120
Tabla 26.	Tabla ANOVA para Desv_opt por Nro_iteraciones (C241) .....	121
Tabla 27.	Instancias para experimentación (capacidad vehículos).....	123
Tabla 28.	Niveles de Capacidad de vehículos .....	123
Tabla 29.	Análisis de Varianza para Desv opt - Suma de Cuadrados Tipo III ..	125

Tabla 30.	Prueba Rangos múltiples Desv Opt - Nro Clientes .....	125
Tabla 31.	Contrastes DesvOpt - Nro clientes.....	126
Tabla 32.	Rangos múltiples para DesvOpt-Capacidad .....	127
Tabla 33.	Contrastes DesvOpt - Capacidad .....	127
Tabla 34.	Análisis de Varianza para Desv k - Suma de Cuadrados Tipo III .....	128
Tabla 35.	Prueba Rangos Múltiples Desv K - Nro clientes .....	129
Tabla 36.	Contrastes DesvK - nro clientes.....	130

## LISTA DE ILUSTRACIONES

Ilustración 1.	Cadena de suministros.....	24
Ilustración 2.	Actividades logísticas .....	25
Ilustración 3.	Logística de Distribución .....	28
Ilustración 4.	Esquema Logística Inversa .....	30
Ilustración 5.	Representación gráfica de un problema de ruteo de vehículos..	37
Ilustración 6.	Variaciones del VRP.....	40
Ilustración 7.	Intercambio 2-opt.....	52
Ilustración 8.	Paradigmas de inteligencia computacional .....	55
Ilustración 9.	Actualización de la velocidad y posición de una partícula en un espacio bidimensional en el paradigma PSO .....	63
Ilustración 10.	Representación gráfica de una frontera aleatoria.....	68
Ilustración 11.	Representación gráfica de una frontera absorbente.....	68
Ilustración 12.	Representación gráfica de una frontera reflectora.....	69
Ilustración 13.	Rutas y puntos de orientación de rutas .....	79
Ilustración 14.	Vector posición de una partícula del enjambre .....	80
Ilustración 15.	Problema Ejemplo cheapest insertion .....	87
Ilustración 16.	cheapest insertion heuristic (it. 1) .....	89
Ilustración 17.	Cheapest insertion heuristic(it.2) .....	90
Ilustración 18.	Cheapest Insertion (it.3) .....	91
Ilustración 19.	Cheapest insertion ruta final .....	93
Ilustración 20.	Ruta a optimizar con 2-opt.....	96
Ilustración 21.	Ejemplo 2-opt (it. 1 - intercambio 1).....	97
Ilustración 22.	Gráfico de medias C101 .....	113
Ilustración 23.	Instancia r201 .....	113
Ilustración 24.	Gráfico de medias r201 .....	116
Ilustración 25.	Instancia R121.....	116
Ilustración 26.	Gráfico de medias C221 .....	119
Ilustración 27.	Gráfico de medias R241 .....	120

Ilustración 28.	Gráfico de medias C241 .....	122
Ilustración 29.	Gráfico de Medias - Nro clientes.....	126
Ilustración 30.	Gráfico de medias Desv Opt - capacidad .....	127
Ilustración 31.	Gráfico de Interacciones para Desv_opt.....	128
Ilustración 32.	Gráfico de Medias - Nro clientes.....	130
Ilustración 33.	Gráfico de interacciones Desv_K.....	131

## LISTA DE ANEXOS

Anexo A.	Código de programación .....	144
Anexo B.	Instancias 'VRPSD' .....	162
Anexo C.	Resultados computacionales .....	272
Anexo D.	Artículo de investigación .....	288

## RESUMEN

**TÍTULO:** “ALGORITMO HÍBRIDO PARA LA RESOLUCIÓN DEL PROBLEMA DE RUTEO DE VEHÍCULOS CON ENTREGA Y RECOGIDA SIMULTÁNEAS (VRPSPD).”\*

**AUTORES:** CRUZ JIMÉNEZ, Camilo  
GONZÁLEZ VILLAMIZAR, Ludy Juliana\*\*

**PALABRAS CLAVE:**

Ruteo de vehículos, algoritmo híbrido, enjambre de partículas, optimización, entrega y recolección, metaheurísticas.

**DESCRIPCIÓN:**

El problema de ruteo de vehículos con entrega y recolección simultánea es una variación del VRP que ha generado gran interés en la comunidad académica y en los sectores productivos debido a que considera la posibilidad de entregar y recoger bienes de forma simultánea. Por lo tanto, es un problema de gran aplicabilidad en una variedad de campos de la logística y la logística inversa. Particularmente, la complejidad de este problema está en la necesidad de servir un conjunto de clientes localizados en un espacio geográfico considerando la carga fluctuante de los vehículos en cada uno de los trayectos de las rutas.

Matemáticamente, el VRPSPD es un problema de optimización combinatoria de tipo NP-hard para el cual encontrar buenas soluciones implica consumo de recurso computacional de tipo exponencial. Por esta razón, es común el uso de métodos aproximados como metaheurísticas y heurísticas, que permitan obtener buenas soluciones en tiempo computacional razonable.

En este proyecto de investigación se propone el uso de un algoritmo aplicado al problema VRPSPD constituido por el método de “optimización de enjambre de partículas” (PSO por sus siglas en inglés), una heurística de construcción de rutas cheapest insertion heuristic y una heurística para mejoramiento de rutas de búsqueda local 2-opt. El método propuesto arroja buenos resultados y fue comparado con otros métodos propuestos para resolver 18 instancias disponibles en la literatura del problema VRPSPD.

---

\* Proyecto de Grado

\*\* Facultad de Ingenierías Fisicomecánicas. Escuela de Estudios industriales y empresariales.  
Director: PhD Henry Lamos Díaz

## ABSTRACT

**TITLE:** "HYBRID ALGORITHM FOR SOLVING THE VEHICLE ROUTING PROBLEM WITH SIMULTANEOUS PICKUP AND DELIVERY (VRPSPD)."

**AUTORS:** CRUZ JIMÉNEZ, Camilo  
GONZÁLEZ VILLAMIZAR, Ludy Juliana\*\*

**KEY WORDS:**

Vehicle routing, Hybrid algorithm, Particle swarm, Optimization, Pickup and Delivery, Metaheuristics.

**DESCRIPTION:**

The vehicle routing problem with simultaneous pickup and delivery (VRPSPD) is a variation of basic vehicle routing problem that has gained relevance in the academic community and in general industry due to the consideration of picking up and delivering goods simultaneously. Thus it is a very applicable problem on the field of logistics and reverse logistics of new environment's organizations. Particularly, the complexity of this problem is based on the need of serving a set of customers situated on a geographical space considering the fluctuating load of vehicles on each route sections.

Mathematically, VRPSPD is a NP-hard complexity combinatorial optimization problem which involves the use of computational resources of exponential type, required to find good solutions. Hence, it is common the use of heuristic and metaheuristic approaches which can be useful for finding good solutions in acceptable computational time.

In this research work, an algorithm applied to solve the VRPSPD using the Particle Swarm Optimization (PSO) approach is proposed, mixed with the construction heuristic known as *Cheapest insertion heuristic*, and a local -search improvement heuristic, *2-opt*. The proposed method is good for solving 18 instances available in the literature, compared with the work of some other authors who have developed different approaches for solving the VRPSPD.

---

\* Degree project

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

## INTRODUCCIÓN

El mundo contemporáneo se caracteriza por una dinámica económica constante representada por la rápida respuesta de las organizaciones a los deseos y necesidades cambiantes de la sociedad. En este sentido, la gestión de la cadena de abastecimiento se convierte en un punto crítico del desempeño empresarial. Por lo anterior, la investigación en los campos de logística y optimización de sistemas productivos adquieren vigencia debido a la cantidad de problemas que deben ser resueltos.

El diseño de redes de distribución como parte del objetivo de creación de valor de las empresas ha sido ampliamente estudiado debido a la sensibilidad de dicho proceso en el diseño total de la función logística organizativa. Para el contexto colombiano, el sector de transporte, relacionado con la movilización de carga, transportaba para el año 2011, 191 millones de toneladas al año, lo cual implica un crecimiento del 91,15% con respecto al año 2001<sup>1</sup>. Por consiguiente, una planeación estratégica eficiente debe existir con el propósito de optimizar las rutas de transporte de carga con el objetivo de maximizar el valor creado por las empresas colombianas.

El problema de ruteo de vehículos (VRP por sus siglas en inglés) es un problema de optimización combinatoria de tipo NP-hard introducido por primera vez por Dantzig y Ramer (1959). Básicamente, el problema consiste en la construcción de rutas desde un depósito hasta un número determinado de clientes ubicados en un espacio geográfico, con una flota de vehículos. Dada su vigencia en la gestión empresarial, han surgido variaciones del problema relacionadas con el tipo de actividad logística llevada a cabo en la asignación de rutas de transporte, y, de

---

<sup>1</sup> MINISTERIO DE TRANSPORTE. Transporte en cifras. Edición 2011. Bogotá: MINISTERIO DE TRANSPORTE, 2012. 43 p.

igual manera, los métodos de resolución del problema han evolucionado para encontrar soluciones adecuadas en tiempo razonable.

Durante las últimas décadas, una creciente preocupación por los temas medioambientales ha impulsado esfuerzos para garantizar su protección tanto por la industria privada como por la legislación. Para enfrentar estas necesidades, se requiere el despliegue de diversas actividades que resultan en un flujo de materiales desde el usuario final regresando a la cadena de suministros.<sup>2</sup>

A partir de lo anterior, una de las variaciones del VRP que encuentra gran aplicación en las operaciones logísticas de las empresas en la actualidad es el *Problema de ruteo de vehículos con entrega y recolección simultáneas*<sup>3</sup> (VRPSPD por sus siglas en inglés). En este problema, 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 tiempo con el objetivo de minimizar costos, distancias, o tiempos de viaje.

La solución óptima de un problema combinatorio ha sido un tema ampliamente estudiado, y se ha encontrado que sólo puede ser hallada para instancias relativamente pequeñas. Los métodos exactos que se utilizan para encontrar una solución óptima para el problema requieren un tiempo de cómputo significativamente grande debido a la complejidad algorítmica de tipo exponencial. Hasta hace una década, ningún algoritmo exacto era capaz de resolver consistentemente instancias con más de 50 clientes<sup>4</sup>, y su tiempo de ejecución se hace inadecuado e irrazonable.

---

<sup>2</sup> DETHLOFF, J. Vehicle Routing and reverse logistics: the vehicle routing problems with simultaneous delivery and pick-up. 2001. Or Spektrum, Vol 23, p 79-96

<sup>3</sup> MIN, H. The multiple vehicle routing problem with simultaneous delivery and pickup points. 1989. Transportation Research A. Vol 23, p. 337-386.

<sup>4</sup> J.-F. Cordeau, M. Gendreau, G. Laporte, J.-Y. Potvin, and F. Semet, "A guide to vehicle routing heuristics," Journal of the Operational Research Society, vol. 53, no. 5, pp. 512–522, May 2002.

Los métodos alternativos de solución de dicho problema, esto es, aquellos que no son exactos, varían de acuerdo a la combinación de técnicas heurísticas y metaheurísticas que, generalmente, consiguen soluciones factibles bastante adecuadas para cada situación. Estos procedimientos se han aplicado al VRP en el marco de los paradigmas de la inteligencia computacional (CI), dentro de los cuales se encuentran las redes neuronales, la computación evolutiva, la inteligencia de partículas, entre otros. Al interior de éste último (Inteligencia de partículas), se encuentran algoritmos de comportamiento poblacional como la colonia de hormigas o el enjambre de partículas.

De acuerdo al planteamiento anterior, en este trabajo de investigación se propone un algoritmo híbrido para la resolución del problema de ruteo de vehículos con entrega y recolección simultáneas (VRPSPD) utilizando un enfoque de optimización de enjambre de partículas (PSO) combinado con la técnica heurística de construcción de rutas *cheapest insertion heuristic* aplicada para decodificar la solución y una heurística 2-opt de búsqueda local que permite reoptimizar soluciones cuya pertinencia radica en la aplicación del algoritmo propuesto al contexto empresarial.

## **1. ANTECEDENTES DEL PROYECTO**

### **1.1 JUSTIFICACIÓN**

Los grandes cambios del entorno económico requieren operaciones más eficientes que permitan el crecimiento de mercados más satisfechos. Debido a la vigencia del VRPSPD en la gestión de las operaciones logísticas empresariales, encontrar soluciones del mismo en tiempo computacional razonable se convierte en una prioridad. De esta manera, el desarrollo del proyecto de investigación encuentra pertinencia en dos aspectos fundamentalmente.

En primer lugar, se ha podido constatar mediante la revisión de literatura que existe una relación directa entre el mundo real y los aportes teóricos desarrollados en materia del problema VRPSPD, es decir, existe pertinencia práctica en la creación de herramientas desde un enfoque académico, pues los algoritmos trabajados en investigación apuntan a encontrar resultados cada vez más cercanos al óptimo sin omitir el hecho de que la complejidad del problema aumenta cuando éste se acerca a simular las circunstancias reales de los sistemas productivos.

En segundo lugar, se encuentra pertinencia teórica en el desarrollo del proyecto debido a la construcción de nuevo conocimiento desde el grupo de investigación de la Escuela de Estudios Industriales y Empresariales: OPALO. Este hecho es de gran relevancia debido a que sustenta los procesos misionales del mismo y, en general, de la Universidad Industrial de Santander. Con el desarrollo del proyecto se pretende hacer un aporte al campo de investigación en optimización de sistemas productivos y logísticos.

## 1.2 PLANTEAMIENTO DEL PROBLEMA

Dentro de todas las etapas de la cadena de suministro, surge la necesidad de considerar el transporte de bienes como un problema de alta relevancia puesto que frecuentemente éste representa un alto porcentaje de los costos logísticos para la mayoría de industrias<sup>5</sup>. De ahí que el Problema de Ruteo de Vehículos constituya uno de los temas más estudiados en el campo de la optimización de los sistemas productivos. Específicamente el VRPSPD, cobra relevancia por el hecho de contemplar simultáneamente las situaciones de entrega y recogida de bienes a clientes específicos, lo que significa, además de consideraciones y restricciones adicionales al problema, un tema de gran interés dado que está directamente ligado a los estudios de logística inversa.

Se ha definido conceptualmente al VRPSPD como un problema de optimización combinatoria del tipo NP-Hard<sup>6</sup>, lo que significa que es un problema computacionalmente difícil de resolver. Para instancias pequeñas (i.e. 10 clientes), el VRPSPD puede ser resuelto mediante métodos exactos que arrojen una solución óptima. Sin embargo, a medida que el sistema crece y las restricciones aumentan, los recursos consumidos y el tiempo computacional serán excesivamente altos en el intento de encontrar la solución óptima. En la literatura existen numerosas investigaciones dedicadas al uso de métodos heurísticos para la solución del VRPSPD, y sin embargo, aunque las heurísticas arrojan soluciones factibles, no dan plena certeza de que la solución hallada, sea realmente la óptima o, por lo menos, cercana a ella.

Es por esto que surge la necesidad de continuar en el ejercicio de la investigación acerca del comportamiento de diferentes y nuevos métodos aproximados como

---

<sup>5</sup> Hebert W. Davis y William H. Drumm. Logistics Costs and Service Database. Annual Conference Proceedings .San Francisco, CA, 2002: Council of Logistics Management.

<sup>6</sup> P. Toth y D. Vigo, "An exact algorithm for the capacitated shortest spanning arborescence," Annals of Operations Research, vol. 61, Dec. 1995, pp. 121-141.

algoritmos híbridos aplicados al VRPSPD, que permitan determinar aún mejores soluciones siempre considerando un consumo razonable de recursos (menor costo computacional), lo que a su vez sustenta la realización de nuevas alternativas para explorar el espacio de búsqueda de resultados cada vez más cercanos a la solución óptima y encaminados a suplir la necesidad de planeación estratégica de los sistemas logísticos.

### **1.3 OBJETIVOS**

#### **1.3.1 Objetivo General**

Desarrollar un algoritmo híbrido para la solución del problema de ruteo de vehículos con entrega y recolección simultánea.

#### **1.3.2 Objetivos específicos**

- Recopilar, seleccionar y organizar la literatura existente concerniente al método metaheurístico Optimización Enjambre de Partículas PSO (Particle Swarm Optimization, por sus siglas en inglés) y su aplicación a programación lineal entera mixta.
- Desarrollar un algoritmo basado en la metaheurística PSO junto con métodos heurísticos como el 2-opt para resolver el problema VRPSPD.
- Implementar en MATLAB los algoritmos basados en la Optimización de Enjambre de Partículas para la solución del problema VRPSPD.
- Realizar experimentos numéricos que permitan medir la eficiencia del algoritmo desarrollado para la solución del problema VRPSPD.
- Entregar un artículo de carácter publicable con los resultados del proyecto de investigación.

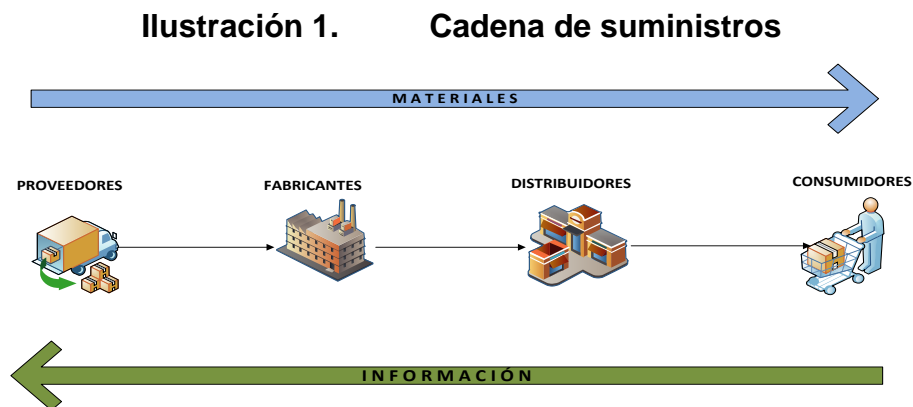
#### **1.4 ALCANCE DEL PROYECTO**

El desarrollo de este trabajo de investigación concluye con una aplicación de Matlab (toolbox) que permita dar una alternativa de solución al problema de ruteo de vehículos con entrega y recolección simultáneas mediante el uso de los algoritmos “Optimización de enjambre de partículas” y “Heurística de búsqueda local 2-opt”.

## 2. GENERALIDADES

### 2.1 SISTEMAS LOGÍSTICOS

Los sistemas logísticos están definidos por todas las actividades que ordenan y administran los flujos de bienes, materiales o información que intervienen en la cadena de suministros de un sector económico dado. Tradicionalmente, la cadena de suministros se define como un sistema de proveedores, fabricantes, distribuidores y clientes donde el flujo de materiales va desde los proveedores hasta los clientes, y el flujo de información fluye en sentido contrario<sup>7</sup>, como se muestra en la ilustración 1.



**Fuente: Autores**

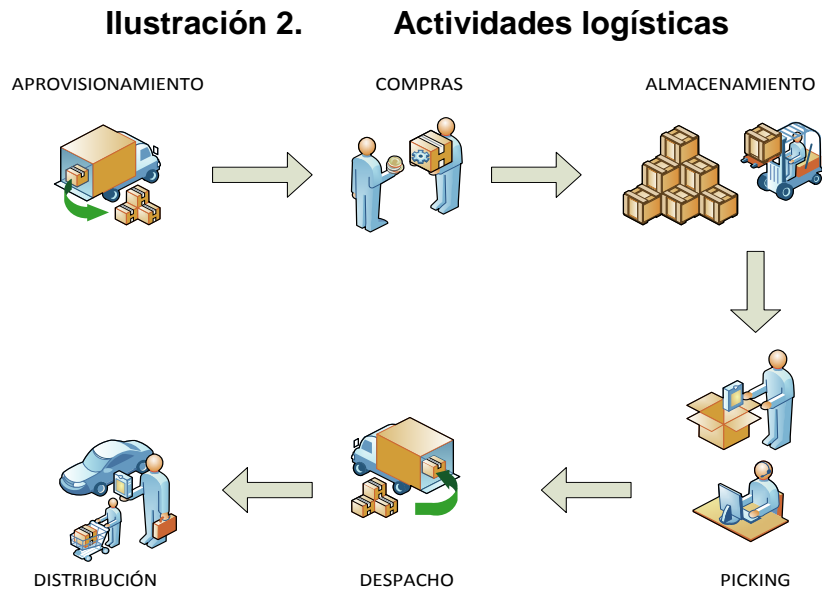
La ejecución, planificación y control eficiente de las actividades involucradas dentro de la logística, permiten aumentar el valor y la competitividad de una organización mediante varios objetivos, dentro de los que se encuentran generar mejores flujos de información a través de la cadena de suministros, implementar modelos de distribución flexibles que permitan la adaptación a mercados

<sup>7</sup> PAU COS, Jordi. y IAVASCUES Y GASCA, Ricardo. Manual de logística integral [online]. Ediciones Díaz de Santos. Madrid, España: Google Books, 2013 [Citado el 30 de Marzo de 2013]. Disponible en: <http://books.google.com.co/Books>.

cambiantes, entre otros beneficios, que finalmente representan un mayor nivel de satisfacción de los clientes.

En la ilustración 2, se observa que las actividades logísticas dentro de una organización siguen una lógica de flujo secuencial que comienza con el aprovisionamiento de materiales e insumos que son obtenidos de proveedores y que son almacenados físicamente dentro de las instalaciones de una fábrica o empresa de distribución. Luego de ser transformados en las áreas de producción son seleccionados y ordenados en pedidos predefinidos (picking), según los requerimientos de los clientes, para finalmente ser despachados mediante las redes de distribución hasta los clientes finales.

Por lo tanto, las organizaciones deben establecer las estrategias que asumirán para cumplir con sus objetivos, identificando los procesos que seguirán para lograr disminución de costos de operación y servicio así como para crear mecanismos de información ágiles y eficaces con proveedores y clientes.



**Fuente: Autores**

## 2.2 LOGÍSTICA DE DISTRIBUCIÓN

La Logística de Distribución es una de las actividades más complejas en términos de la gestión requerida para llevarla a cabo exitosamente. Su objetivo principal es nivelar los flujos de bienes entre los diferentes agentes de la cadena de suministros, teniendo en cuenta la demanda de los clientes y la capacidad de los sistemas de distribución.

Las actividades de distribución se refieren, principalmente, a todas las operaciones que deben realizarse desde que se tienen los bienes listos para ser distribuidos en un depósito hasta la entrega de la mercancía a cada uno de los clientes ( que pueden ser otros centros de distribución, minoristas, consumidores finales, etc.) de manera que se disminuyan los costos y se supla adecuadamente con sus requerimientos. Estos flujos entre proveedor y cliente, pueden identificarse a lo largo de toda la cadena de suministros, por lo que representan una serie de eslabones de suma importancia, como se observa en la Ilustración 3.

Los costos asociados a la distribución en los que debe incurrir la industria en general, son los que representan el mayor porcentaje de los rubros de las actividades logísticas. Tal y como lo describe Garcés Ramírez<sup>8</sup>, según estudios de GS1, el 52,2% de los costos logísticos en Colombia corresponden al transporte de bienes. Si además se tiene en cuenta el almacenamiento, con el 20%, la logística de distribución representa casi el 75% del costo total de la logística.

Lo anterior describe la importancia de la búsqueda continua de metodologías que permitan alcanzar mayor efectividad en las operaciones de distribución y de

---

<sup>8</sup> GARCÉS RAMIREZ, Carlos Duván. Modelo de entregas directas para la reducción de costos logísticos de distribución en una empresa de consumo masivo. Aplicación en una empresa piloto de Caldas. Trabajo de grado para Optar al título de Magíster en Administración de Negocios, Director William Ariel Sarache Castro. Universidad Nacional de Colombia, Facultad de Administración, Manizales. 2010.

estrategias de planeación de largo plazo que otorguen un mejor nivel de servicio a los clientes y que sirvan como punto clave para mejorar la competitividad de las empresas. Es por esto, que este trabajo se enfoca en la formulación y solución de un problema específico dentro de la logística de distribución.

### **2.3 LOGÍSTICA INVERSA**

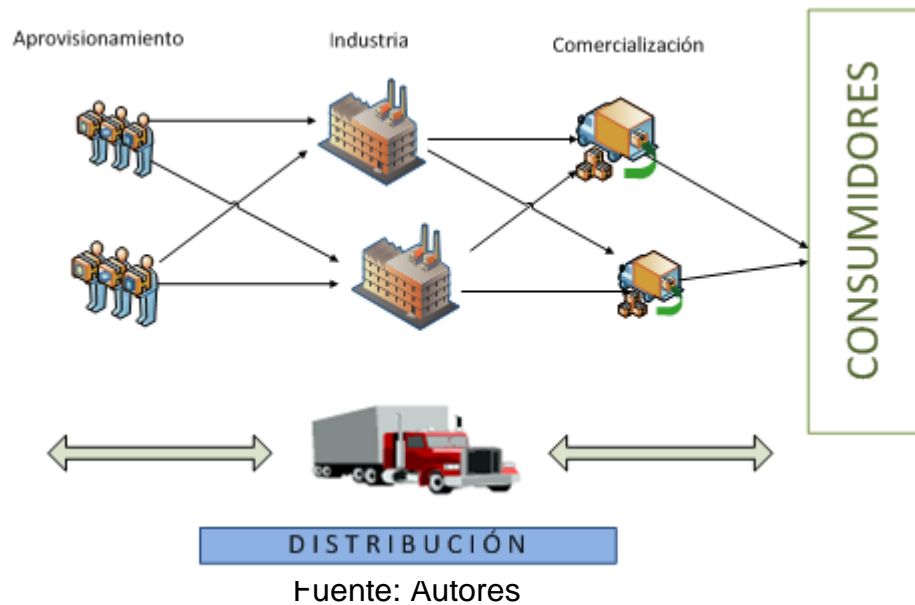
En los últimos años, las necesidades de cumplir con regulaciones ambientales cada vez más estrictas, así como de responder con exigencias sociales y culturales sobre el mejor manejo de desperdicios, han obligado a las empresas a considerar componentes adicionales de la administración de la cadena de suministros, incorporando actividades como el desensamble, reciclaje parcial o completo, re manufactura y reutilización de los bienes económicos usados al final de su ciclo de vida, además de empaques, embalajes y otros dispositivos de carga. Mientras que la reducción de la energía consumida ha sido perseguida por la industria privada puesto que está asociada directamente con los costos, la reducción de desperdicios producidos ha entrado a ser parte del blanco de las organizaciones como resultado de un sentido de responsabilidad ambiental así como consecuencia de enfrentar nuevas regulaciones y leyes al respecto<sup>9</sup>.

Numerosos países han desarrollado normatividad en la industria, obligando a los fabricantes a tomar responsabilidad del ciclo de vida completo de sus productos. Reciclaje de materiales, recuperación de empaques o envases e incluso la obligación de llevar de regreso productos completos al inicio de la cadena de suministros, han sido reglamentados para un gran número de categorías de productos incluyendo equipos electrónicos, automóviles, productos químicos, entre otros.

---

<sup>9</sup> DETHLOFF, J. Vehicle Routing and reverse logistics: the vehicle routing problems with simultaneous delivery and pick-up. 2001. Or Spektrum, Vol 23, p 79-96

### Ilustración 3. Logística de Distribución



Muchos nuevos problemas de planeación y gestión surgen a causa de estas labores de logística inversa y deben ser consideradas cuidadosamente para lograr evitar daños innecesarios al medio ambiente, pero además, así como cualquier otro tipo de decisión administrativa de la organización, deben tratarse como actividades que generan valor agregado en las funciones de las organizaciones. Las actividades de logística inversa están siendo vistas por las compañías como un elemento de generación de nuevas oportunidades (combinando intereses medioambientales con simples beneficios financieros) producidas por reducción en costos de producción y acceso a nuevos segmentos de mercado, así como por compensación de los efectos medioambientales positivos deseados del reciclaje, la re-manufactura y la reutilización de materiales y productos.

Específicamente, diferentes autores han dado diferentes definiciones a la logística inversa (Stock, Kopicki et al., Pohlen y Farris, Rogers y Tibben-Lembke, entre otros), y todos coinciden en que incluyen flujos de bienes usados de regreso al

fabricante original. Fleischmann<sup>10</sup>, sin embargo, intenta dar una interpretación que abarca objetos con características similares que permiten estudiar problemas de decisión logística, y que están en línea con la practicidad actual:

- Primero, se plantea a la Logística Inversa como un elemento de la creciente diversidad de Sistemas Logísticos. Como se mencionó en la sección 2.1, las cadenas de suministro se han percibido tradicionalmente como estructuras unidireccionales con una jerarquía definida. Sin embargo, dadas las condiciones económicas de las últimas décadas, se han desarrollado sistemas logísticos como parte de una red general de las organizaciones que no pueden ser ordenadas a manera de flujo unidireccional; Existe una creciente importancia de los flujos de materiales en doble sentido. Fleischmann resalta el hecho de que estos flujos “inversos” no deben ser separados del panorama global de la toma de decisiones y merecen una atención importante.
- Segundo, La logística inversa concierne a flujos de bienes secundarios en el sentido que se refiere a productos que han completado su vida útil o no han podido ser usados para lo que fueron concebidos originalmente (productos defectuosos). Esto quiere decir que la Logística Inversa debe ocuparse de los derivados de algún uso previo. El objetivo es maximizar el valor económico de aquel uso derivado. Este uso puede incluir disposición o alguna forma de recuperación para generar nuevos productos.
- Tercero, el término de Logística Inversa, expresa en gran medida la perspectiva de la parte receptora. Por lo tanto, el concepto puede concebirse como una forma especial de la logística de recepción.
- A partir de estas características, Fleischmann resume una definición de Logística Inversa.

*“Logística Inversa es el proceso de planeación, implementación y control eficiente y efectivo del flujo de llegada, almacenamiento de bienes secundarios e*

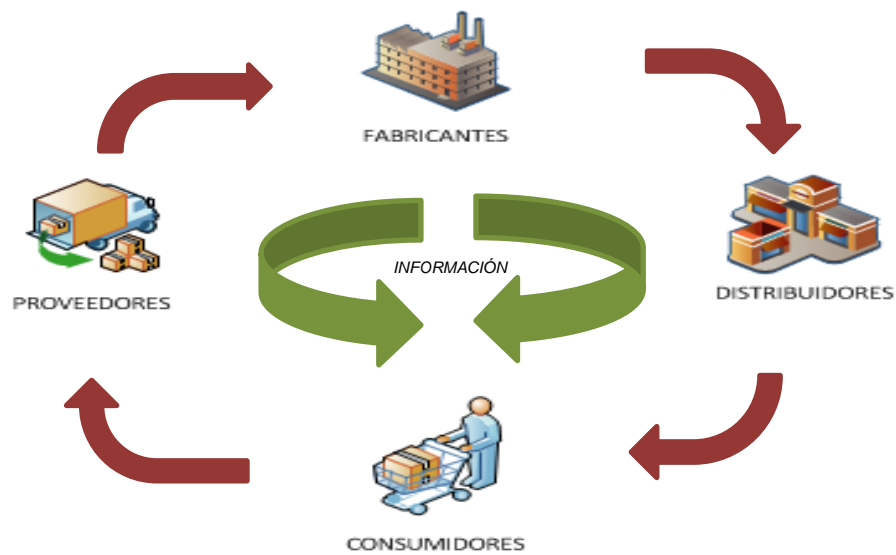
---

<sup>10</sup> FLEISCHMANN, M. Quantitative models for Reverse Logistics. Thesis to obtain the degree of Doctor from the Erasmus University Rotterdam, Rotterdam. July 2000

*información relacionada, opuesta al sentido de la cadena de suministros tradicional con el propósito de recuperación de valor o disposición apropiada.”*

De esta manera, la Logística Inversa representa el último eslabón que falta para completar el ciclo que proponen los nuevos enfoques de Cadena de Suministros. Así, no es un sistema lineal sino un sistema en el que el flujo de bienes no termina al finalizar su vida útil, como se muestra en la ilustración 4.

**Ilustración 4. Esquema Logística Inversa**



### 3. MARCO TEÓRICO

#### 3.1 OPTIMIZACIÓN COMBINATORIA

En un contexto académico o científico, el término optimizar se refiere a encontrar la mejor solución a un problema determinado. Para encontrar dicha solución existen numerosas técnicas y criterios de decisión que responden al objetivo y las variables de decisión de cada problema. En la industria y en la ciencia existen numerosos problemas de optimización tales como los problemas de distribución, planeación de rutas, programación de producción, redes de comunicación, entre otros<sup>11</sup>.

En este sentido, la optimización combinatoria se dedica a la búsqueda de la mejor configuración de un conjunto de elementos presentes en un problema<sup>12</sup>. Los problemas de optimización combinatoria son aquellos que buscan un valor máximo o mínimo dentro de un espacio finito o infinito numerable de soluciones factibles.

Para la resolución de problemas de ruteo de vehículos, propios de la optimización combinatoria, se requiere reformularlos a través de un modelo matemático, esto es, mediante símbolos y expresiones matemáticas que representen de forma ideal la esencia del problema<sup>13</sup>.

Desde el punto de vista de la formulación de los problemas de optimización, los problemas de optimización combinatoria pertenecen al grupo de problemas cuyas soluciones se encuentran cifradas en variables discretas.

---

<sup>11</sup> MARTÍ, R. Procedimientos Metaheurísticos en Optimización combinatoria. Departamento de Estadística e Investigación Operativa. Facultad de Matemáticas. Universidad de Valencia. 2003.

<sup>12</sup> SÁNCHEZ, M. G. Las matemáticas del siglo XX: una mirada en 101 artículos. Págs 115-120. Universidad de la Laguna, Sociedad Canaria Isaac Newton de profesores de matemáticas: Nivola. 2000.

<sup>13</sup> HILLIER. F.S., LIEBERMAN. G.J. Introducción a la investigación de operaciones. Novena Edición. Editorial Mc. Graw-Hill. México, 2010.

Por lo anterior, un problema de optimización combinatoria se caracteriza por tener los siguientes elementos<sup>14</sup>:

- Un conjunto de variables con un dominio establecido;
- Restricciones para cada variable;
- Y una función objetivo  $f$  por minimizar o maximizar.

El conjunto de posibles soluciones de un problema  $S$ , se define como:

$S = \{s = \{(x_1, v_1), \dots, (x_n, v_n)\} \mid v_i \in D_i\}$   $s$  satisface todas las restricciones, donde:

$X = \{x_1, \dots, x_n\}$

$D_i$ : Dominios de la variable

Las técnicas utilizadas para la resolución de problemas de optimización combinatoria han sido objeto de estudio en investigación operacional debido al carácter práctico de los mismos, lo cual ha llevado a la creación de algoritmos que entreguen buenas soluciones con un tiempo computacional razonable.

### 3.1.1 Problemas Combinatorios

Los problemas genéricos de optimización combinatoria son aquellos cuyo objetivo es encontrar el máximo (o el mínimo) de una determinada función objetivo sobre un conjunto finito de soluciones<sup>15</sup>. Debido a esto, las variables han de ser discretas, restringiendo su dominio a una serie finita de valores. Cuando las variables se agrupan en conjuntos que representan objetos, casos o grafos, la técnica de optimización usada es de carácter combinatorio, y se encarga de ubicar dichas variables en ciertas posiciones. El conjunto de posiciones se llama

---

<sup>14</sup> BLUM, C., ROLI, A. Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. ACM Computing Surveys, Vol. 35, No. 3, September 2003, pp. 268–308.

<sup>15</sup> MAREK L. A note on robustness tolerances for combinatorial optimization problems. Information Processing Letters. Vol 110. 2010. p.725-729.

configuración y la mejor configuración, es la optimización combinatoria del problema. Habitualmente, el número de elementos de ese conjunto de soluciones es muy elevado, haciendo impracticable la evaluación de todas sus soluciones para determinar el óptimo.

Existen numerosos tipos de problemas combinatorios de acuerdo a distintos criterios de clasificación. Existen algunos que son relativamente fáciles de resolver, como el caso de los *problemas lineales*, en los que tanto la función objetivo como las restricciones son expresiones lineales<sup>16</sup>. Si las variables de decisión de los problemas lineales pertenecen a los enteros positivos, el problema de optimización se denomina *problema de programación lineal entera*.

Sin embargo, la mayor parte de los problemas de optimización que se pueden encontrar en la práctica, son problemas que pertenecen a otras categorías “más difíciles de resolver”. Por ejemplo, problemas donde las variables de decisión son de naturaleza binaria, es decir, del tipo booleano. Si además a las variables se les incorpora cierto nivel de incertidumbre, se tiene un problema de programación estocástica. Otro caso son los *problemas de programación lineal entera mixta*, en los cuales las variables son de naturaleza discreta, binaria y continua.

### **3.1.2 Complejidad Computacional**

Uno de los principales objetivos de la ciencia de la teoría computacional es comprender la cantidad de recursos (tiempo, memoria, comunicación, etc.) necesarios para resolver problemas computacionales de gran interés. Por esta razón, la teoría de la complejidad computacional está interesada en alcanzar los límites inferiores de aquella cantidad de recursos empleados, ya que existen

---

<sup>16</sup> MARTÍ R. Op cit

problemas cuyo consumo de tiempo, memoria, etc., es bastante elevado, incluso en instancias pequeñas<sup>17</sup>.

Los problemas computacionales de optimización ‘difíciles de resolver’ son aquellos en los que el número de interrelaciones de las variables discretas es muy grande, lo que implica que no se puede garantizar el hallazgo de una solución óptima en un tiempo de cómputo razonable<sup>18</sup>.

Una convención estándar consiste en denominar como ‘factibles’ a aquellos problemas computacionales que pueden ser ejecutados en tiempo polinomial, es decir, si existe un polinomio  $p$ , tal que el algoritmo de solución pueda ejecutarse en un tiempo máximo  $p(n)$ , siendo  $n$  la extensión de entradas del problema. Los problemas factibles que pueden resolverse mediante un método determinístico, se encuentran en la clasificación de problemas tipo  $P$ <sup>19</sup>.

Sin embargo, cuando los métodos exactos no pueden encontrar una solución óptima, se debe recurrir a métodos estocásticos que permitan encontrar buenas soluciones a esos problemas. A este tipo de problemas que se pueden resolver en tiempo polinomial mediante el uso de métodos estocásticos se les agrupa en la clasificación de problemas tipo  $NP$ <sup>20</sup>.

El problema de ruteo de vehículos se considera un problema de optimización combinatoria de tipo  $NP$ -hard ya su solución óptima sólo puede ser obtenida para pequeñas instancias del problema. Esto se debe a que el límite inferior bajo el cual las restricciones pueden ser definidas (conocido en inglés como “*sharp bound*” es difícil de derivar, lo cual significa que los algoritmos determinísticos conocidos

---

<sup>17</sup> L., TREVISAN. “Lecture Notes on Computational Complexity”. Computer Science Division. Berkeley, University of California. 2004.

<sup>18</sup> R. MARTÍ. Op. cit.

<sup>19</sup> L. TREVISAN. Op. cit.

<sup>20</sup> T. WEISE., M. ZAPF, R. CHIONG, A. NEBRO. Nature-Inspired Algorithms for Optimization. Series: Studies in Computational Intelligence, Vol 193. Springer, 2009.

tendrían una baja tasa de convergencia.<sup>21</sup> No existe un método exacto conocido para resolver instancias superiores a 50 clientes para el problema VRP<sup>22</sup>.

### 3.1.3 Optimización Matemática

Para estudiar los problemas que se presentan en el mundo real en el campo de la industria, se requiere construir una representación aproximada de los procesos y actividades involucrados en la toma de decisiones. A estas representaciones se les llama modelos.

El objetivo de la construcción de un modelo es abstraer la esencia de un evento u objeto para analizar su comportamiento a partir de indagaciones sobre el mismo. Los modelos de Investigación de operaciones sirven para mostrar relaciones matemáticas que permiten describir problemas de la vida real, y así facilitar su análisis. Dado a que el conocimiento del problema está basado en una representación, las conclusiones que se derivan del análisis del modelo corresponden a éste mismo y no al problema directamente. Es por esto que se requiere que los modelos objeto de estudio tengan validez, para lo cual se prefieren modelos robustos basados en la matemática para la modelación de problemas<sup>23</sup>.

Los modelos de optimización matemática buscan encontrar las mejores soluciones de un problema entre un espacio de soluciones. Estos modelos están constituidos por tres partes: *Variables de decisión*, que representan un conjunto de decisiones cuantificables las cuales se relacionan entre sí para generar diferentes soluciones del problema; *restricciones*, que limitan los valores que pueden tomar las variables de decisión; y la *función objetivo*, mide el desempeño de las soluciones obtenidas

---

<sup>21</sup> J-F. CORDEAU, M. GENDREAU, G. LAPORTE, J-Y POTVIN, F SEMET. "A guide to vehicle routing heuristics". Journal of the Operational Research Society, 2002. Vol. 53, p. 512-522

<sup>22</sup> P. TOTH, D VIGO. "Exact solutions of the vehicle routing problem". Fleet Management and Logistics. 1998. Kluwer: Boston, p 1-31.

<sup>23</sup> HILLIER, Frederick S. & LIEBERMANN, Gerald J., Introduction to Poerations Research. Ed. 9 Mc-Graw Hill. New York 2010. P. 10-11.

con la combinación de variables evaluadas de acuerdo a uno o varios criterios: Maximizar y/o minimizar.

## **3.2 PROBLEMA DE RUTEO DE VEHÍCULOS**

### **3.2.1 Definición**

El problema de Ruteo de vehículos (VRP, por sus siglas en inglés) es un tema de investigación de gran interés en el campo de la optimización de sistemas productivos y logísticos dadas las aplicaciones prácticas encontradas<sup>24</sup>. VRP es un nombre genérico para una clase de problemas que consiste en determinar un conjunto de rutas de vehículos; la ruta vehicular parte de un depósito, sirve a un conjunto de clientes ubicados en una zona geográfica, visitando una sola vez a cada uno de ellos, y retorna al mismo depósito<sup>25</sup>, de tal forma que se satisfagan los requerimientos de la demanda, las restricciones de la operación y se minimice el costo total de transporte.

La red de vías, usadas para el transporte de bienes, generalmente se representa a través de un grafo, cuyos arcos describen los tramos de las vías y cuyos vértices corresponden a intersecciones de las vías y a la localización de los clientes. Los arcos pueden ser dirigidos o no dirigidos, dependiendo de si pueden ser recorridos en un solo sentido, o bien, en ambos sentidos. Cada arco tiene un costo asociado que generalmente representa su longitud o su tiempo de viaje.

En la ilustración 5 se observa un problema típico de Ruteo de Vehículos. El conjunto de vértices (clientes y depósito) conectados entre sí por aristas o arcos,

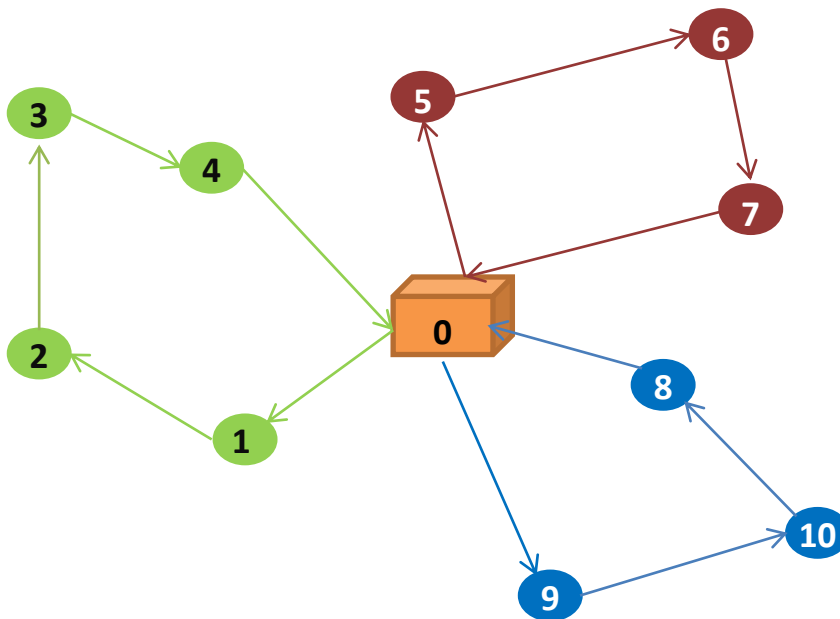
---

<sup>24</sup> SUNGUR, Ilgaz. The Robust Vehicle Routing Problem. Ann Arbor, MI – USA, 2007, 24 h. PhD Thesis (Doctor of Philosophy, Industrial and systems engineering). University of Southern California, Faculty of the Graduate School.

<sup>25</sup> The jin Ai, Voratas Kachitvichyanukul, A particle Swarm Optimization for the vehicle routing problem with simultaneous pickup and delivery. Computers & Operations Research, vol 36, 2009, p. 1693 -1702.

representa un grafo. Dentro de los grafos, los caminos que inician y terminan en un mismo punto (en este caso el depósito), y pasan por cada vértice una sola vez, se conocen como Ciclos Hamiltonianos. Cada línea del grafo es un arco que indica el camino entre dos clientes, quienes están representados con números.

### Ilustración 5. Representación gráfica de un problema de ruteo de vehículos



Fuente: Autores

Las rutas destinadas a atender a los clientes comienzan y terminan en uno o varios depósitos localizados en los vértices del grafo. Cada depósito se caracteriza por un número y tipo de vehículos asociados y por el monto total de bienes que debe manejar. En algunas aplicaciones reales, los clientes se *clusterizan* y se asocian a cada depósito, y los vehículos deben regresar a su depósito respectivo al final de cada ruta. En estos casos, el problema completo puede ser descompuesto en varios sub problemas independientes, cada uno asociado a un

depósito distinto<sup>26</sup>. De ahí que el VRP sea una generalización del Problema del Vendedor Viajero (TSP, por sus siglas en inglés), en el cual se busca encontrar el camino más corto asociado a la visita de un conjunto de nodos y el costo de viaje entre cada uno de ellos, regresando al punto de partida<sup>27</sup>.

Las características típicas de los clientes son<sup>28</sup>:

1. Monto de bienes (demanda), posiblemente de diferentes tipos, que deben ser entregados o recogidos a los clientes;
2. Horarios en el día (ventanas de tiempo) durante los cuales los clientes pueden ser atendidos, sea por los periodos en los que el acceso a la localización de los clientes está permitida (por restricciones de tráfico) o por el horario de atención de los clientes;
3. Tiempos requeridos para entregar o recoger la mercancía a los clientes, debido a las características de los vehículos para carga y descarga.

El transporte de bienes es realizado por una flota de vehículos cuya composición y tamaño pueden ser definidos de acuerdo a los requerimientos de los clientes. Las características típicas de los vehículos son<sup>29</sup>:

1. Capacidad del vehículo expresado en máximo peso, volumen, número de pallets que el vehículo puede cargar, etc.;
2. Posible subdivisión del vehículo en compartimientos con el fin de cargar bienes de diferentes tipos;
3. Dispositivos disponibles en las operaciones de cargue y descargue;
4. Costos asociados a la utilización del vehículo.

---

<sup>26</sup> P. Toth, D. Vigo. THE VECHICLE ROUTING PROBLEM. Vol 9 of Monographs on Discrete Maths and Application Series. SIAM, 2002. ISBN: 978 – 089 – 87 – 1579 - 8

<sup>27</sup> Applegate, D. L.; Bixty, R.M.; Chvátal, V; Cook, W.J. The Travelling Salesman Problem, 2006. ISBN: 0-691-12993-2

<sup>28</sup> P. Toth, D. Vigo. Op cit.

<sup>29</sup> Ibid

Así, por todas las restricciones operacionales que deben satisfacer las rutas, dependiendo de los bienes transportados, del nivel de servicio y de las características de los clientes y de los vehículos, se entiende al VRP como un problema de optimización combinatoria del tipo NP-Hard<sup>30</sup> que involucra los siguientes elementos:

- Un conjunto de variables de decisión (Variables independientes asociadas a determinar el conjunto de rutas con el menor costo de transporte)
- Una función objetivo que mide la efectividad de cada sistema de decisiones, en función de los costos fijos y variables asociados a la operación de distribución.
- Un conjunto de restricciones que representan las limitaciones bien sea de capital, recursos, tiempo, capacidad de los vehículos, etc.

Se trata de un problema de optimización combinatoria, ya que las variables que definen las rutas, los vehículos y los clientes son de naturaleza discreta y se relacionan de diferentes maneras unas a otras<sup>31</sup>. El VRP fue introducido por primera vez por Dantzig y Ramser<sup>32</sup> (1959), y a partir de ese momento, diferentes y más complejas variaciones al problema fueron tratadas a lo largo de los años<sup>33</sup>(ilustración 6). Algunos tienen en cuenta el horario de entrega o recogida en cada lugar de interés (ventanas de tiempo), la naturaleza de la demanda de los clientes (determinística, estocástica) o el tipo de la flota de los vehículos (homogénea, heterogénea); otros consideran la disponibilidad de múltiples depósitos o la aleatoriedad de algunos parámetros del problema, entre otros.

---

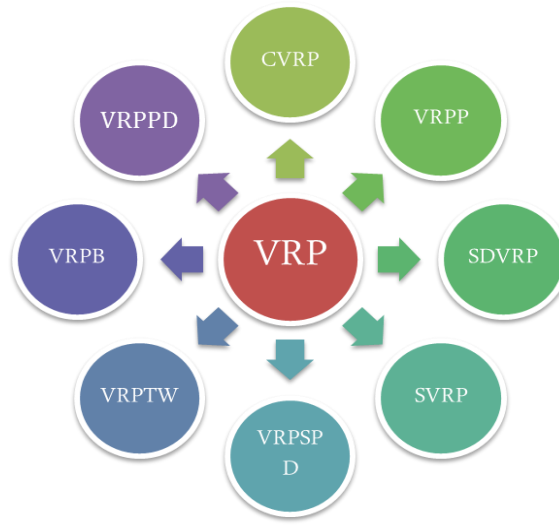
<sup>30</sup> P. Toth and D. Vigo, "An exact algorithm for the capacitated shortest spanning arborescence," *Annals of Operations Research*, vol. 61, Dec. 1995, pp. 121-141.

<sup>31</sup> Garey, M.R., Johnson, D.S., 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York.

<sup>32</sup> G. B. Dantzig, J. H. Ramser. *The Truck Dispatching Problem*. *Management Science*, Vol 6. No 1, Oct. 1959, pp. 80 - 91

<sup>33</sup> F. P. Goksal, I. Karaoglan, F. Altiparmak. *A Hybrid Discrete Particle Swarm Optimization for Vehicle Routing Problem with Simultaneous Pickup and Delivery*. Jan. 2012, *Computers & Industrial Engineering*, 2012.

### Ilustración 6. Variaciones del VRP



Fuente: Autores

### 3.3 PROBLEMA DE RUTEO DE VEHÍCULOS CON ENTREGA Y RECOLECCIÓN SIMULTÁNEAS

Recientemente el VRP con entrega y recolección (VRPPD) ha recibido mayor atención por profesionales e investigadores debido a su alta importancia en aplicaciones prácticas de logística inversa. Dentro de las variantes del VRPPD se encuentra el VRP con retornos (VRPB, VRP with backhauls), el VRP con entrega y recolección mixta, el VRP con entrega y recolección simultánea (VRPSPD) y el problema Dial-a-Ride (DARP).

En 1989, H. Min se interesó por un problema de la vida real de transporte de libros en una biblioteca, y desde ahí, planteó por primera vez la variación del VRP con entrega y recolección simultánea (VRPSPD). En su trabajo, planteó una aproximación a la solución que consistía en lo siguiente: (i) En primer lugar los clientes son clusterizados de forma que la capacidad del vehículo no sea excedida

en cada grupo; (ii) un vehículo es asignado a cada cliente; y (iii) para cada grupo de clientes se resuelve el Problema del Agente Viajero (TSP)<sup>34</sup>.

Las aplicaciones del VRPSPD son encontradas frecuentemente en los sistemas de distribución de las cadenas de abarrotes y alimentos, donde cada tienda de abarrotes usualmente presenta demandas tanto de entrega (comida o bebidas frescas) como de recolección (alimentos vencidos o botellas vacías) y es atendida en una sola visita del proveedor. Con esto, los minoristas pueden negociar la devolución del exceso de productos no vendidos a los productores con efectos beneficiosos para ambas partes. Esto también ocurre en industrias que aplican el uso de pallets o contenedores para el transporte de mercancías, en las cuales se reutilizan estos elementos. Además, como se mencionó en el numeral 2.3, dado que las leyes fuerzan a las empresas a tomar responsabilidad por la vida útil de sus productos, estas se ven obligadas a coleccionar productos usados con el fin de darles un apropiado procesamiento o disposición final. La planeación de rutas de vehículos para tales funciones, también es un VRPSPD<sup>35</sup>.

### 3.3.1 Formulación Matemática

El problema VRPSPD<sup>36</sup> se puede formular de la siguiente manera:

Sea  $G=(V,A)$  un grafo en donde  $V=\{v_0, v_1, \dots, v_n\}$  es un conjunto de nodos, y

$A=\{(v_i, v_j) \mid v_i, v_j \in V, i \neq j\}$  es un conjunto de arcos. Asociada a  $A$  se encuentra una

matriz de distancias  $(d_{ij})$  Y una matriz de tiempo de viaje  $(t_{ij})$ . El nodo  $v_0$  representa un depósito en el cual se encuentra un número de  $m$  vehículos homogéneos, mientras que los demás nodos corresponden a  $n$  clientes por atender. Cada cliente tiene una cantidades no negativas de recogida  $p_i$ , de entrega  $q_i$  y un tiempo

---

<sup>34</sup> F. P. Goksal, I. Karaoglan, F. Altiparmak. Op. Cit.

<sup>35</sup> Dethloff, J. (2001). Vehicle routing and reverse logistics: The vehicle routing problem with simultaneous delivery and pick-up. OR Spektrum, 23, 79–96.

<sup>36</sup> The jin Ai, Voratas Kachitvichyanukul. Op cit. p.1694.

de servicio  $s_i$ . Cada vehículo tiene un costo fijo  $f$ , un costo variable por unidad de distancia  $g$ , capacidad  $Q$ , y duración límite de servicio  $D$ . El problema VRPSPD consiste en el diseño de un grupo de máximo  $m$  rutas de tal manera que:

- (1) Cada ruta empieza y termina en el depósito;
- (2) cada cliente es visitado una vez por un solo vehículo;
- (3) la carga total del vehículo en el trayecto entre dos clientes (considerando la demanda de entrega  $q_i$  y de recogida  $p_i$ ) no excede la capacidad de carga total del vehículo ( $Q$ );
- (4) la duración total de cada ruta (incluyendo el tiempo de viaje y de servicio) no excede el límite previamente establecido  $D$ ; y
- (5) el costo total de ruteo es mínimo.

El VRPSPD también se puede formular como problema de programación lineal entera mixta.

### **Variables de decisión**

- $X_{ijk}$  es una variable binaria que indica si el arco  $(i,j)$  es recorrido o no por el vehículo  $k$   
 $x_{ijk} = 1$  si el vehículo  $k$  recorre el arco  $(i,j)$   
 $x_{ijk} = 0$  si el vehículo  $k$  no recorre el arco  $(i,j)$
- $Y_{ijk}$  carga del vehículo  $k$  mientras recorre el arco  $(i,j)$
- $\delta_{ijk}$  tiempo de inicio de servicio del cliente  $i$  efectuado por el vehículo  $k$

### **Función Objetivo**

$$\text{Minimizar } Z = f \sum_{k=1}^m \sum_{j=1}^n x_{0,jk} + g \sum_{i=0}^n \sum_{j=1}^{n+1} \sum_{k=1}^m d_{ij} * x_{ijk} \quad (1)$$

**Sujeto a:**

$$\sum_{i=0}^n \sum_{k=1}^m x_{ijk} = \mathbf{1}, \quad \text{para } 1 \leq j \leq n \quad (2)$$

$$\sum_{j=0}^n x_{jik} = \sum_{j=1}^{n+1} x_{ijk}, \quad \text{para } 1 \leq i \leq n, 1 \leq k \leq m \quad (3)$$

$$\sum_{j=1}^n x_{0jk} \leq \mathbf{1}, \quad \text{para } 1 \leq k \leq m \quad (4)$$

$$\delta_{ik} + s_i + t_{ij} + \delta_{jk} \leq (1 - x_{ijk})M, \quad \text{para}$$

$$0 \leq i \leq n, 1 \leq j \leq n+1, 1 \leq k \leq m \quad (5)$$

$$\delta_{n+1,k} - \delta_{0k} \leq D, \quad \text{para } 1 \leq k \leq m \quad (6)$$

$$y_{ijk} \leq x_{ijk} * Q, \quad \text{para } 0 \leq i \leq n, 1 \leq j \leq n+1, 1 \leq k \leq m \quad (7)$$

$$\sum_{j=1}^n y_{ojk} = \sum_{j=1}^n q_j * \sum_{i=0}^n x_{ijk}, \quad \text{para } 1 \leq k \leq m \quad (8)$$

$$\sum_{i=0}^n y_{ijk} + (p_j - q_j) * \sum_{i=0}^n x_{ijk} = \sum_{i=1}^{n+1} y_{ijk}, \quad \text{para}$$

$$1 \leq j \leq n, 1 \leq k \leq m \quad (9)$$

$$x_{ijk} \in \{0,1\}, \quad \text{para } 0 \leq i \leq n, 1 \leq j \leq n+1, 1 \leq k \leq m \quad (10)$$

$$y_{ijk} \geq 0, \quad \text{para } 0 \leq i \leq n, 1 \leq j \leq n+1, 1 \leq k \leq m \quad (11)$$

$$\delta_{ijk} \geq 0, \quad \text{para } 0 \leq i \leq n, 1 \leq j \leq n+1, 1 \leq k \leq m \quad (12)$$

En la formulación anterior, la función objetivo (1) muestra la minimización de costos de transporte teniendo en cuenta los costos fijos y variables asociados a los vehículos.

La restricción (2) indica que cada cliente es visitado únicamente por un vehículo; la restricción (3) indica que cada vehículo que llega a un nodo (cliente) debe salir de él y (4), ese vehículo puede servir como máximo a una ruta. Esta restricción debe además garantizar que cada vehículo retorne al depósito luego de haber visitado el último cliente de la ruta, por lo cual debe considerarse un arco desde el nodo  $n$  hasta el nodo  $n+1$ , siendo este último el depósito al final del recorrido. Las restricciones (5) y (6) explican la relación entre las variables de tiempo y los parámetros del modelo. Así, la restricción (5) relaciona el tiempo de inicio de servicio entre un cliente y otro, y la restricción (6) indica la relación entre el primer y último cliente en cada ruta, es decir, establece el límite de duración del servicio. La restricción (7) indica la máxima capacidad ( $Q$ ) del vehículo  $k$  mientras sirve al cliente  $j$  tras haber servido al cliente  $i$ . Por otro lado, la restricción (8) asegura que todas las entregas a clientes partan desde el depósito, y la restricción (9) se encarga de equilibrar la carga del vehículo  $k$  después del servicio a un cliente. Finalmente, las restricciones (10)-(12) indican el dominio de las variables de decisión, lo cual se había descrito anteriormente en la definición formal del VRPSPD.

### **3.4 MÉTODOS DE SOLUCIÓN PARA EL VRPSPD**

En años recientes, varios investigadores han estudiado el problema de ruteo de vehículos con entrega y recolección simultánea debido tanto a su pertinencia práctica, pues modela muchos sistemas reales en los que se resalta la importancia de la logística inversa, como a su pertinencia científica, dada la complejidad del problema de optimización. Las herramientas estudiadas para la solución del

VRPSPD han sido, en su gran mayoría, métodos aproximados con heurísticas y meta heurísticas, debido a la complejidad del problema en instancias de gran escala, para el cual los métodos exactos son ineficientes. Si bien el problema puede ser abordado mediante métodos exactos para instancias pequeñas, se encuentra que los casos reales de aplicación del VRPSPD son siempre problemas de gran tamaño<sup>37</sup>.

### 3.4.1 Métodos Exactos

La utilización de algoritmos exactos busca encontrar la solución óptima de un problema y garantizar la veracidad de este hecho. Sin embargo, los métodos exactos usados para resolver el problema del VRPSPD no han sido muy frecuentes debido a la alta complejidad computacional del problema. Angelelli y Mansini<sup>38</sup> investigaron el VRP con ventanas de tiempo, aplicando el algoritmo *branch-and-price*, en el cual se resuelve primero el problema relajado, y se utilizan métodos de generación de columnas, integrando variables relajadas en cada iteración para encontrar la solución del problema. Allí se aplica el algoritmo *branch-and-bound* y programación dinámica para obtener soluciones enteras. El autor encuentra que la presencia de restricciones de retorno o ventanas de tiempo incrementan la eficacia de este tipo de algoritmos puesto que instancias con más restricciones tienen espacios de estados más pequeños a explorar con la programación dinámica. Por esta razón, algoritmos desarrollados para VRPSPD con restricciones adicionales no pueden resolver instancias puras de VRPSPD de tamaños similares.

---

<sup>37</sup> P. Toth, D. Vigo. SIAM, 2002. Op cit., p.225

<sup>38</sup> Angelelli E., Mansini R., The Vehicle Routing Problem with time Windows and simultaneous pick-up and delivery. Lecture Notes in Economics and Mathematical Systems, A. Klose et al (eds), 249 – 267, Springer, 2002.

El primer trabajo publicado en métodos exactos para el VRPSPD fue presentado por Dell'Amico *et al*<sup>39</sup>, en el cual se estudió cómo puede ser aplicada la técnica de *branch-and-price* para la solución del problema, y en particular, se compararon dos formas diferentes para resolver el subproblema de pricing: programación dinámica exacta y relajación de los espacios de estado. El subproblema de pricing está implícito en el algoritmo *branch-and-price* que consiste en encontrar una ruta elemental con restricciones adicionales a medida que avanza el algoritmo. El autor concluye que el algoritmo planteado es viable para la solución del VRPSPD como un método de optimización para instancias de tamaño pequeño o mediano y propone futuras investigaciones en cuanto a los métodos de solución del subproblema de pricing.

### 3.4.2 Métodos Heurísticos

Los métodos heurísticos son procedimientos inteligentes para encontrar soluciones aproximadas al óptimo de un problema, mediante la ejecución de una tarea que no es producto de un riguroso análisis del problema, sino de conocimiento experto sobre lo que se quiere solucionar. Esto quiere decir que se encuentran buenas soluciones mediante técnicas aplicables y la experiencia acerca de un problema, usando una cantidad de recursos razonables. Son aplicados para la mayoría de problemas de optimización, dada la ineficiencia de los métodos exactos por la complejidad computacional que los caracteriza. El término heurística proviene de la raíz griega *heuriskein*, que significa encontrar o descubrir.

Como estrategia para solucionar el problema VRPSPD, Nagy y Salhi<sup>40</sup> propusieron heurísticas basadas en inserción en las cuales se construyen rutas parciales para

---

<sup>39</sup> Dell'Amico M, Girghini G, Salani M. A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection. *Transport Science* 2006;40(2): 235-47

<sup>40</sup> Salhi S, Nagy G. A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. *Journal of the Operational Research Society* 1999;50(10):1034--42.

un conjunto de clientes, y luego se insertan los clientes faltantes a las rutas existentes. Ambas técnicas probaron encontrar buenas soluciones tanto para problemas multidepósito como para problemas de un solo depósito. Más tarde, Dethloff <sup>41</sup> propuso una heurística de inserción en la que los clientes son insertados en las rutas creadas de acuerdo a tres criterios: distancia de viaje, capacidad residual y distancia desde el depósito. En esta heurística no se incluyó ninguna rutina de mejoramiento.

Dos heurísticas de búsqueda local fueron desarrolladas por Tang y Galvão<sup>42</sup>, ambas desarrolladas inicialmente para el VRP simple. Una es una adaptación del trabajo de Beasley<sup>43</sup>, quien desarrollo una heurística de partición de tours, y la segunda utiliza el algoritmo de barrido propuesta por Gillet y Miller<sup>44</sup>. Estos procedimientos resuelven el problema del TSP con entrega y recolección simultánea para cada ruta individual, adaptando métodos propuestos originalmente para el caso de no simultaneidad.

Posteriormente, Salhi y Nagy <sup>45</sup> propusieron una heurística de búsqueda local con cuatro fases para resolver el VRPSPD. La heurística combina la eficiencia de diferentes rutinas de manera organizada, adaptadas al VRPSPD tales como 2-opt, 3-opt, cambios e intercambios, entre otros. Luego de encontrar una solución inicial en la primera fase, esta es continuamente mejorada en cada una de las siguientes fases manteniendo las condiciones de factibilidad.

---

<sup>41</sup> Dethloff - OR Spektrum 23, Op. Cit. p.79–96

<sup>42</sup> Tang FA, Galvão RD. Vehicle routing problems with simultaneous pick-up and delivery service. Journal of the Operational Research Society of India (OPSERACH) 2002; 39; 19 – 33.

<sup>43</sup> Beasley JE. Route first-cluster second methods for vehicle routing. Omega 1983; 11:403 – 8.

<sup>44</sup> Gillet BE, Miller LR. A heuristic algorithm for the vehicle dispatch problem. Operations Research 1974;22:340 – 9.

<sup>45</sup> Nagy G, Salhi S. Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. European Journal of Operational Research 2005; 162(1):126--41.

Gajpal y Abad<sup>46</sup> presentaron las heurísticas de ahorros y de ahorros paralelos basados en el algoritmo de Clark & Wright para el VRPSPD. Es su trabajo, estos autores desarrollaron una estrategia de red de recolección acumulativa para evaluar factibilidad cuando se unen dos rutas existentes.

Más recientemente, Salhi y Naghy complementaron sus trabajos preliminares con la implementación de heurísticas de inserción para el VRP con retornos, que consiste en insertar en primer lugar los clientes con demandas de entrega, y luego, insertar aquellos con demandas de recogida. En este tipo de heurísticas suele suceder que se insertan clientes con demanda de recogida con un alto costo operacional, dadas las inserciones previas a la ruta.

### **3.4.3 Métodos Metaheurísticos**

Los procedimientos Metaheurísticos son una clase de métodos aproximados de alto nivel que están diseñados para resolver problemas difíciles de optimización combinatoria, que guían y modifican otras heurísticas para explorar soluciones más allá del óptimo local. Son métodos a partir de los cuales se pueden crear nuevos algoritmos híbridos a partir de diversos conceptos de inteligencia artificial. Recientes esfuerzos por dar solución al problema VRPSPD mediante herramientas metaheurísticas han encontrado resultados satisfactorios.

Crispim y Brandao<sup>47</sup> son los primeros autores que presentaron un método metaheurístico para el VRPSPD. Éste consiste en un algoritmo híbrido basado en búsqueda Tabú (TS) y Búsqueda Variable de vecindarios (VND, variable neighborhood Search). Allí, se utiliza un método de barrido para obtener una

---

<sup>46</sup> Gajpal, Y., & Abad, P. (2012). Saving-based algorithms for the vehicle routing problem with simultaneous delivery and pickup. *Computers and Operations Research Society*, 61(1), 1498-1509.

<sup>47</sup> Crispim, J., & Brandao, J. (2005). Metaheuristics applied to mixed and simultaneous extensions of vehicle routing problems with backhauls. *Journal of the operational Research Society*, 56, 1296 – 1302.

solución inicial. Si alguna ruta de la solución inicial no es factible debido a la sobrecarga en sus arcos, se rediseña la ruta intercambiando el orden de los clientes hasta alcanzar factibilidad. En una fase de mejoramiento, se implementan movimientos de intercambio e inserción. Soluciones no factibles son permitidas pero son penalizadas de acuerdo al nivel de sobrecarga que generen en los vehículos.

Tang y Galvao <sup>48</sup> desarrollaron un algoritmo de búsqueda tabú, el cual combina diferentes esfuerzos por obtener soluciones internas de cada ruta y entre las rutas alternativas. Este algoritmo usa tres tipos de movimientos para obtener soluciones adyacentes entre rutas (reubicación de un cliente de una ruta a otra, intercambio de pares de clientes entre rutas y cruce entre dos rutas) y procedimientos 2-opt para obtener soluciones alternativas en cada ruta. En este trabajo se implementaron cuatro tipos de vecindarios, y se probaron dos estrategias de búsqueda diferentes para elegir el siguiente movimiento de la metaheurística.

Wassan et al. <sup>49</sup> diseñaron un método de Búsqueda Tabú Reactivo que consiste en una fase de construcción de una solución inicial, y una segunda fase de mejoramiento. La solución inicial se construye mediante una modificación del método de barrido, que consiste en excluir clientes cercanos al depósito para ejecutar el barrido y cuya finalidad es la de dar flexibilidad para insertar aquellos clientes excluidos en mejores posiciones durante la fase de mejoramiento.

---

<sup>48</sup> Tang FA, Galvao RD. A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Computers and Operations Research* 2006;33(3):595--619.

<sup>49</sup> Wassan, N.A., Wassan, A. H. & Nagy, G. (2007). A reactive tabu search algorithm for the vehicle routing problem with simultaneous pickups and deliveries. *Journal of combinatorial Optimization* 15. 368 – 386.

Chen et al<sup>50</sup> fueron unos de los primeros en aplicar la metaheurística PSO en su versión discreta, combinada con el procedimiento de búsqueda llamado Recocido simulado (SA, Simulated Annealing) para resolver el problema VRP básico. Otros métodos de búsqueda basados en comportamiento poblacional, como algoritmos genéticos y optimización de colonia de hormigas, han sido ampliamente utilizados. Bianchessi y Righini<sup>51</sup> evaluaron y compararon el desempeño de diferentes heurísticas constructivas, métodos de búsqueda local e implementaciones de búsqueda Tabú para VRPSPD. En particular, se enfocaron en la dificultad de aplicar el paradigma de Búsqueda Tabú a algoritmos basados en vecindades variables y complejas. Para este propósito, se combinaron vecindades basadas en intercambio de arcos y en intercambio de nodos, empleando listas tabú diferentes pero relacionadas.

Gajpal y Abad<sup>52</sup> presentaron una metaheurística de optimización de Colonia de Hormigas. En su trabajo, utilizaron una regla de construcción de rutas y dos rutinas de búsqueda local: 2-opt, inserción e intercambio de clientes multi-ruta, e intercambio de sub secuencias multi-ruta.

Ai y Kachitvichyanukul<sup>53</sup> propusieron una estrategia de PSO con múltiples estructuras sociales aplicada al VRPSPD. Las rutinas de decodificación están basadas en la construcción de listas de prioridad de clientes para entrar a las rutas y una matriz de prioridad de vehículos para servir a los mismos, de acuerdo a la posición de las partículas del enjambre generadas en cada iteración.

---

<sup>50</sup> Chen AL, Yang GK, Wu ZM. Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem. *Journal of Zhejiang University Science A* 2006;7(4):607--14.

<sup>51</sup> Bianchessi, N., Righini, G. (2007). Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery. *Computers & Operations Research* 34 (2), 578 – 594.

<sup>52</sup> Gajpal, Y., Abad, P. (2009). An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup. *Computers & Operations Research*.  
Doi:10.1016/j.cor.2009.02.017.

<sup>53</sup> Kachitvichyanukul, V., Jin, A. Op cit.

Zachariadis y Kiranoudis (2011)<sup>54</sup>, propusieron una metodología metaheurística basada en un algoritmo de Búsqueda Local que usa dos conceptos algorítmicos llamados la estrategia Static Move Descriptor (SMD) para explorar eficientemente las soluciones de vecindades, y el *promise concept* para evitar búsquedas cíclicas e inducir diversificación. Estos algoritmos fueron aplicados previamente para modelos de VRP Abierto y VRPB con resultados satisfactorios. La experimentación del algoritmo luego de una comparación con 18 instancias de gran tamaño obtenidas de la literatura arrojó buenos resultados, obteniendo mejoramiento de soluciones obtenidas previamente.

#### 3.4.4 Heurísticas de mejoramiento

Con la implementación de estrategias metaheurísticas para resolver el problema del VRP en todas sus variantes, se han utilizado heurísticas de mejoramiento de la solución basadas generalmente en rutinas de búsqueda local. Algunas heurísticas de mejoramiento que se han combinado con métodos Metaheurísticos son:

- **Heurística 2-opt**

La Heurística de búsqueda local 2-opt fue introducida originalmente por Lin<sup>55</sup> como método para mejorar las soluciones del Problema del Vendedor Viajero (TSP). Es una de las metodologías de búsqueda local más conocidas y se basa en la estrategia de intercambio de arcos<sup>56</sup>. En un problema de VRP, cada ruta puede ser considerada un TSP. El procedimiento consiste en obtener un tour a partir de una secuencia inicial reemplazando 2 de sus arcos por otros 2 arcos diferentes

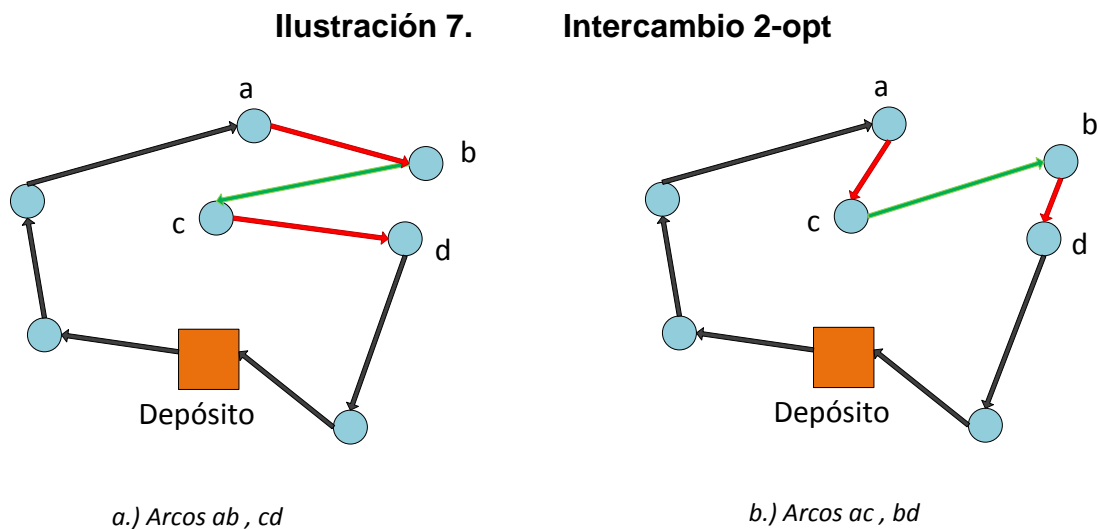
---

<sup>54</sup> E. Zachariadis, C. Kiranoudis. A local search metaheurística algorithm for the vehicle routing problem with simultaneous pick-ups and deliveries. *Expert Systems with Applications* 2011; vol 38 (3): p. 2717-2726.

<sup>55</sup> Lin S. (1965). Computer solutions for the travelling salesman problem. *Bell System Technology Journal*; 44:2245-69.

<sup>56</sup> Gajpal, Abad. Ant colony system for the vehicle routing problem with simultaneous pickup and delivery. *Op. Cit.*

hasta que no haya mejoramiento de la ruta. Tales intercambios son llamados 2-intercambios y un tour que no puede ser mejorado por un 2-intercambio se llama 2-óptimo<sup>57</sup>. En el VRP básico, la nueva secuencia mejorada con 2-opt siempre será factible puesto que la demanda total de una ruta dada no cambia. Sin embargo, esto no es cierto para el VRPSPD. En el VRPSPD, la carga de un vehículo fluctúa en cada uno de los arcos y puede exceder la capacidad en alguno de ellos. Por esto, en el algoritmo propuesto en el presente trabajo, se define una ruta factible como la ruta en la cual la carga a lo largo de cada arco es menor o igual a la capacidad del vehículo. Lo anterior se profundizará en el numeral [4.3.1]. En la ilustración 7 se muestra el intercambio de arcos que ejecuta la heurística 2-opt. El par de aristas (ab, cd) se intercambia con el par (ac, bd), y así también, se invierte la dirección de la secuencia entre los clientes b y c.



**Fuente: Autores**

<sup>57</sup> Bräysy, O., Gendreau, M. Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms. Chemical Engineering PasAmerican Collaboration CEPAC. Tomado de internet el 2 de Julio de 2013  
[\[http://cepac.cheme.cmu.edu/pasi2011/library/cerda/braysy-gendreau-vrp-review.pdf\]](http://cepac.cheme.cmu.edu/pasi2011/library/cerda/braysy-gendreau-vrp-review.pdf)

- **Cheapest insertion Heuristic**

La heurística de Inserción más barata se trata de una heurística de construcción de rutas en la cual los clientes son insertados uno a uno a la ruta hasta que no pueda hacerse más inserciones debido a la no factibilidad por capacidad del vehículo. La heurística puede construir varias rutas paralelamente. El criterio de inserción consiste en el costo (distancia) adicional que implica un cliente adicional en la ruta el cual debe ser minimizado.

Un cliente es seleccionado como primer cliente inicial de la ruta, el cual es llamado 'cliente semilla' y se construye una ruta de ida y regreso desde el depósito como solución inicial. Para los demás clientes aún no insertados, se calcula el valor del criterio de inserción (distancia de viaje) para todas las posibles posiciones en las que puede ser insertado. Luego de evaluar factibilidad en cada una de las posibilidades, se selecciona la que genere el menor costo adicional en la ruta. Esta operación se repite hasta que no pueda hacerse una nueva inserción en la ruta<sup>58</sup>.

Bajo ciertos procedimientos de construcción de rutas, la heurística de inserción más barata puede insertar clientes uno a uno en rutas diferentes, según otros criterios de inserción, como la distancia euclidiana desde el cliente a una zona geográfica donde sirve la ruta o por limitaciones de la capacidad del vehículo.

### **3.4.5 Inteligencia Computacional**

Uno de los principales retos en materia de desarrollo algorítmico, es el desarrollo de modelos que puedan resolver problemas cada vez más complejos. Con este propósito, han surgido disciplinas como la Inteligencia Artificial, la cual logra combinar ciencias como la computación, la biología, la filosofía, la fisiología, entre otras.

---

<sup>58</sup> Dethloff . Vehicle Routing and reverse logistics (2001). Op. Cit.

La creación de la Inteligencia Artificial, denominada AI por sus siglas en inglés, ha permitido conocer la forma en que la inteligencia humana logra desarrollar aspectos como la creatividad, las competencias, el razonamiento, la conciencia y la intuición. Para ello se han creado paradigmas que representan los sistemas biológicos y naturales haciendo uso de la lógica, las máquinas simbólicas de aprendizaje, el razonamiento deductivo, entre otros.

Sin embargo, mucho se ha discutido acerca de la imposibilidad de lograr que la inteligencia de un computador pueda igualar a la inteligencia humana debido al componente intuitivo, emocional y de conciencia que no ha podido ser representado con éxito por los ordenadores. En este sentido, una de las subramas de AI dedicadas al estudio de mecanismos relacionados con la inteligencia adaptativa y asociativa de los sistemas es la Inteligencia Computacional (CI).

La Inteligencia Computacional consta de 5 paradigmas fundamentales de estudio<sup>59</sup>. Éstos son:

- Redes neuronales artificiales (NN)
- Computación evolutiva (EC)
- Inteligencia de enjambres (SI)
- Sistemas inmunes artificiales (AIS)
- Sistemas difusos (FS)

La combinación de paradigmas y técnicas probabilísticas permite la formación de sistemas híbridos ya que ningún paradigma se considera mejor y su rendimiento dependerá del problema y el objetivo (Ver Ilustración 8). Por esta razón, es importante tener en cuenta que todos los paradigmas de CI tienen sus orígenes en sistemas biológicos.

---

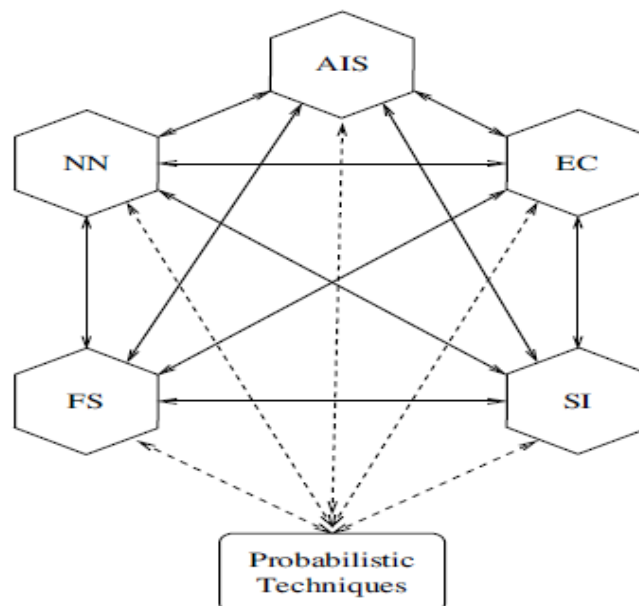
<sup>59</sup> Engelbrecht, A. Computational intelligence An Introduction. Editorial John Wiley. 2º Edición. Chippenham, Great Britain. 2007 - ISBN: 978-0-470-03561-0 (HB)

- **Motivación biológica de los paradigmas CI**
  - **Redes Neuronales Artificiales**

Como su nombre lo indica, este paradigma hace uso del complejo funcionamiento del cerebro humano como principal objeto de estudio. Entre las características de las redes neuronales del cerebro más importantes para abordar este paradigma están: la rapidez de respuesta, la habilidad para aprender, memorizar y generalizar.

El elemento básico de este paradigma son las células nerviosas del cerebro, es decir, las neuronas. A partir de los procesos de sinapsis de las neuronas, se pueden modelar problemas con un único objetivo de tamaño no muy grande que pueden ser resueltos con facilidad. En la actualidad, se han obtenido buenos resultados utilizando un número considerablemente pequeño de redes neuronales artificiales que resuelven una tarea específica.

**Ilustración 8. Paradigmas de inteligencia computacional**



**Fuente:** Engelbrecht, A. Computational intelligence An Introduction. Editorial John Wiley. 2° Edición, 2007.

Sin embargo, se estima que el cerebro humano cuenta con, entre 10 y 500 millones de éstas células. Esta proporción de neuronas y la distribución del cerebro en áreas de trabajo capaces de resolver varias tareas de manera simultánea, dan una dimensión mucho más grande del paradigma y los resultados que podrían obtenerse con la aplicación del mismo.

#### - **Computación Evolutiva**

Este paradigma trata de imitar los procesos de evolución natural que se rigen bajo el principio de supervivencia. Por esta razón, se introducen los conceptos de genética asociados a las características de cada individuo de una población tales como los cromosomas y los genes. Para este paradigma, la función objetivo de un problema es equivalente a la supervivencia del individuo en una población.

A partir del principio de evolución han surgido distintas variaciones del algoritmo que contemplan aspectos como la influencia de la cultura de la población en la supervivencia de sus miembros y los mecanismos de reproducción de los individuos.

El paradigma de la computación evolutiva ha sido aplicado con éxito en problemas del mundo real como la minería de datos, optimización combinatoria, 'clusterización', entre otros.

#### - **Inteligencia de Enjambres**

El origen del paradigma se encuentra en el estudio del comportamiento social de colonias o enjambres de especies naturales como aves, hormigas o peces.

A partir de la simulación del movimiento sincronizado de las bandadas de aves fue diseñado el enfoque de 'Optimización de enjambre de partículas' (PSO). Este algoritmo de búsqueda está basado en el proceso de aprendizaje de los individuos de una población que se deriva de la interacción entre miembros de la misma. El

PSO es aplicado a problemas de optimización, formación de clusters, aproximación de funciones, entre otros.

Por otro lado, los estudios de colonias de hormigas han contribuido al desarrollo de algoritmos inteligentes. Este paradigma es aplicado en optimización de rutas, redes de telecomunicación y problemas de asignación.

#### - **Sistemas Inmunológicos Artificiales**

Este paradigma tiene su origen en el sistema inmunológico natural. Este sistema cuenta con una capacidad excepcional de reconocimiento de patrones, pues de esta manera reconoce células ajenas a un cuerpo y es capaz de crear una estructura de antígeno que permita responder con mayor rapidez la próxima vez que un agente externo ingrese al cuerpo.

Los modelos artificiales de sistemas inmunológicos recogen los aspectos más importantes del sistema natural y son aplicados en tareas de clasificación, problemas de reconocimiento de patrones y detección de anomalías tales como detección de fraude o de virus en computadores.

#### - **Sistemas difusos**

La lógica binaria requiere que el valor de los parámetros sea siempre 0 o 1. De igual manera, en teoría de conjuntos, un elemento pertenece o no pertenece a un conjunto. Esta característica de respuesta exacta no es aplicable al razonamiento humano ya que éste siempre está permeado por algún grado de incertidumbre. Éste es el origen del paradigma de sistemas difusos.

De esta manera, la lógica difusa permite encontrar un tipo de razonamiento aproximado en el cual son válidos los hechos ciertos e inciertos para inferir nuevos razonamientos. La incertidumbre asociada a estos sistemas está asociada a la ambigüedad y la imprecisión. Por lo tanto, no puede asumirse como incertidumbre estadística.

Este paradigma ha sido aplicado con éxito en sistemas de control, sistemas de frenos y transmisión en vehículos, control de tráfico, entre otros.

### **3.5 OPTIMIZACIÓN DE ENJAMBRE DE PARTÍCULAS (PSO)**

El algoritmo de Optimización de Enjambre de Partículas, conocido como PSO por sus siglas en inglés y desarrollado por Kennedy y Heberhart (1995)<sup>60</sup>, fue utilizado inicialmente en el campo de la simulación de un modelo social simple y, posteriormente, se implementó como método de búsqueda en optimización.

En este sentido, los primeros precursores del PSO fueron aquellos científicos interesados en simular el comportamiento de los organismos en una bandada de aves o en un cardumen de peces. Reynolds<sup>61</sup>, Heppner y Grenander<sup>62</sup> analizaron aspectos como la estética de las coreografías de las bandadas de aves, o las reglas que regían el movimiento sincronizado de los miembros de esa bandada, teniendo en cuenta los cambios repentinos de dirección, la desviación y el reagrupamiento<sup>63</sup>.

A través de una serie de planteamientos acerca de la importancia evolutiva que tienen los comportamientos sociales en los procesos cognitivos de los miembros de una especie, se pudo pasar de hablar de una bandada de aves o peces, a un enjambre cuya capacidad para almacenar 'recuerdos' acerca de su propia experiencia y la experiencia de su entorno, podía converger hacia un mismo objetivo.

---

<sup>60</sup> Kennedy J, Eberhart R. Particle swarm optimization. Op Cit.

<sup>61</sup> C.W. Reynolds. "Flocks, herds and schools: a distributed behavioral model". Computer Graphics. 1987, 21(4) P. 25-34

<sup>62</sup> F. Heppner, U. Grenander. "A stochastic nonlinear model for coordinated bird flocks. Editorial In S. Krasner. "The Ubiquity of Chaos". 1990. Publicaciones AAAS, Washington D.C.

<sup>63</sup> J. Kennedy, R. Eberhart. Particle Swarm Optimization. Op. Cit.

El algoritmo original está constituido por un enjambre de **L** partículas, con masa y volumen despreciables, que se mueve sobre un espacio de **H** dimensiones. Para realizar este movimiento, el enjambre cuenta con términos asociados a la 'memoria' que tiene cada partícula acerca de su experiencia y las de sus vecinos, de manera que la partícula actualiza su velocidad con respecto al comportamiento del enjambre. En este ajuste se agrega una variable aleatoria que caracteriza una distribución más real del comportamiento de la población.

El modelo PSO combina dos tipos de aprendizaje para cada partícula del enjambre y un objetivo fijo que es común a cada una de ellas. El primer tipo de aprendizaje de la partícula está asociado a la experiencia personal que ésta desarrolla en la medida en que se desplaza por el espacio de búsqueda. Este comportamiento trata a los miembros del enjambre como seres aislados, y recibe el nombre de "*comportamiento cognitivo*".

El segundo tipo de aprendizaje corresponde al término que se relaciona con el aprendizaje que obtiene la partícula de su interacción con el enjambre. Éste es denominado "*comportamiento social*", y se relaciona con la experiencia del vecindario de cada partícula.

De esta manera, el movimiento del enjambre dentro del espacio multi-dimensional está definido por éstos dos elementos de aprendizaje. La posición de cada partícula se puede denotar como:  $\Theta_{l(\tau)} = [\theta_{l1}, \theta_{l2}, \dots, \theta_{lH}]$  y puede ser codificada como la 'solución' de distintos problemas de optimización.

$\Theta_{l(\tau)}$  = Vector de posición de la partícula *l* en la iteración  $\tau$ .

$\theta_{lH}$  = Posición de la partícula *l* en la dimensión *H*.

La actualización del vector de posición está asociada a la velocidad, la cual se compone de los términos de aprendizaje de la partícula y el peso inercial de ésta. La velocidad para cada agente  $l$  del enjambre en la iteración  $\tau$  es un vector denotado como:

$$\Omega_l(\tau) = [\omega_{l1}, \omega_{l2}, \dots, \omega_{lH}]$$

$\omega_{lH}$  = Componente de la velocidad de la partícula  $l$  en la dimensión  $H$

De esta manera, la posición de la partícula  $l$ , cambiará en cada iteración  $\tau$  de acuerdo a la siguiente formulación matemática:

$$\Theta_l(\tau + 1) = \Theta_l(\tau) + \Omega_l(\tau + 1)$$

La codificación del algoritmo reemplaza el 'objetivo' del enjambre por una función definida como *fitness*, también conocida como función objetivo en los problemas tradicionales de optimización. A partir del cambio en el desempeño de la función objetivo, el método para codificar dicha solución, permite que la velocidad dirija el movimiento de todo el enjambre. En este direccionamiento se refleja el conocimiento de todas las partículas del enjambre y el efecto de la inercia de cada una de ellas.

El conocimiento del enjambre está integrado, en primer lugar, por el aprendizaje cognitivo o personal, también conocido como '*personal best*', el cual corresponde a la mejor posición obtenida por la partícula durante su experiencia de vuelo en el espacio de búsqueda. En segundo lugar, se encuentra el aprendizaje social, denominado '*global best*', el cual es proporcional a la mejor posición obtenida por todo el vecindario en un momento determinado.

Para el PSO propuesto inicialmente, también denominado *Global Best PSO*, el enjambre es considerado por sí solo un vecindario, así que sólo existirán los

componentes global y personal de aprendizaje. De esta manera, el componente ‘*personal best*’ para la  $l$ -ésima partícula, denotado como  $\Psi_l$  para este algoritmo, sigue siendo la posición con mejor desempeño en la función *fitness* obtenida por cada partícula de manera individual; y el ‘*global best*’, denotado como  $\Psi_g$  es la posición de la partícula con mejor desempeño entre todo el enjambre.

Vector de mejor posición personal de la partícula  $l$ :  $\Psi_l = [\psi_{l1} \ \psi_{l2} \dots \psi_{lH}]$

Vector de mejor posición global del enjambre:  $\Psi_g = [\psi_{g1} \ \psi_{g2} \dots \psi_{gH}]$

Para llevar a cabo la actualización del vector de velocidad en cada iteración del PSO, se deben calcular los componentes de aprendizaje de acuerdo al criterio de optimización determinado para la función *fitness* ( $Z$ ). Para problemas de minimización, cada partícula actualizará su ‘*personal best*’ ( $\Psi_l$ ) bajo la condición matemática:

$$\Psi_l(\tau+1) = \begin{cases} \Psi_l(\tau) & \text{si } Z(\Theta_l(\tau+1)) \geq Z(\Psi_l(\tau)) \\ \Theta_l(\tau+1) & \text{si } Z(\Theta_l(\tau+1)) < Z(\Psi_l(\tau)) \end{cases}$$

$$Z: \square^n \rightarrow \square$$

Por otro lado, la actualización del elemento ‘*global best*’ ( $\Psi_g$ ) se realiza siempre que exista una partícula que obtenga mejor desempeño entre toda la población. Esto, para un problema de minimización, es equivalente a:

$$\Psi_g(\tau) \in \{\Psi_0(\tau), \dots, \Psi_L(\tau)\} \mid Z(\Psi_g(\tau)) = \min Z(\Psi_0(\tau)), \dots, Z(\Psi_L(\tau))$$

Siendo  $L$  el número total de partículas que conforman el enjambre.

Una vez actualizados los elementos de aprendizaje del enjambre, el nuevo vector de velocidad en la siguiente iteración  $\tau$  del algoritmo se calcula como se presenta a continuación:

$$\omega_{lh}(\tau+1) = w(\tau)\omega_{lh}(\tau) + c_p u_p (\psi_{lh} - \theta_{lh}(\tau)) + c_g u_g (\psi_{gh} - \theta_{lh}(\tau))$$

En donde,  $\omega_{lh}(\tau)$  es la velocidad de la partícula  $l$  en la iteración  $\tau$ ;  $w(\tau)$  es el peso inercial de la iteración;  $c_p, c_g$  son constantes de aceleración que ponderan cada comportamiento; y  $u_p, u_g \in U(0,1)$  son números aleatorios distribuidos de manera uniforme en el rango entre 0 y 1. En la Ilustración 7., se muestra el comportamiento del paradigma PSO motivado únicamente por la experiencia personal y el resultado global del enjambre en un espacio de dos dimensiones.

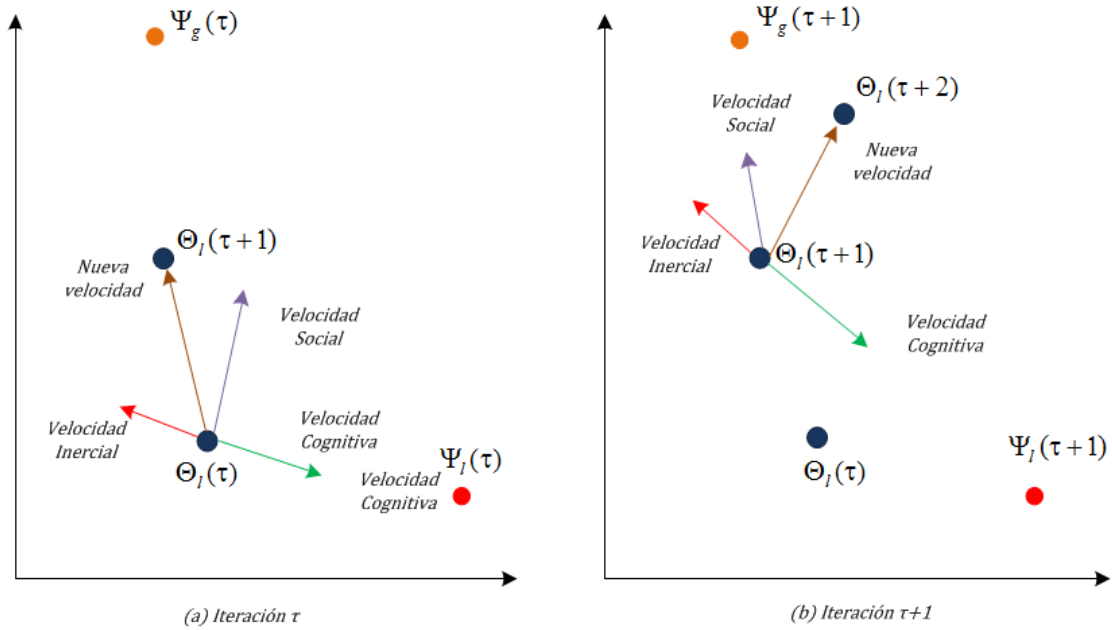
### 3.5.1 Local Best PSO

Una variación del paradigma Global Best PSO es el caso particular del denominado Local Best PSO, en el cual se forman vecindarios más pequeños para evaluar el desempeño de las partículas en la función *fitness*. Esto lleva a una topología de anillo de la estructura social de cada partícula del enjambre.

En este algoritmo, el conocimiento de las partículas refleja la interacción de éstas con su vecindario. Este conocimiento influirá en el cálculo de la velocidad para la siguiente iteración del algoritmo, la cual se define como:

$$\omega_{lh}(\tau+1) = w(\tau)\omega_{lh}(\tau) + c_p u_p (\psi_{lh} - \theta_{lh}(\tau)) + c_l u_l (\psi_{lh}^L - \theta_{lh}(\tau))$$

**Ilustración 9. Actualización de la velocidad y posición de una partícula en un espacio bidimensional en el paradigma PSO**



**Fuente:** Autores

En donde,  $\omega_{lh}(\tau)$  es la velocidad de la partícula  $l$  en la dimensión  $h$  e iteración  $\tau$  ;  $w(\tau)$  es el peso inercial de la iteración;  $c_p, c_l$  son constantes de aceleración que ponderan cada comportamiento; y  $u_p, u_l \in U(0,1)$  son números aleatorios distribuidos de manera uniforme en el rango entre 0 y 1.

La mejor posición local, es decir, el componente 'local best' de cada partícula ( $\Psi_l^L$ ) que reemplaza al elemento global, será aquella posición con mejor desempeño en la función *fitness* encontrada en el vecindario, definida como:

$$\Psi_l^L(\tau+1) \in \{K_l \mid Z(\Psi_l^L(\tau+1)) = \min\{Z(\Theta)\}, \quad \forall \Theta \in K_l\},$$

Con un vecindario denotado matemáticamente como:

$$K = \{k_{l-k}(t), k_{l-k+1}(t), \dots, k_{l-1}(t), k_l(t), k_{l+1}(t), \dots, k_{l+k}(t)\}$$

para vecindarios de tamaño  $k$ . La posición '*local best*' es también referida como la mejor posición del vecindario (*neighborhood best position*).

Es importante tener en cuenta que la formación de los vecindarios en este algoritmo se hace a partir de los índices de las partículas y no de la similitud espacial de las mismas. Este tipo de construcción de vecindarios es conveniente ya que es menos costoso computacionalmente y, dado que de esta manera se distribuye mejor la información a todo el enjambre, el resultado es una solución con mayor convergencia de las partículas en una misma solución de tipo global. Es por esto que el Global Best PSO es considerado un caso especial del Local Best PSO en el cual el vecindario es igual a todo el enjambre.

Las diferencias entre los dos tipos de PSO desarrollados inicialmente, se deben principalmente a la alta interconectividad que existe en las partículas del Gbest PSO, éste puede converger con mayor rapidez a una solución. Sin embargo, esto implica menos diversidad que la obtenida por el Lbest PSO. Por lo tanto, el alto grado de diversificación o de cubrimiento del espacio de búsqueda hará que el Lbest PSO sea menos susceptible a quedarse atrapado en óptimos locales de la función *fitness*.

### **3.5.2 Aspectos relacionados con la implementación del algoritmo PSO**

#### **Inicialización y criterios de parada**

El algoritmo de Optimización de Enjambre de Partículas es iterativo. Por lo tanto, se ejecutará hasta alcanzar un criterio de parada previamente establecido. Esto implica el cálculo de mejores posiciones personales y sociales, y ajustes de velocidad y posición del enjambre durante cada iteración. Para poder llevar a cabo este proceso, es necesario hacer una evaluación de la función *fitness* durante  $\tau$  iteraciones para las  $L$  partículas que conforman el enjambre.

El primer paso en la ejecución del PSO debe ser la inicialización del enjambre y de los demás parámetros que definen el paradigma. El proceso de inicialización del enjambre define en gran medida la efectividad del mismo en la búsqueda de soluciones. Cuando el enjambre es inicializado con poca dispersión de las partículas sobre el espacio de búsqueda, es más difícil que una de ellas encuentre la mejor solución del problema. Por esta razón, se recomienda hacer una inicialización del enjambre cuya diversidad permita cubrir la mayor proporción del espacio.

Para un dominio de la región de búsqueda comprendido entre  $\theta^{\min}$  y  $\theta^{\max}$ , siendo estos los rangos máximos para cada dimensión, un método eficiente de inicialización de la posición de las partículas es:

$$\theta(0) = \theta_j^{\min} + u_j (\theta_j^{\max} - \theta_j^{\min}), \quad \forall j = 1, \dots, H, \forall i = 1, \dots, L$$

Siendo  $u_j$  un número aleatorio entre 1 y 0.

Por otro lado, aunque es posible inicializar la velocidad de cada partícula de manera aleatoria, esto puede orientar al enjambre hacia los límites del espacio de búsqueda, lo cual a su vez hace que se requieran más iteraciones para obtener una buena solución del problema. Es por esto que la inicialización del vector de velocidad es:

$$\omega_{lh}(0) = 0$$

De igual forma, se debe inicializar la mejor posición de cada partícula del enjambre, la cual corresponde a la posición inicial, es decir:

$$\Psi_l(0) = \Theta_l(0)$$

Otro aspecto a considerar al iniciar la ejecución del PSO, es el criterio bajo el que se va a detener el algoritmo. Este elemento es importante debido a que se debe encontrar un balance entre consumo de recurso computacional y rapidez de convergencia del enjambre.

Otro aspecto a considerar al iniciar la ejecución del PSO, es el criterio bajo el que se va a detener el algoritmo. Este elemento es importante debido a que se debe encontrar un balance entre consumo de recurso computacional y rapidez de convergencia del enjambre.

Los tipos de criterio de parada utilizados en la implementación del PSO en la literatura disponible, varían de acuerdo al objetivo con el cual se ejecute el algoritmo y al tipo de problema tratado, y pueden agruparse de la siguiente manera.

- Terminar el algoritmo cuando se alcance un número máximo de evaluaciones de la función objetivo, siendo muy útil cuando se quiere encontrar una buena solución con un límite de tiempo determinado.
- Terminar el algoritmo cuando una solución aceptable haya sido encontrada, lo cual implica encontrar una desviación o error que permita hablar de una solución 'aceptable'. La dimensión de ese error puede tener un gran impacto en el tiempo computacional necesario para ejecutar el algoritmo.
- Terminar el algoritmo cuando, después de una serie de iteraciones, no se observa mejora en la solución. Esto se puede determinar midiendo los cambios en la posición de las partículas y la cercanía a cero de la velocidad en las iteraciones subsiguientes ya que siendo éste el vector que direcciona al enjambre, es posible que ya se haya alcanzado la convergencia. Esto implica la adición de más parámetros para el modelo.

## **Definición de fronteras para el enjambre**

Para medir la precisión y la eficiencia de un algoritmo se debe encontrar un balance entre dos rasgos característicos del mismo: la exploración y la explotación. En primer lugar, la exploración se define como la capacidad del algoritmo para recorrer la mayor proporción del espacio de búsqueda con el objetivo de encontrar buenas soluciones óptimas. En segundo lugar, la explotación es, como su nombre lo indica, la habilidad del algoritmo para concentrarse en un área específica con el fin de refinar a un candidato para la solución de un problema.

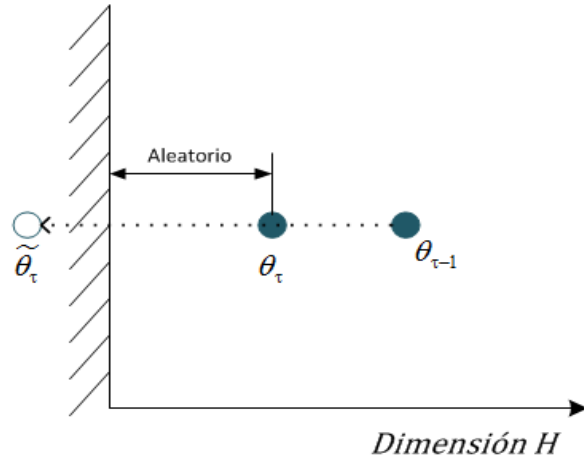
Para encontrar ese balance en la ejecución del PSO, se deben tener en cuenta las condiciones de frontera asociadas a la velocidad y a la posición. Distintos esquemas de administración de fronteras han sido propuestos en la literatura a lo largo de la aplicación del paradigma en distintos problemas de optimización. Los más populares son<sup>64</sup>:

- 1. Esquema de frontera aleatoria:** Cuando una partícula sobrepasa alguno de los límites establecidos dentro de los parámetros del algoritmo, un número aleatorio entre el valor mínimo y máximo de ese parámetro es asignado a la partícula.
- 2. Esquema de frontera absorbente:** Una partícula que se sale de los límites de uno de los parámetros es re-ubicada en la frontera del parámetro para la dimensión en cuestión. De esta manera, la frontera ‘absorbe’ dicha partícula.
- 3. Esquema de frontera reflectora:** Cuando una partícula sobrepasa uno de los límites de algún parámetro, la frontera actúa como un espejo y refleja el desplazamiento de la partícula.

---

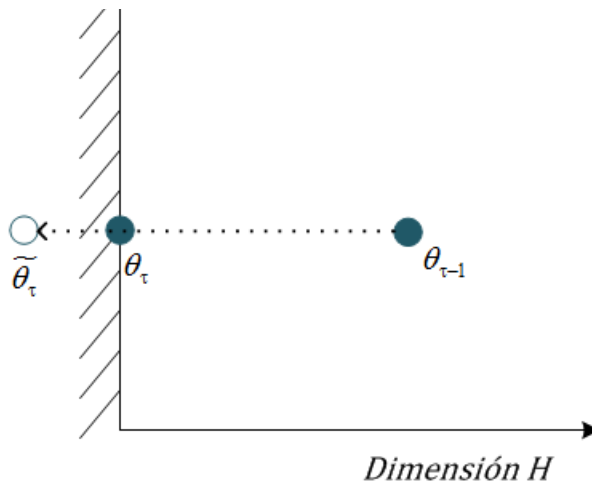
<sup>64</sup> W. Chu, X. Gao, S. Sorooshian. ‘Handling boundary constraints for particle swarm optimization in high-dimensional search space. Information Sciences. 2011, Vol. 181, p. 4569-4581

**Ilustración 10. Representación gráfica de una frontera aleatoria**



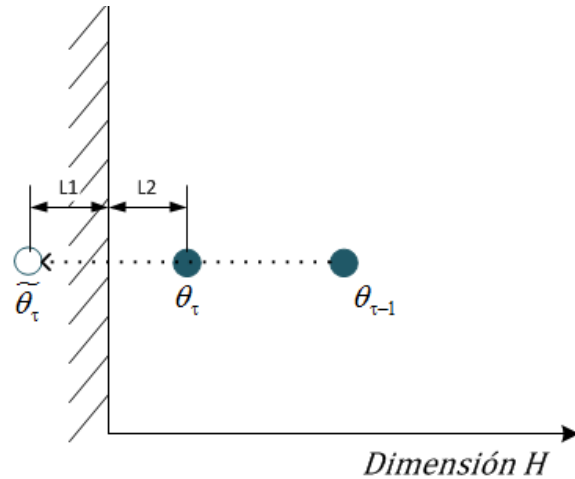
**Fuente:** Autores

**Ilustración 11. Representación gráfica de una frontera absorbente**



**Fuente:** Autores

**Ilustración 12. Representación gráfica de una frontera reflectora**



**Fuente:** Autores

### 3.5.3 Peso inercial y constantes de aceleración

El peso inercial fue introducido con dos objetivos fundamentales: el primero, controlar o balancear la relación entre exploración y explotación del algoritmo; el segundo, para eliminar, al menos parcialmente, la necesidad de restringir la velocidad de las partículas.

El peso inercial, denotado como  $w$ , controla la influencia que la dirección del vuelo anterior de cada partícula tiene sobre el nuevo vector de velocidad. El uso adecuado del peso inercial es muy importante en la convergencia del algoritmo.

Cuando  $w \geq 1$ , la velocidad tiende a incrementarse hasta los límites establecidos (si se ha definido una velocidad máxima), lo cual implica divergencia en el algoritmo. Por otro lado, cuando  $w < 1$ , las partículas desaceleran el movimiento

hasta que la velocidad es cercana a cero (esto depende de las constantes de aceleración previamente definidas).

De esta manera, altos valores de peso inercial harán que el enjambre tienda a moverse en la dirección previa, lo cual es recomendable al inicio del algoritmo<sup>65</sup>; mientras que valores muy pequeños de peso inercial, permitirán que la partícula siga con mayor facilidad los comportamientos cognitivos y social de su memoria. En las primeras aplicaciones del PSO, se utilizó un enfoque de peso inercial estático a lo largo de la ejecución del algoritmo. Sin embargo, en aplicaciones más recientes se han propuestos distintos enfoques de peso inercial dinámico. Por esta razón, la definición de este parámetro depende del problema.

Exista una relación importante entre el peso inercial y las constantes de aceleración definidas para la velocidad de las partículas. En el cálculo del peso inercial deben contemplarse las constantes de aceleración tal como lo proponen Van den Bergh y Engelbrecht<sup>66</sup>, pues de esta manera se garantiza la convergencia del enjambre.

La condición que debe satisfacerse es:

$w > \frac{1}{2}(c_1 + c_2) - 1$ , donde  $c_1$  y  $c_2$  son las constantes de aceleración del comportamiento cognitivo y social, respectivamente.

Cuando el peso inercial es dinámico, se han propuesto distintos enfoques de actualización o cambio del mismo. Las categorías de dichos enfoques son:

---

<sup>65</sup> The jin Ai, Voratas Kachitvichyanukul. Op cit., p1965

<sup>66</sup> F. van den Bergh and A.P. Engelbrecht. A Study of Particle Swarm Optimization Particle Trajectories. Information Sciences, 176(8):937–971, 2006.

- **Ajuste aleatorio**, en donde un peso inercial aleatorio es generado para cada iteración del algoritmo. Debe garantizarse que este peso sea menor que uno.
- **Decrecimiento lineal**, en donde un valor grande de peso inercial se reduce linealmente a un valor pequeño. Es frecuente utilizar 0.9 al principio y 0.4 al final.
- **Decrecimiento no-lineal**, en donde un valor grande de inercia decrece de manera no lineal hasta un valor pequeño de dicho peso. Este método da prioridad a la explotación, destinando menos tiempo a la exploración.
- **Inercia difusa adaptativa**, en donde el peso inercial es ajustado de acuerdo a los fundamentos de los conjuntos difusos y demás características asociadas al paradigma de los sistemas difusos.

Por otro lado, las constantes de aceleración y los números aleatorios involucrados en cada comportamiento del enjambre (cognitivo y social), son los encargados de dar aleatoriedad al movimiento del enjambre. En principio las constantes de aceleración definen el grado de confianza que tienen las partículas en cada comportamiento involucrado en el desplazamiento.

Es importante tener en cuenta que, debido a la relación entre estas constantes y la actualización del movimiento del enjambre, se pueden manejar distintos valores de cada una de acuerdo a la importancia que se le quiere dar a cada tipo de comportamiento. En la Tabla 1. se presenta una explicación de los distintos escenarios que se pueden presentar al cambiar los valores de las constantes de aceleración en la velocidad.

Sin embargo, el enjambre depende de su naturaleza de aprendizaje. Por lo tanto, el desplazamiento de las partículas será más efectivo si:  $c_1 \approx c_2$ , es decir, si se encuentra un balance de coexistencia entre cada una.

**Tabla 1. Escenarios posibles en cada combinación de constantes de aceleración**

Constante de aceleración cognitiva ( $c_1$ )	Constante de aceleración social ( $c_2$ )	Significado
=0	=0	Si no hay peso inercial, las partículas continuarán moviéndose con su propia rapidez hasta encontrar una frontera.
>0	=0	Las partículas se comportan como individuos independientes.
=0	>0	El enjambre será atraído hacia un único punto en el espacio ( $g_{best}$ )

**Fuente:** Autores

### 3.5.4 Otros parámetros del algoritmo

- **Tamaño del enjambre**

Este parámetro corresponde al número de partículas que conforman el enjambre del algoritmo. Mientras mayor sea el número de partículas, mayor es la diversidad, es decir, mayor área del espacio de búsqueda es cubierta. Sin embargo, el número de partículas puede aumentar la complejidad computacional del algoritmo, y esto podría degenerar las soluciones eventualmente.

Desde otro punto de vista, el número de partículas puede reducir el número de iteraciones necesarias para encontrar una solución global del problema.

Estudios empíricos desarrollados con el PSO<sup>67</sup>, han demostrado que el algoritmo tiene la habilidad de encontrar soluciones globales con un tamaño de enjambre comprendido entre 10 y 30 partículas. No obstante, el valor óptimo del tamaño de enjambre depende del problema ya que un espacio de búsqueda regular requiere menos partículas que uno irregular. Por lo tanto, en caso de querer optimizar este parámetro, es necesario recurrir a métodos de validación cruzada.

---

<sup>67</sup> Engelbrecht, A. Op. cit. p.312

- **Tamaño del vecindario**

La definición del tamaño de los vecindarios de partículas está asociada al grado de interacción entre las partículas del enjambre. Mientras que los vecindarios pequeños corresponden a un menor grado de interacción y convergen con menos rapidez, la cohesión entre ellos permite encontrar soluciones óptimas confiables y los hace menos vulnerables al estancamiento en óptimos locales.

Este parámetro puede ser ajustado de manera creciente a lo largo de la ejecución del algoritmo para medir el desempeño del mismo.

- **Número de iteraciones**

Este parámetro depende del tipo de problema. Sin embargo, se debe tener en cuenta que el número de iteraciones debe ser suficiente para no terminar de manera prematura el algoritmo, pero no debe ser excesivo en términos de complejidad computacional.

### **3.5.5 Optimización de Enjambre de partículas con múltiples estructuras sociales: GLNPSO**

Una variación del PSO básico introducida por Pongchairerks y Kachitvichyanukul<sup>68</sup> es el GLNPSO. Éste está compuesto por la combinación de tres estructuras sociales estudiadas anteriormente de manera independiente: Global Best, Local Best y Near Neighbor Best (Ver Tabla 2.)

Los dos primeros comportamientos, el decir, gbest y lbest, fueron abarcados en su totalidad en la sección anterior. Sin embargo, una nueva estructura social es introducida al paradigma.

---

<sup>68</sup> P. Pongchairerks, V. Kachitvichyanukul. A non-homogeneous particle swarm optimization with multiple social structures. Proceedings of the 2005 International Conference on Simulation and Modeling

La estructura de Near-Neighbor best fue presentada por Veeramachaneni como una alternativa para representar la influencia social de unas partículas sobre otras. De esta manera, el efecto de la influencia múltiple entre las partículas crea la posibilidad de existencia de ‘conversaciones’ cruzadas entre las redes neuronales de aprendizaje

**Tabla 2. Estructuras sociales involucradas en la metaheurística GLNPSO**

<b>ESTRUCTURA SOCIAL</b>	<b>DESCRIPCIÓN</b>
<b>Global Best</b>	Se denota como $\Psi_g$ , y corresponde a la mejor posición obtenida por las L partículas del enjambre.
<b>Local Best</b>	Es denotada como $\Psi_i^L$ , y se define como la mejor posición obtenida entre los K vecinos inmediatos.
<b>Near neighbor Best</b>	Se denota como $\Psi_i^N$ , y utiliza una relación <i>fitness-distance-ratio</i> entre las partículas del enjambre.

**Fuente:** Autores

Si bien el PSO básico contempla ésta influencia, existen dos planteamientos tomados del comportamiento biológico de las especies que justifican el cálculo de esta estructura particular. Éstos son:

1. Un organismo tiene mayor tendencia a ser influenciado por su vecindario.
2. Entre los miembros de un vecindario, aquel organismo que obtiene más éxito en su experiencia es más apto para generar esa influencia social en los demás.

. Esto supone que cada uno de los pulsos generados en dichas interacciones entre varias partículas, podrían anularlas desaprovechando el beneficio de los movimientos efectuados por cada una de ellas.

Para contrarrestar esa posible degeneración del comportamiento, la estructura del Near-Neighbor contempla la existencia de una función FDR (Fitness-distance-

ratio), la cual selecciona sólo una partícula de interacción para cada iteración del algoritmo, basando esta decisión en el cumplimiento de dos criterios:

1. La partícula seleccionada debe estar cerca a la partícula que va a ser actualizada.
2. La partícula seleccionada debe haber obtenido un valor más alto en la función *fitness*.

Para cumplir estos criterios, la partícula seleccionada debe maximizar el radio de la diferencia entre el valor *fitness* y la distancia a una dimensión de la partícula.

$$FDR = \frac{Z(\Theta_l) - Z(\Psi_j)}{|\psi_{jl} - \theta_{lh}|}$$

En donde:

$\Theta_l$  : Posición de la partícula  $l$

$\Psi_j$  : Mejor posición de la partícula

$\psi_{jl}$  : Mejor posición de la partícula  $l$  en la dimensión  $j$

$\theta_{lh}$  : Posición actual de la partícula  $l$  en la dimensión  $h$

$l = 1, 2, \dots, L$

$h = 1, 2, \dots, H$

$L$  : Número de partículas

$H$  : Número de dimensiones de la partícula

En el algoritmo 1, se presenta el pseudo-código del algoritmo con la estructura Near-Neighbor Best y Global Best como comportamientos sociales del enjambre. Este algoritmo recibe el nombre de FDR-PSO.

Sin embargo, en la ejecución del GLNPSO, la adición de múltiples estructuras sociales modifica también la conformación del vector de velocidad de cada partícula ya que no involucra un componente personal y uno social, sino uno personal y tres de tipo social.

Por lo tanto, la actualización de la velocidad para la ejecución del GLNPSO se denota como:

$$\omega_{lh}(\tau+1) = w(\tau)\omega_{lh}(\tau) + c_p u_p (\psi_{lh} - \theta_{lh}(\tau)) + c_g u_g (\psi_{gh} - \theta_{lh}(\tau)) + c_l u_l (\psi_{lh}^L - \theta_{lh}(\tau)) + c_n u_n (\psi_{lh}^N - \theta_{lh}(\tau))$$

En donde, además de la notación presentada en ecuaciones previas se incluyen:

$c_n$  : Constante de aceleración de Near neighbor Best

$u_n$  : Número aleatorio entre 0 y 1 para la estructura Nbest

$\psi_{lh}^N$  : Posición Near – neighbor best

#### 4. DISEÑO DE LA SOLUCIÓN

Con el objetivo de encontrar mejores soluciones del problema VRPSPD y basados en los algoritmos de Ai y Kachitvichanukul (2009) para resolver el problema del VRPSPD, se propone la aplicación de un algoritmo basado en la metaheurística PSO con múltiples estructuras sociales (GLNPSO) junto con heurísticas de construcción y mejoramiento de rutas que operan en las fases de codificación de la solución, como se especificará a continuación. El programa desarrollado se efectuó con la ayuda de la herramienta MatLab R2012a.

##### Algoritmo 1. FDR-PSO

**Iniciar Algoritmo**

**Para cada iteración,  $\tau = 1 \dots T$ ,**

**Para cada partícula del enjambre,  $l = 1 \dots L$ ,**

**Para todas las dimensiones del problema,  $h = 1 \dots H$ ,**

*Aplicar la ecuación de actualización de velocidad:*

$$\omega_{lh}(\tau+1) = w(\tau)\omega_{lh}(\tau) + c_p u_p (\Psi_{lh} - \theta_{lh}(\tau)) + c_g u_g (\Psi_{gh} - \theta_{lh}(\tau)) + c_n u_n (\Psi_{lh}^N - \theta_{lh}(\tau)),$$

*en donde,  $\Psi_{lh}$ , es la mejor posición visitada hasta entonces por cada partícula,*

*$\Psi_{gh}$ , es la mejor posición visitada por alguna de las partículas,*

*y,  $\Psi_{lh}^N$ , es la posición que maximiza la función:*

$$FDR = \frac{Z(\Theta_i) - Z(\Psi_j)}{|\Psi_{ji} - \theta_{lh}|}$$

*Actualizar la posición:*

$$\theta_{lh}(\tau+1) = \theta_{lh}(\tau) + \omega_{lh}(\tau+1)$$

**Fin H dimensiones**

*Calcular desempeño de la función fitness:  $Z(\Theta_i(\tau+1))$*

*Si es necesario, actualizar la información histórica correspondiente a  $\Psi_{lh}$  y  $\Psi_{gh}$*

**Fin L partículas**

*Terminar si  $\Psi_{gh}$  alcanza los requerimientos del problema.*

**Fin T iteraciones.**

**Terminar algoritmo**

**Fuente:** Autores

A partir de la formulación matemática del problema VRPSPD se considera el criterio de minimización de costos, lo que implica que el algoritmo desarrollado tiene como característica primordial la minimización. El criterio de parada definido para el algoritmo GLNPSO con múltiples estructuras sociales aplicado en este proyecto ha sido el número de iteraciones.

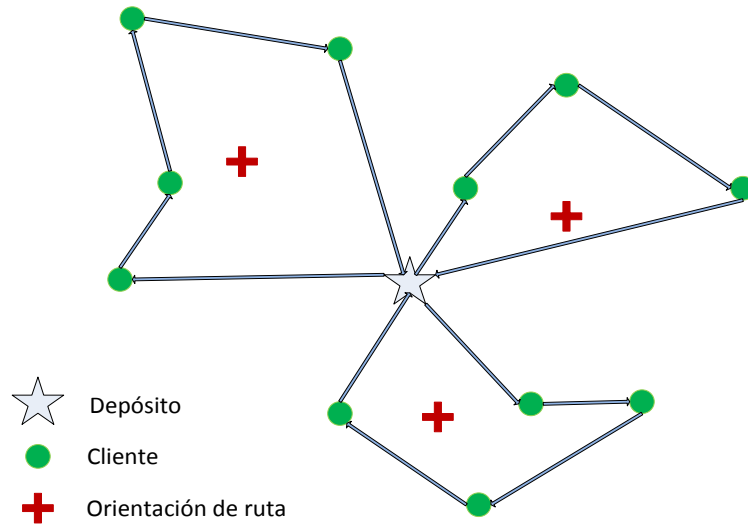
#### **4.1 REPRESENTACIÓN DE LA SOLUCIÓN.**

Una *codificación* de la solución de un problema se utiliza para representarla en los términos del método de solución aplicado. La representación de la solución del VRPSPD a través del paradigma PSO consiste de dos partes en las cuales se usa la posición de las partículas con muchas dimensiones para representar una solución en un vector codificado, cuyos valores son números reales: La primera parte del vector codificado está relacionada a los clientes y la segunda a los vehículos.

La primera parte de la representación se usa para establecer la prioridad de los clientes para ingresar a las rutas en el paso de construcción de rutas. Esta primera parte consta de  $n$  dimensiones de la partícula (dimensiones en el espacio de búsqueda), en la que cada dimensión es asignada a un cliente. El menor valor de la dimensión corresponde a la mayor prioridad de asignación.

La segunda parte de la representación está basada en la idea de *orientación de rutas de los vehículos*. Esta orientación de rutas se define como un punto en el mapa de servicio que representa cierta área en la cual el vehículo tiende a servir. Consecuentemente, un vehículo tendrá una mayor tendencia a visitar a los clientes que estén en los alrededores de su correspondiente punto de orientación. En la ilustración 13 se muestra la relación del área que recorre un vehículo representada por su punto de orientación.

### Ilustración 13. Rutas y puntos de orientación de rutas



Fuente: Autores

Un punto de orientación de la ruta se define por el par de coordenadas  $x$ - $y$  en el mapa de servicios. Cada punto de orientación se representa por dos dimensiones de una partícula, una dimensión para el valor de la coordenada en  $x$  y otra dimensión para el valor de la coordenada en  $y$ . De esta manera, la representación deberá consistir en  $2m$  dimensiones de las partículas que corresponden a la flota de  $m$  vehículos disponibles.

La orientación de las rutas se utiliza como otra base para la construcción de rutas, adicional a la mencionada prioridad de clientes. Una vez todos los puntos de orientación de las rutas se identifican, se determina la preferencia de los vehículos para servir a cada uno de los clientes, basado en la distancia del cliente al punto de orientación. Estas preferencias se definen para asegurar la cercanía espacial entre los clientes de una ruta, teniendo en cuenta que la cercanía entre los clientes y el punto de referencia de la ruta se mantiene. Así, la distancia total de la ruta será más corta y su correspondiente costo será minimizado.

De esta manera, la representación de la solución para el VRPSPD para  $n$  clientes y  $m$  vehículos requiere partículas con  $n+2m$  dimensiones. Las primeras  $n$  dimensiones corresponden al número de clientes y las siguientes  $2m$  relacionadas con la flota de vehículos. Por ejemplo, si el problema consta de 8 clientes y una flota de 2 vehículos, las dimensiones del enjambre serán las mostradas en la ilustración 14.

**Ilustración 14. Vector posición de una partícula del enjambre**



**Fuente: Autores**

## 4.2 DESCRIPCIÓN DE LA METAHEURÍSTICA GLNPSO

La idea de utilizar la metaheurística PSO con múltiples estructuras sociales surge del intento de encontrar mejores soluciones al problema del VRPSPD. Los múltiples componentes sociales permiten a las partículas del enjambre tener una mayor robustez para encontrar buenas soluciones dada por la combinación de las características de cada uno de los componentes. Como se dijo en el numeral 3.5.5, las partículas del *Global Best PSO* convergen con mayor rapidez pero con el inconveniente de tener una menor diversidad en la exploración del espacio de búsqueda, contrario a lo que ofrece el *Local Best PSO* que tiene un mayor grado de diversificación pero con convergencia menor. El *Near Neighbor Best*, adicionalmente, involucra la interacción existente entre partículas a través de la elección de partículas para actualizar la velocidad de manera que se maximice la función FDR definida anteriormente.

Los detalles del algoritmo GLNPSO para resolver el problema del VRPSPD se muestran en el algoritmo 2.

### Notación

$\tau$	índice de la iteración; $\tau = 1 \dots T$
$l$	índice de la partícula; $l = 1 \dots L$
$h$	índice de la dimensión; $h = 1 \dots H$
$u$	número aleatorio con distribución uniforme $[0,1]$
$w(t)$	peso inercial de la $\tau$ –ésima iteración
$\omega_{lh}(t)$	velocidad de la $l$ -ésima partícula en la $h$ -ésima dimensión en la $\tau$ –ésima iteración
$\theta_{lh}(t)$	posición de la $l$ -ésima partícula en la $h$ -ésima dimensión en la $\tau$ –ésima iteración
$\psi_{lh}$	mejor posición personal (pbest) de la $l$ -ésima partícula en la $h$ -ésima dimensión
$\psi_{gh}$	mejor posición global (gbest) en la $h$ -ésima dimensión
$\psi_{lh}^L$	mejor posición local (lbest) de la $l$ -ésima partícula en la $h$ -ésima dimensión
$\psi_{lh}^N$	mejor posición de vecindario (nbest) de la $l$ -ésima partícula en la $h$ -ésima dimensión
$c_p$	constante de aceleración de la posición pbest
$c_g$	constante de aceleración de la posición gbest
$c_l$	constante de aceleración de la posición lbest
$c_n$	constante de aceleración de la posición nbest
$\theta_{\max}$	Valor de posición máximo
$\theta_{\min}$	Valor de posición mínimo
$\Theta_l$	vector posición de la $l$ -ésima partícula, $[\theta_{l1}, \theta_{l2}, \dots, \theta_{lH}]$
$\Omega_l$	vector velocidad de la $l$ -ésima partícula, $[\omega_{l1}, \omega_{l2}, \dots, \omega_{lH}]$
$\Psi_l$	vector posición pbest de la $l$ -ésima partícula, $[\psi_{l1}, \psi_{l2}, \dots, \psi_{lH}]$
$\Psi_g$	vector posición gbest de la $l$ -ésima partícula, $[\psi_{g1}, \psi_{g2}, \dots, \psi_{gH}]$
$\Psi_l^L$	vector posición lbest de la $l$ -ésima partícula, $[\psi_{l1}^L, \psi_{l2}^L, \dots, \psi_{lH}^L]$
$R_l$	el $l$ -ésimo conjunto de rutas de vehículos
$Z(\Theta_l)$	Valor de desempeño de $\Theta_l$
$FDR$	fitness-distance-ratio

## Algoritmo de Decodificación.

Un método de *decodificación* se utiliza para convertir la codificación de los componentes de la heurística en una solución al problema original. El algoritmo de decodificación utilizado consta de tres pasos: Primero, extraer la información de las primeras  $n$  dimensiones para construir una lista de prioridad de clientes. Segundo, tomar la información de las últimas  $2m$  dimensiones para determinar los puntos de orientación de las rutas y usarla para crear la matriz de prioridad de los vehículos. Tercero, construir las rutas basándose en la lista de prioridad de clientes y la matriz de prioridad de los vehículos.

### Algoritmo 2. GLNPSO

#### Comenzar Programa

$\tau = 1$

para todas las partículas inicializar  $\Theta_i$ ,  $\Omega_i = 0$  y  $\Psi_i = \Theta_i$

decodificar  $\Theta_i(\tau)$  en un conjunto de rutas  $R_i$

guardar las mejores posiciones  $pbest$ ,  $gbest$ ,  $lbest$  y generar  $nbest$

**mientras** ( $\tau < T$ )

actualizar *inercia, velocidad y posición de las partículas*

$$w(\tau) = w(T) + \frac{\tau - T}{1 - T} [w(1) - w(T)]$$

$$\begin{aligned} \omega_{jh}(\tau+1) = & w(\tau)\omega_{jh}(\tau) + c_p u(\Psi_{jh} - \theta_{jh}(\tau)) + c_g u(\Psi_{gh} - \theta_{jh}(\tau)) \\ & + c_i u(\Psi_{ih}^I - \theta_{jh}(\tau)) + c_n u(\Psi_{jh}^N - \theta_{jh}(\tau)) \end{aligned}$$

$$\theta_{jh}(\tau+1) = \theta_{jh}(\tau) + \omega_{jh}(\tau+1)$$

Evaluar y decodificar el nuevo  $\Theta_i(\tau)$  en un conjunto de rutas  $R_i$

Actualizar y guardar las mejores soluciones del enjambre  $pbest$ ,  $gbest$ ,  $lbest$  y generar  $nbest$

$\tau = \tau + 1$

**fin\_mientras**

**fin\_programa**

Fuente: Autores

- 1) En el primer paso, se construye la lista de prioridad de clientes siguiendo la regla de mayor prioridad al cliente con menor valor en la dimensión correspondiente, a partir de las primeras  $n$  dimensiones de la partícula. La manera más simple de implementar esta regla es ordenando en orden ascendente los valores de posición de las  $n$  dimensiones y tomando el índice de dimensión como la lista de prioridades.
- 2) El siguiente paso es extraer los puntos de orientación de las rutas y construir la matriz de prioridad de los vehículos. La matriz se construye a partir de la distancia relativa entre los puntos de orientación y los clientes, los cuales pueden ser todos graficados en un mismo plano cartesiano. Un cliente es servido, en primer lugar, por el vehículo cuyo punto de referencia esté más cercano. Cada columna de la matriz contiene la prioridad de los vehículos para clientes con la misma prioridad.
- 3) El último paso de decodificación es construir rutas con base en la lista de prioridad de clientes y la matriz de prioridad de vehículos. Uno a uno, los clientes en la lista de prioridad de vehículos son asignados a los vehículos según su prioridad y sus restricciones de capacidad del vehículo y límite de duración de servicio. Cada nueva asignación de un cliente es insertada en la mejor posición de la ruta existente basado en el menor costo adicional, con la ayuda de la *heurística de inserción más barata*. Una vez el cliente sea asignado, se aplica una heurística de re-optimización para mejorar la ruta emergente. Esta heurística es el método conocido como *2-opt*.

Los detalles del procedimiento de decodificación se describen en el algoritmo 3.

#### **4.2.1 Heurística de Inserción más barata (Cheapest insertion heuristic)**

A continuación se describe la heurística de inserción más barata (cheapest insertion heuristic) utilizada para la construcción de las rutas en el método de decodificación. La heurística parte de rutas existentes para poder evaluar qué posición representa el menor costo adicional de la ruta con los nuevos clientes que ingresan a la secuencia.

Esta heurística evalúa todas las posiciones posibles en donde puede ser insertado un nuevo cliente en una ruta existente en la que haya al menos 1 cliente asignado previamente. Existen  $n+1$  posibilidades de ubicación de un nuevo cliente, siendo  $n$  el número de clientes que hacen parte de la ruta existente. El código de programación de la heurística se expresa en el algoritmo 4.

Es importante resaltar que la heurística parte de una ruta previamente construida. La rutina está diseñada para construir una secuencia inicial, agregando al cliente quien tiene el vehículo asociado con mayor prioridad para servirlo. Es decir, si un vehículo no tiene aún asignado ningún cliente, éste se asigna directamente sin la ejecución de la heurística de inserción más barata.

Para un mejor entendimiento de este procedimiento, se ilustrará a continuación un ejemplo de construcción de una ruta.

### Algoritmo 3. Método de decodificación

```

Comenzar programa
  Construir lista de prioridad de Clientes ( $U$ )
    Construir el conjunto  $S = (1, 2, \dots, n)$  y  $U = \emptyset$ 
    mientras  $S \neq \emptyset$ 
      Seleccionar cliente  $c$  del conjunto  $S$  tal que corresponda a la menor dimensión de  $\theta_{ik}$ 
      Agregar  $c$  a la última posición de  $U$ 
      Eliminar  $c$  del conjunto  $S$ .
    fin_mientras
  Construir matriz de prioridad de vehículos. ( $W$ )
    Para  $j = 1 \dots m$ , definir la posición de referencia de los  $m$  vehículos.
       $xref_j = \theta_{i,n+2j-1}$  y  $yref_j = \theta_{i,n+2j}$ 
    Para cada cliente  $i$ ,  $i = 1 \dots n$ 
      Calcular la distancia euclidiana entre el cliente  $i$  y los todos puntos de orientación
      mientras  $S \neq \emptyset$ 
        Construir conjunto  $S = (1, 2, \dots, m)$  y  $W_i = \emptyset$ 
        Seleccionar vehículo  $c$  del conjunto  $S$  tal que tenga la menor distancia euclidiana al cliente  $i$ .
        Agregar  $c$  a la última posición del conjunto  $W_i$ 
      fin_mientras
  Construir rutas de vehículos
     $k = 1$ 
    mientras  $k < n$ 
      agregar clientes uno a uno a la ruta
         $c = U_k$ ,  $b = 1$ 
        mientras  $b < m$ 
           $j = W_{c,b}$ 
          Hacer a  $c$  como candidato a insertar en la posición que genere el menor costo adicional en la ruta  $R_y$ , según heurística de inserción más barata.
          Evaluar la carga y el tiempo de servicio de la nueva ruta.
          sí (cumple con condiciones de factibilidad)
            Actualizar la ruta  $R_y$  y reoptimizar con 2-opt
          fin_si
           $b = b + 1$ 
        fin_mientras
       $k = k + 1$ 
    fin_mientras
fin_programa

```

Fuente: Autores

#### Algoritmo 4. Heurística inserción más barata.

```

Comenzar programa
  Seleccionar cliente a asignar
  Seleccionar vehículo donde se asignará el cliente y obtener la secuencia.
  long = número de clientes de la ruta
  Asignar al nuevo cliente a la secuencia de la ruta, en la última posición
    secuencia ( long + 1 ) = cliente
  Calcular el costo de la nueva ruta
  Índice = long
  si (la ruta del vehículo existe antes de insertar el nuevo cliente, long ≥ 1)
    ejecuta la heurística
    para ( n-1 intercambios posibles de inserción, j = 1...n-1 )
      a = secuencia ( índice )
      secuencia ( índice ) = secuencia ( índice + 1 )
      secuencia ( índice + 1 ) = a
      calcula el nuevo costo de secuencia luego de hacer el cambio
      si ( nuevo costo < costo antes del cambio )
        Actualiza la secuencia con el cambio realizado
      fin_si
      índice = índice - 1
    fin_para
  fin_si
fin_programa

```

Fuente: Autores

#### Ejemplo ilustrativo de cheapest insertion heuristic

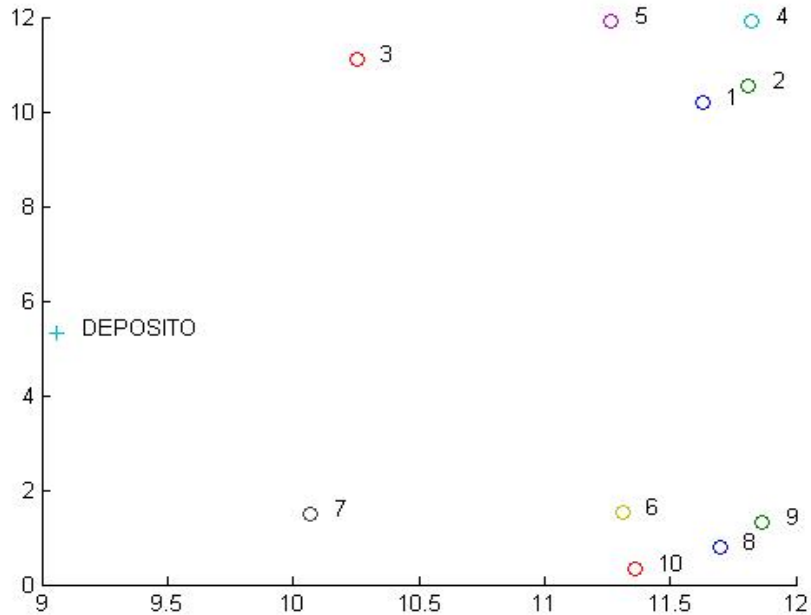
Se tienen 10 clientes que deberán ser insertados en una sola ruta (*para el ejemplo se tomará 1 solo vehículo*) de manera que se minimice la distancia total de la ruta. La ubicación geográfica de los clientes se muestra en la ilustración 15. Los clientes se insertarán uno a uno a la ruta según el orden de prioridad definido en la tabla 3.

**Tabla 3. Lista de prioridad problema ejemplo**

<i>prioridad</i>	1	2	3	4	5	6	7	8	9	10
<i>cliente</i>	2	7	5	10	1	3	8	6	9	4

En la tabla 4 se muestra la matriz de distancias entre todos los clientes y el depósito.

**Ilustración 15. Problema Ejemplo cheapest insertion**



Fuente: Autores

**Tabla 4. Matriz de distancias ejemplo problema**

	1	2	3	4	5	6	7	8	9	10	Depósito
1	0	0.40516247	1.6430331	1.73121288	1.77262468	8.68542273	8.84708499	9.41087834	8.88732685	9.85646059	10.8823653
2	0.40516247	0	1.64750358	1.35810194	1.477697	9.0553364	9.23614252	9.77319945	9.24621269	10.2247124	11.2873155
3	1.6430331	1.64750358	0	1.77428394	1.3116877	9.63648399	9.60923225	10.4099861	9.915057	10.8078714	11.2387061
4	1.73121288	1.35810194	1.77428394	0	0.56222709	10.4122907	10.5754422	11.1313013	10.6041381	11.5821513	12.5379678
5	1.77262468	1.477697	1.3116877	0.56222709	0	10.4144018	10.5114651	11.1537519	10.6359437	11.5877749	12.3491506
6	8.68542273	9.0553364	9.63648399	10.4122907	10.4144018	0	1.24040214	0.82704048	0.59289812	1.17400798	4.41637271
7	8.84708499	9.23614252	9.60923225	10.5754422	10.5114651	1.24040214	0	1.77175943	1.80509623	1.72116375	3.22954651
8	9.41087834	9.77319945	10.4099861	11.1313013	11.1537519	0.82704048	1.77175943	0	0.55318319	0.55818633	4.66116304
9	8.88732685	9.24621269	9.915057	10.6041381	10.6359437	0.59289812	1.80509623	0.55318319	0	1.09488751	4.90893181
10	9.85646059	10.2247124	10.8078714	11.5821513	11.5877749	1.17400798	1.72116375	0.55818633	1.09488751	0	4.2970767
Depósito	10.8823653	11.2873155	11.2387061	12.5379678	12.3491506	4.41637271	3.22954651	4.66116304	4.90893181	4.2970767	0

### **Primera iteración de la heurística**

- 1) Obtener el primer cliente a ser agregado de acuerdo a la prioridad. Para el caso, el primer cliente en prioridad es el cliente 2 y por ende será el primer candidato en ser insertado a la ruta.
- 2) Se selecciona la secuencia que hasta el momento está vacía y el vehículo se encuentra estacionado en el depósito.

$$\text{secuencia} = [D]$$

- 3) Longitud de la secuencia = 0
- 4) Se inserta al cliente en la siguiente posición y se calcula el costo

$$\text{secuencia} = [D - 2 - D]$$

$$\text{costo} = 5,917 + 5,917 = 11,834$$

La secuencia se puede observar gráficamente en la ilustración 16. Como es el primer cliente insertado en la ruta, se prosigue a la siguiente iteración.

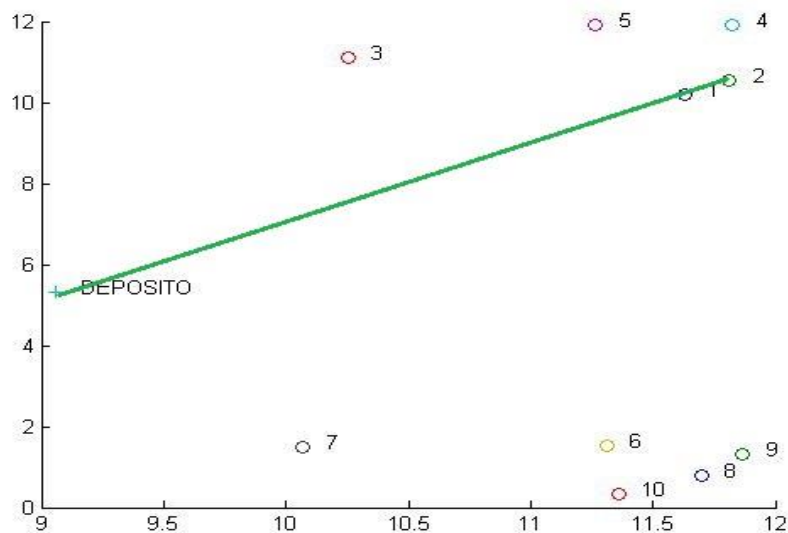
### **Segunda iteración**

- 1) El segundo cliente en prioridad es el cliente 7, quien será el siguiente a ser agregado a la ruta
- 2) La secuencia de la ruta es [ D – 2 – D ]
- 3) La longitud de la secuencia es  $long = 1$
- 4) Se inserta al cliente en la posición (  $long + 1$  ) y se calcula su costo, así:

$$\text{secuencia} = [D - 2 - 7 - D]$$

$$\text{costo} = 5,917 + 9,236 + 3.963 = 19,116$$

**Ilustración 16. cheapest insertion heuristic (it. 1)**



**Fuente:** Autores

5) Como la secuencia ya existía antes de insertar al cliente 7, se puede hacer la *heurística de inserción más barata*, en la que existen  $long - 1$  posibilidades de intercambio respecto a la ruta recién construida. Para el caso, existe otra posibilidad para insertar al cliente:

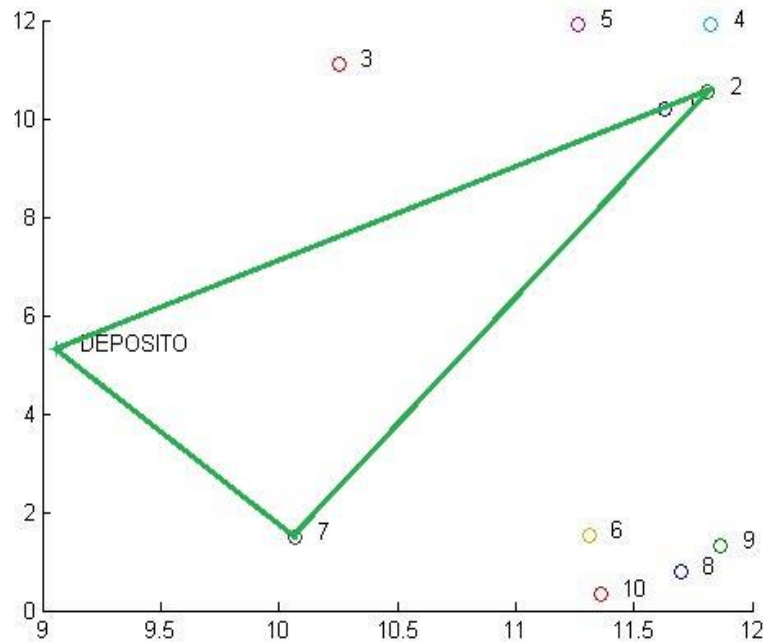
$$\text{nuevasecuencia} = [D - 7 - 2 - D]$$

Para la cual se calcula el nuevo costo:

$$\text{nuevocosto} = 3,963 + 9,236 + 5.917 = \mathbf{19,116}$$

En este caso en particular en el que hay dos clientes, ambas posibilidades de la secuencia forman un triángulo idéntico y recorrer la ruta en cualquier sentido tiene el mismo costo, como se puede apreciar en la ilustración 17. Por defecto se toma la primera opción como la nueva secuencia, para poder seguir con la siguiente iteración.

**Ilustración 17. Cheapest insertion heuristic(it.2)**



**Fuente:** Autores

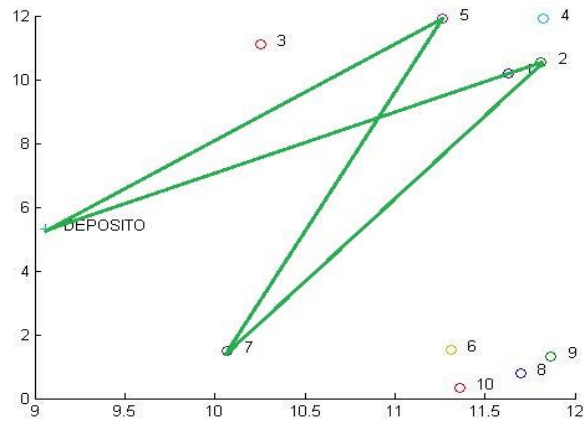
***Tercera iteración***

- 1) El siguiente cliente a ser insertado es el cliente 5 según la lista de prioridades.
- 2) La secuencia actual de la ruta es [  $D - 2 - 7 - D$  ]
- 3)  $Long = 2$
- 4) Se inserta el nuevo cliente en la posición ( $long + 1$ ) (ver ilustración 18.a) y se calcula su costo.

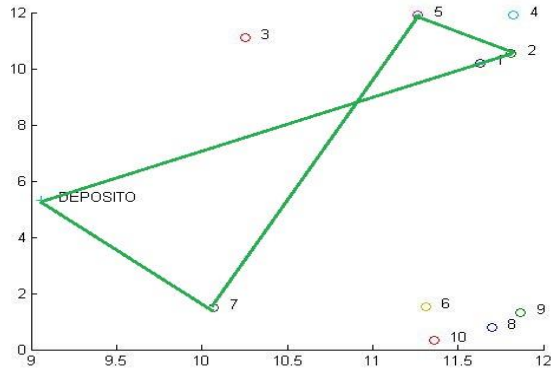
$$\text{secuencia} = [D - 2 - 7 - 5 - D]$$

$$\text{costo} = 5,917 + 9,236 + 10,511 + 6,969 = 32,633$$

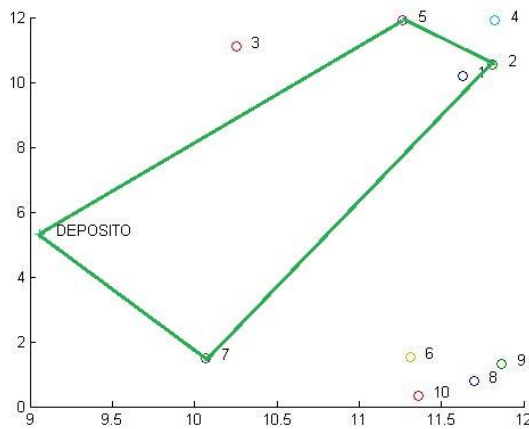
Ilustración 18. Cheapest Insertion (it.3)



**a**



**b**



**c**

Fuente: Autores

5) La heurística de inserción más barata evalúa 2 posibilidades de intercambio a partir de esta última secuencia:

$$\text{Opción 1: } \text{nuevasecuencia} = [D - 2 - 5 - 7 - D]$$

$$\text{nuevocosto} = 5,917 + 1,477 + 10,511 + 3,963 = 21,868$$

Si (nuevocosto < costo)

$$\text{secuencia} = [D - 2 - 5 - 7 - D]$$

$$\text{costo} = 21,868$$

En este intercambio se encontró una mejor solución pues el costo total de la secuencia disminuyó (ver ilustración 18.b). Ahora, la nueva secuencia será la correspondiente a la mejora para continuar con los intercambios necesarios

$$\text{Opción 2: } \text{secuencia} = [D - 5 - 2 - 7 - D]$$

$$\text{nuevocosto} = 6,969 + 1,477 + 9,236 + 3,963 = 21,645$$

Si (nuevocosto < costo)

$$\text{secuencia} = [D - 5 - 2 - 7 - D]$$

$$\text{costo} = 21,645$$

Una vez más, se encontró una secuencia con menor costo total, por lo tanto se toma ésta como la secuencia definitiva en esta iteración (ver ilustración 18.c).

### ***Criterio de parada***

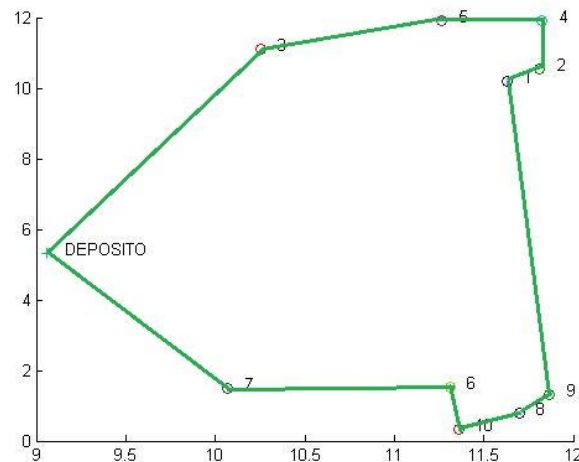
El algoritmo se detiene en el momento que inserte a todos los clientes en la ruta. Para el ejemplo ilustrado, se hizo relajación de las restricciones de capacidad del vehículo con el objetivo de mostrar únicamente el funcionamiento de la heurística de inserción más barata. Sin embargo, al tratar de solucionar el problema considerando todas las restricciones, existe la posibilidad de que queden clientes sin asignar, ya que exceden la capacidad de carga de los vehículos. Esta situación puede ser corregida si se cuentan con más vehículos disponibles. Los clientes que no puedan ser ingresados a la ruta de un vehículo, serán inmediatamente

ingresados a otra ruta de un vehículo diferente que cuente con capacidad disponible.

Al finalizar la heurística, la mejor secuencia encontrada para el problema de 10 clientes es la siguiente y se observa en la ilustración 19:

$$\text{secuencia} = [D - 7 - 6 - 10 - 8 - 9 - 1 - 2 - 4 - 5 - 3 - D]$$

**Ilustración 19. Cheapest insertion ruta final**



**Fuente:** Autores

Gráficamente se puede observar que la secuencia tiene un patrón deseable pues se no se aprecian cruces entre los arcos ni grandes saltos inesperados en la ruta. En este momento se podría pensar que la solución arrojada por al heurística de inserción más barata es fiable y eficaz. Sin embargo, es posible encontrar secuencias no deseables para ciertos problemas, como la observada en la ilustración 19. Este tipo de situación puede ser atribuida al hecho de que a pesar de garantizar la inserción de los clientes en la posición menos costosa, el orden como los clientes son insertados influye directamente en la secuencia construida. El orden definido para insertar a los clientes está basado en el valor de las dimensiones de la partícula, y se ve afectado únicamente por el movimiento del

enjambre. Para solucionar este problema, se aplica una heurística de mejoramiento de búsqueda local, conocida como *2-opt*.

#### 4.2.2 Heurística 2-opt

Las heurísticas de mejoramiento, como el método *2-opt*, son efectuadas a partir de una solución inicial factible arrojada por algún procedimiento de construcción de rutas, para la cual se realizan modificaciones hasta encontrar una mejor solución. La base para las heurísticas de mejoramiento es la búsqueda local.

Se propone el método *2-opt* con el fin de re optimizar las rutas que se construyen en la fase de decodificación de la solución. Como se puede observar en el algoritmo 2, siempre que un nuevo cliente sea insertado en una ruta con el menor costo adicional y la secuencia es factible, se mejora con *2-opt*. Para un mejor entendimiento del procedimiento de mejoramiento de la ruta, se especifica el método en el algoritmo 5.

A manera de ejemplo, se quiere aplicar la heurística *2-opt* para mejorar la ruta que se muestra en la ilustración 20, la cual tiene la siguiente secuencia:

$$\text{secuencia} = [D - 18 - 48 - 25 - 24 - 57 - 20 - D]$$

La secuencia es la ruta de un vehículo que hace parte de la solución de una instancia de 100 clientes para el VRPSPD, cuya matriz de distancias entre los clientes involucrados se muestra en la tabla 5.

**Tabla 5. Matriz de distancias ejemplo para 2-opt**

	<i>Deposito</i>	<i>18</i>	<i>20</i>	<i>24</i>	<i>25</i>	<i>48</i>	<i>57</i>
<i>Deposito</i>	0	45.177428	35.0570963	35.0570963	45.2769257	45.0444225	26.925824
<i>18</i>	45.177428	0	10.198039	11.6619038	9	2	24.4131112
<i>20</i>	35.0570963	10.198039	0	4	12.2065556	10	15.6204994
<i>24</i>	35.0570963	11.6619038	4	0	10.4403065	10.7703296	12.8062485
<i>25</i>	45.2769257	9	12.2065556	10.4403065	0	7	20.6155281
<i>48</i>	45.0444225	2	10	10.7703296	7	0	23.3238076
<i>57</i>	26.925824	24.4131112	15.6204994	12.8062485	20.6155281	23.3238076	0

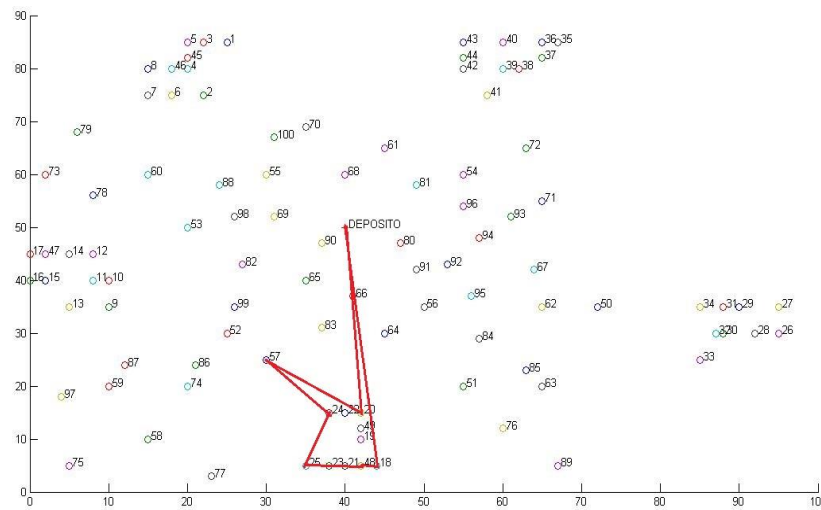
## Algoritmo 5. Heurística 2-opt

```
comenzar_programa
  sea secuencia la ruta que será optimizada
  calcular  $n =$  longitud de la secuencia
   $z_{min} = -1$ 
  mientras ( $z_{min} < 0$ ), mientras el tour sea 2-óptimo
     $z_{min} = 0$ 
     $i = 1$ 
     $b =$  secuencia ( $i$ )
    mientras ( número de nodos  $\geq 4$  )
       $a = b$ 
       $i = i + 1$ 
       $b =$  secuencia ( $i$ )
      calcular distancia entre los nodos  $a$  y  $b$  ( $D_{ab}$ )
       $j = j + 1$ 
       $d =$  secuencia ( $j$ )
      mientras ( $j < n$ )
         $c = d$ 
         $j = j + 1$ 
         $d =$  secuencia ( $j$ )
        calcular distancia entre nodos  $a$  y  $c$  ( $D_{ac}$ ),  $c$  y  $d$  ( $D_{cd}$ ),  $b$  y  $d$  ( $D_{bd}$ ).
        Calcular diferencia de costos entre ruta original y cambio realizado
          
$$z = (D_{ac} + D_{bd}) - (D_{ab} + D_{cd})$$

        Si ( $z < z_{min}$ ) Si la diferencia es menor que cero
           $z_{min} = z$ 
           $i_{min} = i$ 
           $j_{min} = j$ 
        fin_si
      fin_mientras
    fin_mientras
    Aplicar cambio si el costo disminuyó
    si ( $z_{min} < 0$ )
      Secuencia ( $i_{min} : j_{min} - 1$ ) = secuencia ( $j_{min} - 1 : -1 : i_{min}$ )
    fin_si
  fin_mientras
  calcular costo de la nueva secuencia formada
fin_programa
```

Fuente: Autores

## Ilustración 20. Ruta a optimizar con 2-opt



Fuente: Autores

### Primera iteración

Longitud de la secuencia  $n = 7$

$z_{\min} = -1$

**Mientras** (  $z_{\min} < 0$  ) mientras haya minimización del costo de la ruta

$z_{\min} = 0$

$i = 1$

$b = \text{secuencia}(i)$

**Mientras** (  $i < n - 1$  ) Hacer todos los cambios posibles

$a = b$

$i = 2$

$b = \text{secuencia}(i)$

$D_{ab} = 45,177$

$j = 3$

$d = \text{secuencia}(j)$

**Mientras ( j < n )** Hacer el intercambio del arco a-b con todos los arcos c-d posibles

$$c = d$$

$$j = 4$$

$d = \text{secuencia}(j)$

$$Dac = 45,044, Dcd = 7, Dbd = 9$$

$$z = (Dac + Dbd) - (Dab + Dcd) = 1,867$$

**Si ( z < z.min ) = FALSO**

**Fin\_si**

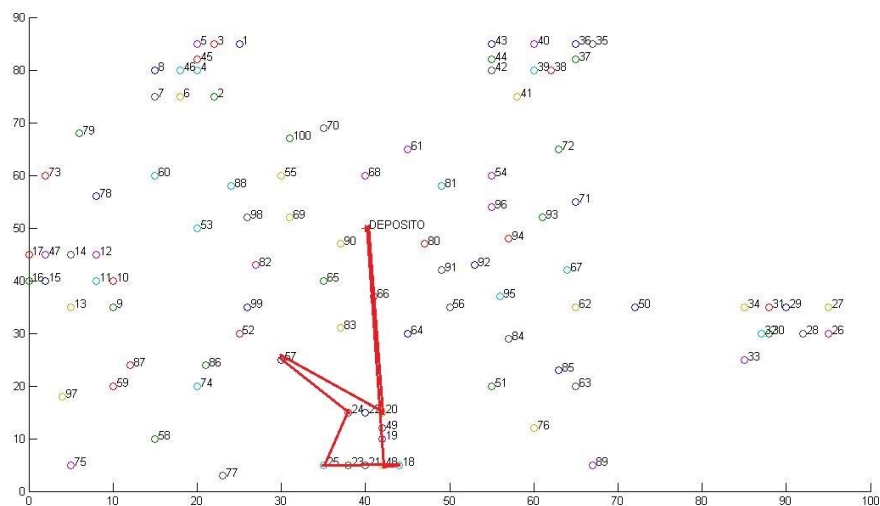
**Fin\_mientras**

**Fin\_mientras**

**Fin mientras**

En este intercambio de arcos (ilustración21), no hubo minimización del costo, así que se continúa con otro intercambio sin actualizar la ruta.

**Ilustración 21. Ejemplo 2-opt (it. 1 - intercambio 1)**



**Fuente: Autores**

**Mientras (  $j < n$  )** Hacer el intercambio del arco a-b con todos los arcos c-d posibles

$$c = d = 25$$

$$j = 5$$

$$d = \text{secuencia}(j) = 24$$

$$Dac = 9, Dcd = 10,44, Dbd = 11,661$$

$$z = (Dac + Dbd) - (Dab + Dcd) = 1,321$$

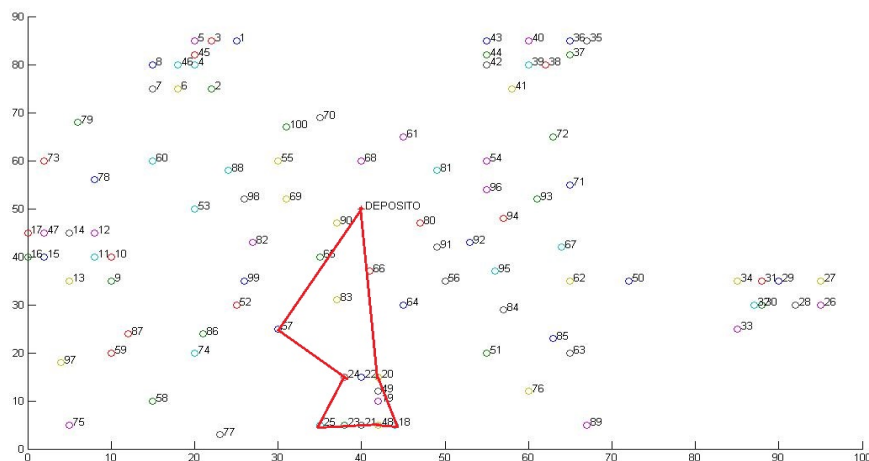
**Si (  $z < z_{\min}$  ) = FALSO**

**Fin\_si**

**Fin\_mientras**

En este intercambio de arcos tampoco hubo minimización del costo, así que se continúa con otro intercambio sin actualizar la ruta. Para esta iteración, el intercambio que ejecuta la heurística con una mayor minimización es la que se realiza con el par [57-20], quedando la secuencia  $\text{secuencia} = [D - 57 - 24 - 25 - 48 - 18 - 20 - D]$ , con una minimización  $z_{\min} = -23,6741$ . En la ilustración 22 se observa la secuencia y se puede evidenciar una ruta con curvas más suaves y menos saltos.

**Ilustración 22. ejemplo 2-opt(it. 1- intercambio final)**



**Fuente: Autores**

## Segunda iteración

**Mientras** (  $z_{\min} < 0$  ) mientras haya minimización del costo de la ruta

$$z_{\min} = -23,6741$$

$$i = 3$$

$$b = \text{secuencia}(i) = 18$$

**Mientras** (  $i < n - 1$  ) Hacer todos los cambios posibles

$$a = b = 18$$

$$i = 3$$

$$b = \text{secuencia}(i) = 48$$

$$D_{ab} = 2$$

$$j = 4$$

$$d = \text{secuencia}(j) = 25$$

**Mientras** (  $j < n$  ) Hacer el intercambio del arco a-b con todos los arcos c-d posibles

$$c = d = 25$$

$$j = 5$$

$$d = \text{secuencia}(j) = 24$$

$$D_{ac} = 9, D_{cd} = 10,44, D_{bd} = 10,77$$

$$z = (D_{ac} + D_{bd}) - (D_{ab} + D_{cd}) = 7,33$$

**Si** (  $z < z_{\min}$  ) = FALSO

**Fin\_si**

**Fin\_mientras**

**Fin\_mientras**

**Fin\_mientras**

Con a y b formando el arco entre los clientes 18 y 48, se continúan evaluando todas las posibilidades de intercambio, hasta el último intercambio que será [c,d] = [57,20].

Al finalizar la heurística haciendo todos los intercambios posibles de arcos, se tiene que la re-optimización de la ruta es posible, siendo la secuencia encontrada en la primera iteración  $\text{secuencia} = [D - 57 - 24 - 25 - 48 - 18 - 20 - D]$ , graficada en la ilustración 22.

### 4.3 BÚSQUEDA DE NÚMERO APROPIADO DE VEHÍCULOS

La representación de la solución propuesta y el método de decodificación están diseñados para solucionar el VRPSPD con un número fijo de clientes ( $n$ ) y número fijo de vehículos ( $m$ ). Si bien en muchos problemas del mundo real el número de clientes es predeterminado y se conoce con anterioridad, el número de vehículos que se utilizarán para servirlos es una variable que podría ser menor que el número de vehículos disponibles.

Una gran ventaja de esta representación es su tendencia a extender uniformemente los vehículos en toda el área de servicio. Es decir, todos los vehículos disponibles son más propensos a ser usados para servir a los clientes. Debido a que el costo fijo es el principal contribuyente al costo total, la reducción del número de vehículos activos sirviendo clientes es necesaria y, por tanto, se propone una rutina adicional para obtener el número apropiado de vehículos. Este procedimiento se implementa como paso de inicialización de las partículas del *algoritmo 2. GLNPSO*.

La idea de esta rutina es reducir el número de vehículos uno a uno mientras se inicializa una partícula, comenzando por un número de vehículos estimado previamente. Esa estimación inicial se realiza sumando las demandas de entrega y recogida de cada uno de los clientes y dividiendo ese total en la capacidad de los vehículos definida en las instancias. Entonces, se genera una representación de la solución con este número preliminar. Una vez se ha decodificado, se intentan remover dos dimensiones de la partícula correspondientes al vehículo que sirve el menor número de clientes, para luego recalcular los clientes servidos por cada una de las rutas, con un vehículo menos. Si la nueva representación alcanza un valor

objetivo menor, se repite el procedimiento removiendo más vehículos hasta que no haya minimización del costo. Finalmente, el número de vehículos ( $m$ ) se ajusta a aquel que arroje un menor valor de la función objetivo. Este número es usado inmediatamente como el número de vehículos disponibles en el proceso para generar las siguientes partículas. En el algoritmo 6 se muestra detalladamente la rutina para determinar el número apropiado de vehículos.

### Algoritmo 6. Rutina para buscar número apropiado de vehículos

**Comenzar\_programa**

Generar una partícula aleatoria para representar  $n$  clientes y  $m$  vehículos, que consiste en  $n + 2m$  dimensiones.  $v = m$ .

Decodificar la partícula en rutas de vehículos usando el *algoritmo 3*. Calcular el desempeño de la solución y definirlo como el valor *fitness* de la partícula,  $Z$ .

$y = 0$

**mientras** ( $y = 0$ ) *mientras haya minimización*

$y = 1$

Calcular el número de clientes servidos por cada vehículo.

Remover las dimensiones correspondientes al vehículo con menor número de clientes en su ruta. Reducir el tamaño de la partícula en 2 dimensiones.  $v = v - 1$ .

Decodificar la partícula actualizada en rutas usando el *algoritmo 3*. Calcular el desempeño de la ruta y definirlo como valor *fitness* de la partícula actualizada,  $Z'$ .

**si** ( $Z' < Z$ )

$Z = Z'$

$y = 0$  ... *hay minimización*.

**else**

$v = v + 1$

**fin\_si**

**fin\_mientras**

$m = v$  Nuevo valor del número de vehículos

**fin\_programa**

**Fuente:** Autores

## 5. RESULTADOS COMPUTACIONALES

Los experimentos numéricos efectuados para evaluar el desempeño de la metaheurística propuesta se llevaron a cabo con la aplicación del método en un conjunto de datos de instancias desarrolladas para el VRPSPD. Se trata de 18 instancias de gran tamaño introducidas por Tang y Galvão<sup>69</sup>. Estas instancias fueron desarrolladas inicialmente por Solomon<sup>70</sup> y Gehring & Homberger<sup>71</sup> para el CVRP que involucran entre 100 y 400 clientes. Tang y Galvão adaptaron estas instancias al VRPSPD, generando aleatoriamente demandas de recolección, en el mismo intervalo usado por Solomon para generar las demandas de entrega. La distancia de los arcos se obtiene calculando la distancia euclidiana entre los vértices, luego la matriz de distancias resultante es simétrica y triangular. El costo fijo por vehículo  $f$  y el costo variable por unidad de distancia  $g$  están definidos en 0 y 1 respectivamente. El límite de duración de servicio  $D$  se calcula generando una solución del Problema del Agente Viajero (TSP) con los clientes de cada una de las instancias, y el número de vehículos  $m$  se calcula luego de totalizar demandas de entrega y recogida de los clientes y dividir sobre la capacidad de los vehículos disponibles. La tabla 6 resume las características de las 18 instancias utilizadas.

El algoritmo es implementado en MatLab, versión R2012a, en un equipo con procesador Intel Core 2 Duo con 4 GB de memoria RAM instalada.

---

<sup>69</sup> Tang FA, Galvao RD. A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service (2006). Op. Cit

<sup>70</sup> Gehring, H., & Homberger, J. A parallel hybrid evolutionary metaheuristics for the vehicle routing problem with time windows. In K. Miettinen, M. Mäkelä, & J. Toivanen (Ed.). Proceedings for EUROGEN99 (Vol. A2(s), pp. 57 – 47 ). Berlin : Springer, 1999.

<sup>71</sup> Solomon, M. M. Algorithms for the vehicles routing and scheduling problems with time window constraints. Operations Research, 35, 254 – 265 (1987)

**Tabla 6. Instancias**

<i>Instancia</i>	<i>n</i>	<i>Q</i>	<i>D</i>	<i>P</i>
r101	100	200	1458	2339
r201	100	1000	1458	2262
c101	100	200	1810	3070
c201	100	700	1810	2910
rc101	100	200	1724	1912
rc201	100	1000	1724	2076
R1_2_1	200	200	3513	4406
R2_2_1	200	1000	3513	4358
C1_2_1	200	200	3530	5370
C2_2_1	200	700	3770	6010
RC1_2_1	200	200	3558	4473
RC2_2_1	200	1000	3558	4299
R1_4_1	400	200	7109	10433
R2_4_1	400	1000	7109	9571
C1_4_1	400	200	7190	12470
C2_4_1	400	700	7560	10050
RC1_4_1	400	200	7127	10065
RC2_4_1	400	1000	7127	10100

**n:** número de clientes, **Q:** Capacidad del vehículo, **D:** Demanda total de entrega, **P:** Demanda total de recogida.

**Fuente:** Autores

Los parámetros utilizados para la aplicación del algoritmo GLNPSO fueron establecidos a partir de resultados de experimentos preliminares realizados para observar el comportamiento del algoritmo en el trabajo de Moghaddam et al. (2012)<sup>72</sup>. Estos parámetros se resumen en la tabla 7.

**Tabla 7. Parámetros del GLNPSO**

<b>Parámetros</b>	<b>Valor</b>
Número de partículas	$N = 50$
Número de vecinos	$K = 5$
Peso inercial Inicial	$w(1) = 0.9$
Peso inercial Final	$w(T) = 0.4$
Constante de aceleración Pbest	$c_p = 0,7$
Constante de aceleración Gbest	$c_g = 0,3$
Constante de aceleración Lbest	$c_l = 1,5$
Constante de aceleración Nbest	$c_n = 1,5$

Se obtuvieron soluciones para cada instancia utilizando diferente número de iteraciones, en su mayoría hasta 200. Sin embargo, en algunos casos no fue posible obtener soluciones después de un tiempo de computación de alrededor de 15 minutos, hasta con 200 iteraciones. En las tablas 8, 9 y 10 se muestra la comparación de los resultados obtenidos con el algoritmo GLNPSO propuesto y la mejor solución conocida para el grupo de instancias mencionado para 100, 200 y 400 clientes respectivamente.

---

<sup>72</sup> Moghaddam, B. F., Ruiz, R., Sadjadi, S. J. Vehicle Routing Problem with uncertain demands: An advanced particle swarm algorithm. Computers & Industrial Engineering 2012; Vol 62. P: 306-317.

En la tabla 11 se muestra la diferencia de la mejor solución obtenida con el algoritmo GLNPSO propuesto, con los resultados de la búsqueda tabú propuesta por Tang-Montanè y Galvão (2006)<sup>73</sup>, la metodología híbrida GLS – Tabu Search de Zachariadis et al. (2009a)<sup>74</sup>, el método evolucionario VLBR de Zachariadis et al (2009b)<sup>75</sup>, y el método APA de Zachariadis et al. (2011)<sup>76</sup>

**Tabla 8. Problemas de 100 clientes**

<i>Problemas de 100 clientes</i>							
<i>PROBLEMA</i>	<i>Capacidad vehículo (Q)</i>	<i>Mejor solución Literatura</i>		<i>Mejor solución GLNPSO</i>		<i>Gap (%)</i>	
		<i>Z</i>	<i>k</i>	<i>Z</i>	<i>k</i>	<i>Z</i>	<i>K</i>
r101	200	1009,95	12	1135.1	13	0.124	0.08
r201	1000	666,2	3	671.17	3	0.007	0
c101	200	1220,18	16	1433.4	17	0.175	0.06
c201	700	662,07	5	664.22	5	0.003	0
rc101	200	1059,32	10	1218.3	11	0.150	0.10
rc201	1000	672,92	3	675.05	3	0.003	0

En los resultados obtenidos para las instancias de 100 clientes, se pudo observar que la desviación entre los resultados obtenidos y los mejores resultados obtenidos en la literatura no superan el 18% para la función de costos. También se puede observar que los menores valores de desviación corresponden a los problemas donde la capacidad de la flota es grande, en las que a su vez se encuentra el mejor número de vehículos en la solución. En todas las instancias de 100 se obtuvieron soluciones con hasta 200 iteraciones.

<sup>73</sup> Tang FA, Galvao RD. A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service (2006). Op. Cit

<sup>74</sup> Zachariadis, E.E., Tarantilis, C.D., & Kiranoudis, C.T. A Hybrid metaheuristic solution approach for the Vehicle Routing Problem with simultaneous delivery and pick-up service. Expert Systems with Applications 36 (2) (2009), pp. 1070 – 1081.

<sup>75</sup> Zachariadis, E.E., Tarantilis, C.D., & Kiranoudis, C.T. An adaptive memory methodology for the vehicle routing problem with simultaneous pick-ups and deliveries. European Journal of Operational Research (2009), doi: 10.1016/j.ejor.2009.05.015.

<sup>76</sup> Zachariadis, E.E., & Kiranoudis, C.T. A local Search metaheuristic algorithm for the vehicle routing problema with simultaneous pick-ups and deliveries. Expert Systems with Applications 38 (3) (2009), pp. 2717-2726.

**Tabla 9. Problemas para 200 clientes**

<i>Problemas de 200 clientes</i>							
<i>PROBLEMA</i>	<i>Capacidad vehículo (Q)</i>	<i>Mejor solución Literatura</i>		<i>Mejor solución GLNPSO</i>		<i>Gap (%)</i>	
		<i>Z</i>	<i>k</i>	<i>Z</i>	<i>k</i>	<i>Z</i>	<i>K</i>
C1_2_1	200	3641,89	28	4463.2	31	0.226	0.11
C2_2_1	700	1726,59	9	1884.7	9	0.092	0.00
R1_2_1	200	3316,94	23	4107.5	25	0.217	0.09
R2_2_1	1000	1665,58	5	1763.20	5	0.059	0
RC1_2_1	200	3316,94	23	4013.5	25	0.210	0.087
RC2_2_1	1000	1560	5	2041.3	5	0.309	0.00

En los resultados obtenidos para las instancias de 200 clientes, se pudo observar que la desviación entre los resultados obtenidos y los mejores resultados obtenidos en la literatura no superan el 31% para la función de costos. El resultado con mayor desviación corresponde a la instancia RC2\_2\_1, para la cual, sin embargo, se encontró el mismo número de vehículos de la mejor solución conocida. También se puede observar que los menores valores de desviación corresponden a los problemas C2\_2\_1 y R2\_2\_1, donde la capacidad de la flota es grande, en las que a su vez se encuentra el mejor número de vehículos en la solución. Particularmente en las instancias C\_1\_2\_1 y RC\_2\_2\_1 no fue posible obtener soluciones con más de 50 y 30 iteraciones en un tiempo de cómputo de 15 minutos.

**Tabla 10. Problemas para 400 clientes**

<i>Problemas de 400 clientes</i>							
<i>PROBLEMA</i>	<i>Capacidad vehículo (Q)</i>	<i>Mejor solución Literatura</i>		<i>Mejor solución GLNPSO</i>		<i>Gap (%)</i>	
		<i>Z</i>	<i>k</i>	<i>Z</i>	<i>k</i>	<i>Z</i>	<i>k</i>
C1_4_1	200	11125,14	63	13800	68	0.240	0.079
C2_4_1	700	3549,2	15	4424.3	16	0.247	0.067
R1_4_1	200	9668,18	53	11891	59	0.230	0.113
R2_4_1	1000	3560,73	10	3997.7	11	0.123	0.10
RC1_4_1	200	9520,06	51	12113	55	0.272	0.078
RC2_4_1	1000	3414,9	11	-	-	-	-

En los resultados obtenidos para las instancias de 400 clientes, se pudo observar en general una mayor desviación entre los resultados obtenidos y los mejores resultados obtenidos en la literatura que en las instancias con menos clientes. Sin embargo, ninguno de las desviaciones supera el 28% para la función de costos.

En cuanto al número de vehículos, se observa que en ningún caso se encontró el mejor número conocido, por lo que se sospecha que la obtención de un buen número de vehículos esté afectada por el número de clientes que tenga la instancia. La desviación para este caso nunca fue mayor del 10%.

Para las instancias C\_1\_4\_1 y RC\_1\_4\_1 no fue posible obtener soluciones con más de 90 y 30 iteraciones. En cuanto a la instancia RC2\_4\_1, no se logró obtener resultados luego de correr el programa por alrededor de 15 minutos.

**Tabla 11. Comparación de Resultados para el VRPSPD con distintos enfoques**

<i>Instancia</i>	TG		GTS		VLBR		APA		GLNPSO	
	Z	k	Z	k	Z	k	Z	k	Z	k
r101	1042,62	12	1019,48	12	1009,95	12	1009,95	12	1135.1	13
r201	671,03	3	666,2	3	666,2	3	666,2	3	671.17	3
c101	1259,79	17	1220,99	16	1220,99	16	1220,18	16	1433.4	17
c201	666,01	5	662,07	5	662,07	5	662,07	5	664.22	5
rc101	1094,15	11	1059,32	10	1059,32	10	1059,32	10	1218.3	11
rc201	674,46	3	672,92	3	672,92	3	672,92	3	675.05	3
R1_2_1	3447,2	23	3393,31	23	3376,3	23	3375,19	23	4107.5	25
R2_2_1	1690,67	5	1673,65	5	1665,58	5	1665,58	5	1763.20	5
C1_2_1	3792,62	29	3652,76	28	3643,82	28	3641,89	28	4463.2	31
C2_2_1	1767,58	9	1735,68	9	1726,59	9	1726,73	9	1884.7	9
RC1_2_1	3427,19	24	3341,25	23	3323,56	23	3316,94	23	4013.5	25
RC2_2_1	1645,94	5	1562,34	5	1560	5	1560	5	2041.3	5
R1_4_1	10027,81	54	9758,77	54	9691,6	54	9668,18	53	11891	59
R2_4_1	3695,26	10	3606,72	10	3572,38	10	3560,73	10	3997.7	11
C1_4_1	11676,27	65	11207,37	63	11179,36	63	11125,14	63	13800	68
C2_4_1	3732	15	3630,72	15	3549,27	15	3549,2	15	4424.3	16
RC1_4_1	9883,31	52	9697,65	52	9645,27	52	9520,06	51	12113	55
RC2_4_1	3603,53	11	3498,3	11	3423,62	11	3414,9	11	-	-

A partir de los resultados, se realizaron procedimientos estadísticos para verificar el comportamiento del algoritmo cuando se varían algunos parámetros tanto de la

metaheurística PSO como del problema VRPSPD. En primer lugar se evaluó el efecto de variar el número de iteraciones que usa el algoritmo propuesto como criterio de parada. Después se verificó si existe algún efecto importante de la capacidad de vehículos y del número de clientes, como parámetros del problema VRPSPD, así como de su interacción en la obtención de soluciones cercanas a la mejor solución conocida para el problema. Estos análisis se llevaron a cabo con la ayuda de la herramienta Statgraphics Centurion XV.

En todos casos, el desempeño del algoritmo se evaluó mediante una métrica definida para el análisis del algoritmo que consiste en el error porcentual entre las soluciones del algoritmo GLNPSO propuesto y las mejores soluciones conocidas de las instancias.

$$Desv\_opt = \frac{Z_{GLNPSO} - Z_{best}}{Z_{best}}$$

Adicionalmente, en el análisis del efecto de la capacidad, y el efecto del número de clientes, se evaluó también el impacto sobre la desviación en el número de vehículos asignados como parte de la solución que arroja el algoritmo.

$$Desv\_K = \frac{K_{GLNPSO} - K_{best}}{K_{best}}$$

Se realizó el análisis de esta manera, consistentemente con los trabajos previos para resolver el VRPSPD en los cual se ignoró el costo fijo de los vehículos. Sin embargo, se ha probado la importancia del costo fijo de los vehículos y se ha considerado como un componente importante de la función objetivo a lo largo de varios estudios en torno a otras áreas del VRP, y es por esta razón que se analizó la variable de salida relacionada con el número de vehículos para evaluar el comportamiento del algoritmo.

## 5.1 ANÁLISIS DE VARIANZA DEL FACTOR NÚMERO DE ITERACIONES

Debido a que los parámetros seleccionados de la literatura para el GLNPSO no incluyeron el número de iteraciones, se realizó un análisis estadístico para determinar el efecto que tienen los niveles de éste factor sobre el desempeño que arroja el algoritmo para solucionar el VRPSPD. Esta es una variable que depende del problema que se quiere resolver, y es importante determinar cuál es el valor más apropiado para aplicarlo al algoritmo respectivo<sup>77</sup>. Se definieron los niveles del número de iteraciones (10, 20, 30, 40, 50 y 60) y se seleccionaron 6 instancias en las cuales se realizó el análisis: dos para 100 clientes, dos para 200 clientes y dos para 400 clientes.

Para llevar a cabo este análisis se diseñó un experimento de un solo factor con 6 niveles y 5 réplicas (30 corridas). Se analizó el efecto del factor en la variable de salida de seis instancias. En la tabla 12 se muestra la secuencia en la que se llevaron a cabo las corridas.

El *modelo de efectos* que describe las observaciones del experimento es:

$$y_{ij} = \mu + \tau_i + \varepsilon_{ij} \begin{cases} i = 1, 2, 3, 4, 5, 6 \\ j = 1, 2, 3, 4, 5 \end{cases}$$

Donde:

$y_{ij}$ :  $j$ -ésima observación cuando el factor número de iteraciones está en el nivel  $i$

$\mu$ : Media *global*

$\tau_i$ : Efecto del  $i$ -ésimo tratamiento

Se quiere verificar el efecto que tiene el factor *Número de iteraciones* en la media global de la variable salida *Desv\_Opt*. Por lo tanto, las hipótesis apropiadas son para probar que los efectos de los tratamientos  $\tau_i$  son cero.

---

<sup>77</sup> Engelbrecht, A. Op. cit. p.312.

$$H_0 : \tau_1 = \tau_2 = \tau_3 = \tau_4 = \tau_5 = \tau_6 = 0$$

$$H_1 : \tau_i \neq 0 \quad \text{para al menos una } i$$

El procedimiento apropiado para probar las hipótesis es el análisis de varianza de un solo factor con efectos fijos. Para cada una de las instancias sometidas a este análisis se construyen pruebas y gráficas para comparar los valores medios de *Desv\_Opt* con los 6 diferentes niveles de número de iteraciones. La prueba-F en la tabla ANOVA determina si hay diferencias significativas entre las medias que indiquen si hay efecto significativo de los diferentes niveles del factor.

**Tabla 12. Aleatorización de las corridas**

<i>Corrida</i>	<i>Nro_iteraciones</i>
1	2
2	1
3	5
4	4
5	1
6	4
7	2
8	6
9	2
10	4
11	2
12	3
13	6
14	1
15	2
16	6
17	4
18	6
19	1
20	5
21	4
22	5
23	5
24	3
25	3
26	3
27	5
28	6
29	1
30	3

**Niveles del Factor:** 1 = 10iteraciones, 2 = 20 iteraciones, 3 = 30 iteraciones, 4 = 40 iteraciones, 5 = 50 iteraciones, 6 = 60 iteraciones

**Instancia c101 (100 clientes)**

**Tabla 13. Instancia c101**

<b>Tratamiento (nivel)</b>	<b>Observaciones</b>					<b>Totales</b>	<b>Promedios</b>
1	0.272	0.297	0.250	0.206	0.247	1.272	0.254
2	0.254	0.249	0.266	0.283	0.271	1.324	0.265
3	0.261	0.246	0.249	0.275	0.269	1.300	0.260
4	0.262	0.261	0.232	0.235	0.188	1.178	0.236
5	0.208	0.204	0.218	0.232	0.247	1.109	0.222
6	0.226	0.230	0.249	0.252	0.252	1.209	0.242

**Tabla 14. ANOVA para Desv\_opt por Nro de iteraciones (c101)**

<b>Fuente</b>	<b>Suma de Cuadrados</b>	<b>Gl</b>	<b>Cuadrado Medio</b>	<b>Razón-F</b>	<b>Valor-P</b>
<i>Entre tratamientos</i>	0.00777146	5	0.00155429	<b>3.61</b>	<b>0.0142</b>
<i>Intra tratamientos</i>	0.0103455	24	0.000431061		
<i>Total (Corr.)</i>	0.0181169	29			

La ANOVA que se muestra en la tabla 14 descompone la varianza de Desv\_opt en dos componentes: un componente entre-tratamientos y un componente dentro-de-tratamientos. La razón-F, que en este caso es igual a 3.60573, es el cociente entre el estimado entre-tratamientos y el estimado dentro-de-tratamientos. Puesto que el valor-P de la prueba-F es menor que 0.05, existe una diferencia estadísticamente significativa entre la media de Desv\_opt entre un nivel de Nro\_iteraciones y otro, con un nivel del 95.0% de confianza. Para determinar cuáles medias son significativamente diferentes de otras, se realiza una prueba de Múltiples Rangos.

## Pruebas de Múltiple Rangos para Desv\_opt por Nro\_iteraciones

Se aplica un procedimiento de comparación múltiple para determinar cuáles medias son significativamente diferentes de otras.

**Tabla 15. Grupos homogéneos c101**

<i>Nro_iteraciones</i>	<i>Casos</i>	<i>Media</i>	<i>Grupos Homogéneos</i>
5	5	0.21784	X
4	5	0.24014	XX
3	5	0.2407	XX
6	5	0.2515	X
1	5	0.2612	X
2	5	0.26694	X

Método: 95.0 porcentaje LSD

En la tabla 15, se han identificado 2 grupos homogéneos según la alineación de las X's en columnas mediante el procedimiento de diferencia mínima significativa (LSD) de Fisher. No existen diferencias estadísticamente significativas entre aquellos niveles que compartan una misma columna de X's.

**Tabla 16. Contrastes entre medias C101**

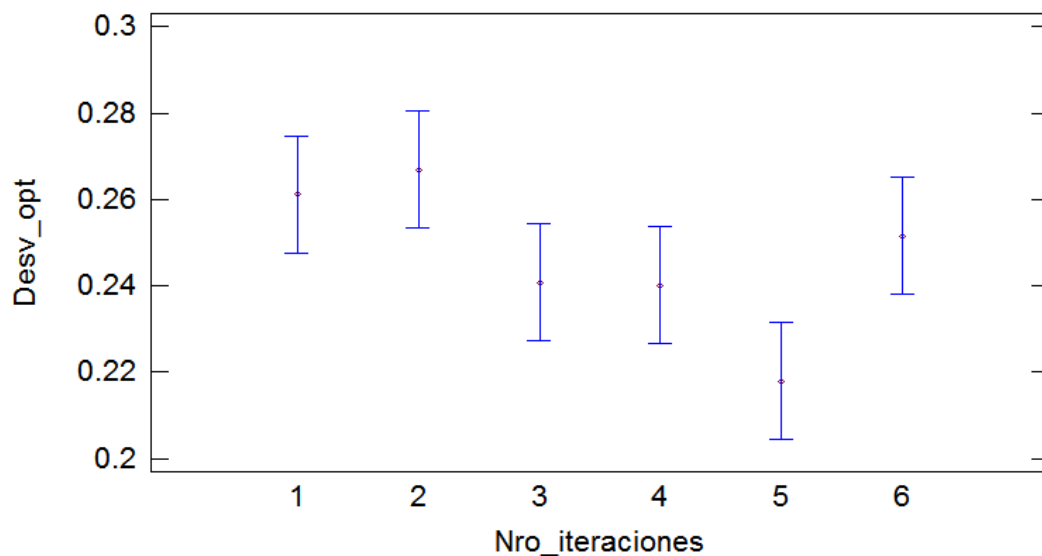
<i>Contraste</i>	<i>Sig.</i>	<i>Diferencia</i>	<i>+/- Límites</i>
1 - 2		-0.00574	0.0271012
1 - 3		0.0205	0.0271012
1 - 4		0.02106	0.0271012
1 - 5	*	0.04336	0.0271012
1 - 6		0.0097	0.0271012
2 - 3		0.02624	0.0271012
2 - 4		0.0268	0.0271012
2 - 5	*	0.0491	0.0271012
2 - 6		0.01544	0.0271012
3 - 4		0.00056	0.0271012
3 - 5		0.02286	0.0271012
3 - 6		-0.0108	0.0271012
4 - 5		0.0223	0.0271012
4 - 6		-0.01136	0.0271012
5 - 6	*	-0.03366	0.0271012

\* indica una diferencia significativa.

La tabla 16 muestra las diferencias estimadas entre cada par de medias. El asterisco que se encuentra al lado de los 3 pares indica que estos pares muestran diferencias estadísticamente significativas con un nivel del 95.0% de confianza.

En la ilustración 22 se observan las medias obtenidas con los diferentes números de iteraciones. Se concluye que con 50 iteraciones en el algoritmo se pueden obtener soluciones con un valor más cercano a la mejor solución encontrada en la literatura, en comparación con 10, 20 y 60 iteraciones.

**Ilustración 22. Gráfico de medias C101**  
Medias y 95.0% de Fisher LSD



**Instancia r201 (100 clientes)**

**Ilustración 23. Instancia r201**

Tratamiento (nivel)	Observaciones					Totales	Promedios
	1	2	3	4	5		
1	0	0.06741234	0.03152552	0.05265371	0.04081147	0.19240303	0.03848061
2	0.04297884	0.02682708	0.0411477	0.04488082	0.03124527	0.18707971	0.03741594
3	0.03062354	0.03242705	0.0394964	0.03389012	0.03275503	0.16919213	0.03383843
4	0.03027965	0.05000465	0.04545347	0.03204623	0.04815461	0.20593861	0.04118772
5	0.03612714	0.02534569	0.0222927	0.03122621	0.02859682	0.14358856	0.02871771
6	0.02795332	0.02358856	0.02534029	0.04519574	0.02279105	0.14486896	0.02897379

**Tabla 17. Tabla ANOVA para Dev\_opt por Nro\_iteraciones (r201)**

<i>Fuente</i>	<i>Suma de Cuadrados</i>	<i>Gl</i>	<i>Cuadrado Medio</i>	<i>Razón-F</i>	<i>Valor-P</i>
<i>Entre tratamientos</i>	0.00150995	5	0.000301989	3.94	0.0094
<i>Intra tratamientos</i>	0.00183762	24	0.0000765676		
<i>Total (Corr.)</i>	0.00334757	29			

La ANOVA de la tabla 17 muestra la descomposición de la varianza de Dev\_opt en dos componentes: el componente entre-tratamientos y el componente dentro-de-tratamientos. La razón-F, que en este caso es igual a 3.94409, es el cociente entre el estimado entre-tratamientos y el estimado dentro-de-tratamientos. Dado que el valor-P de la prueba-F es menor que 0.05, existe una diferencia estadísticamente significativa entre la media de Dev\_opt entre un nivel de Nro\_iteraciones y otro, con un nivel del 95.0% de confianza. Para determinar cuáles medias son significativamente diferentes de otras, se realiza una prueba de Múltiples Rangos.

#### **Pruebas de Múltiple Rangos para Desv\_opt por Nro\_iteraciones**

Se aplica un procedimiento de comparación múltiple para determinar cuáles medias son significativamente diferentes de otras.

**Tabla 18. Grupos homogéneos r201**

<i>Nro_iteraciones</i>	<i>Casos</i>	<i>Media</i>	<i>Grupos Homogéneos</i>
5	5	0.0287177	X
6	5	0.0289738	X
3	5	0.0338384	XX
2	5	0.0374159	XX
4	5	0.0411877	XX
1	5	0.0489453	X

Método: 95.0 porcentaje LSD

En la tabla 18, se han identificado 3 grupos homogéneos según la alineación de las X's en columnas mediante el procedimiento de diferencia mínima significativa (LSD) de Fisher. No existen diferencias estadísticamente significativas entre aquellos niveles que compartan una misma columna de X's.

**Tabla 19. Contrastes entre medias r201**

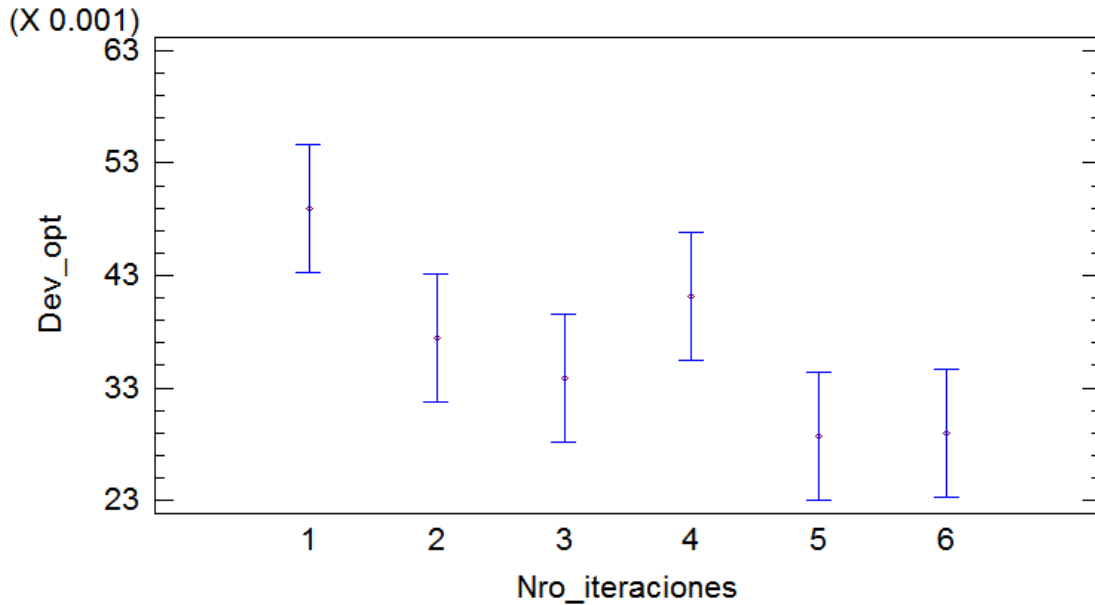
<b>Contraste</b>	<b>Sig.</b>	<b>Diferencia</b>	<b>+/- Límites</b>
1 - 2	*	0.0115293	0.011422
1 - 3	*	0.0151068	0.011422
1 - 4		0.00775755	0.011422
1 - 5	*	0.0202276	0.011422
1 - 6	*	0.0199715	0.011422
2 - 3		0.00357751	0.011422
2 - 4		-0.00377178	0.011422
2 - 5		0.00869823	0.011422
2 - 6		0.00844215	0.011422
3 - 4		-0.00734929	0.011422
3 - 5		0.00512071	0.011422
3 - 6		0.00486464	0.011422
4 - 5	*	0.01247	0.011422
4 - 6	*	0.0122139	0.011422
5 - 6		-0.000256079	0.011422

\* indica una diferencia significativa.

La tabla 19 muestra las diferencias estimadas entre cada par de medias. El asterisco que se encuentra al lado de los 6 pares indica que estos pares muestran diferencias estadísticamente significativas con un nivel del 95.0% de confianza. En la ilustración 24 se observan las medias obtenidas con los diferentes números de iteraciones. Se concluye que existe una diferencia entre obtener valores con 10 iteraciones, comparado con 20, 30, 50 y 60 iteraciones, por lo tanto, se conseguirían peores soluciones con 10 iteraciones. Así mismo existe una diferencia entre obtener soluciones con 40 iteraciones y obtener soluciones con 50 y 60 soluciones. Esto quiere decir, que debería evitarse usar 10 y 40 soluciones para esta instancia.

**Ilustración 24. Gráfico de medias r201**

Medias y 95.0% de Fisher LSD



**Instancia R121 (200 clientes)**

**Ilustración 25. Instancia R121**

Tratamiento (nivel)	Observaciones					Totales	Promedios
	1	2	3	4	5		
1	0.272	0.278	0.249	0.307	0.266	1.372	0.274
2	0.337	0.275	0.273	0.264	0.257	1.406	0.281
3	0.271	0.240	0.275	0.250	0.222	1.258	0.252
4	0.267	0.264	0.293	0.310	0.273	1.408	0.282
5	0.253	0.250	0.264	0.274	0.327	1.369	0.274
6	0.314	0.243	0.260	0.280	0.281	1.379	0.276

**Tabla 20. ANOVA para Desv\_opt por Nro de iteraciones (R121)**

Fuente	Suma de Cuadrados	Gl	Cuadrado Medio	Razón-F	Valor-P
Entre tratamientos	0.00302417	5	0.000604833	0.90	0.4969
Intra tratamientos	0.0161228	24	0.000671783		
Total (Corr.)	0.019147	29			

La ANOVA de la tabla 20 descompone la varianza de Desv\_opt en dos componentes: un componente entre-tratamientos y un componente dentro-de-

tratamientos. La razón-F, que en este caso es igual a 0.90034, es el cociente entre el estimado entre-tratamientos y el estimado dentro-de-tratamientos. Puesto que el valor-P de la razón-F es mayor o igual que 0.05, no existe una diferencia estadísticamente significativa entre la media de Desv\_opt entre un nivel de Nro iteraciones y otro, con un nivel del 95.0% de confianza. Esto es, no existe ningún efecto de los diferentes valores del número de iteraciones en la obtención de un valor objetivo cercano a la mejor solución conocida para la instancia.

**Instancia C221 (200 clientes)**

Tratamiento (nivel)	Observaciones					Totales	Promedios
	1	2	3	4	5		
1	0.2799	0.2654	0.2980	0.2758	0.3505	1.4696	0.2939
2	0.2670	0.2796	0.2597	0.2517	0.3331	1.3912	0.2782
3	0.2821	0.2835	0.2696	0.2331	0.2380	1.3064	0.2613
4	0.2489	0.2600	0.2457	0.2468	0.3027	1.3040	0.2608
5	0.2612	0.2024	0.2336	0.2345	0.2505	1.1822	0.2364
6	0.2413	0.2506	0.2338	0.2309	0.2777	1.2344	0.2469

**Tabla 21. ANOVA para Desv\_opt por Nro de iteraciones (C221)**

Fuente	Suma de Cuadrados	Gl	Cuadrado Medio	Razón-F	Valor-P
Entre tratamientos	0.0108107	5	0.00216214	3.09	0.0272
Intra tratamientos	0.0168052	24	0.000700216		
Total (Corr.)	0.0276159	29			

La ANOVA que se muestra en la tabla 21 descompone la varianza de Desv\_opt en dos componentes: un componente entre-tratamientos y un componente dentro-de-tratamientos. La razón-F, que en este caso es igual a 3.08782, es el cociente entre el estimado entre-tratamientos y el estimado dentro-de-tratamientos. Puesto que el valor-P de la prueba-F es menor que 0.05, existe una diferencia estadísticamente significativa entre la media de Desv\_opt entre un nivel de Nro\_iteraciones y otro, con un nivel del 95.0% de confianza. Para determinar cuáles medias son significativamente diferentes de otras, se realiza una prueba de Múltiples Rangos.

## Pruebas de Múltiple Rangos para Desv\_opt por Nro\_iteraciones

**Tabla 22. Grupos homogéneos C221**

<i>Nro_iteraciones</i>	<i>Casos</i>	<i>Media</i>	<i>Grupos Homogéneos</i>
5	5	0.236437	X
6	5	0.246874	XX
4	5	0.260809	XXX
3	5	0.261272	XXX
2	5	0.278242	XX
1	5	0.293926	X

Método: 95.0 porcentaje LSD

En la tabla 22 se han identificado 3 grupos homogéneos según la alineación de las X's en columnas mediante el procedimiento de diferencia mínima significativa (LSD) de Fisher. No existen diferencias estadísticamente significativas entre aquellos niveles que compartan una misma columna de X's.

**Tabla 23. Contrastes entre medias C221**

<b>Contraste</b>	<b>Sig.</b>	<b>Diferencia</b>	<b>+/- Límites</b>
1 - 2		0.0156841	0.034541
1 - 3		0.032654	0.034541
1 - 4		0.0331173	0.034541
1 - 5	*	0.057489	0.034541
1 - 6	*	0.0470523	0.034541
2 - 3		0.0169699	0.034541
2 - 4		0.0174332	0.034541
2 - 5	*	0.0418049	0.034541
2 - 6		0.0313682	0.034541
3 - 4		0.000463341	0.034541
3 - 5		0.0248351	0.034541
3 - 6		0.0143983	0.034541
4 - 5		0.0243717	0.034541
4 - 6		0.013935	0.034541
5 - 6		-0.0104368	0.034541

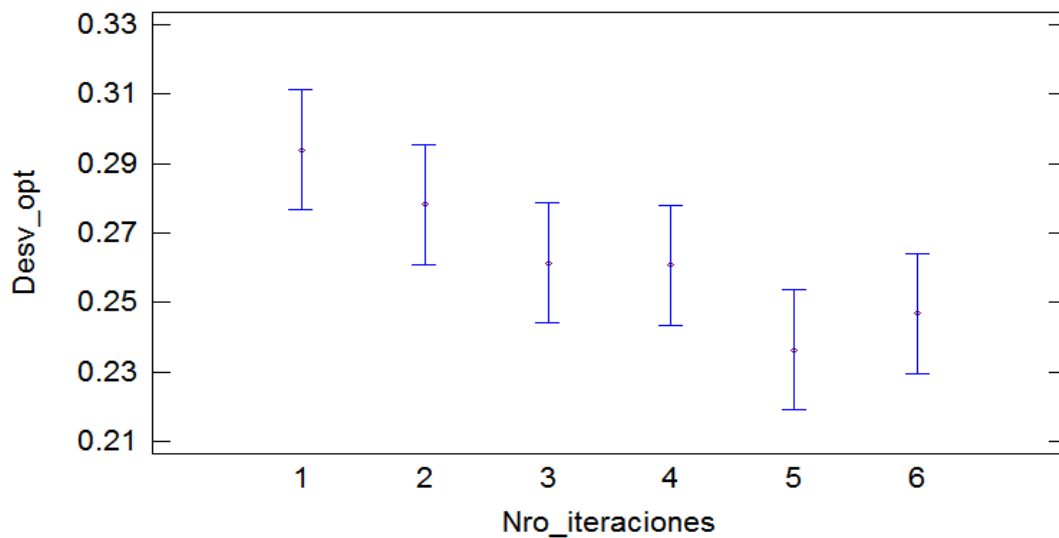
\* indica una diferencia significativa.

La tabla 23 muestra las diferencias estimadas entre cada par de medias. El asterisco que se encuentra al lado de los 3 pares indica que estos pares muestran diferencias estadísticamente significativas con un nivel del 95.0% de confianza.

En la ilustración 26 se observan las medias obtenidas con los diferentes números de iteraciones. Se concluye que con 10 iteraciones en el algoritmo se pueden obtener soluciones con un valor más cercano a la mejor solución encontrada en la literatura, en comparación con 50 y 60 iteraciones, así como que con 50 iteraciones, se obtiene un mejor desempeño comparado con 10 y 20 iteraciones.

**Ilustración 26. Gráfico de medias C221**

Medias y 95.0% de Fisher LSD



**Instancia R241 (400 clientes)**

**Tabla 24. Instancia R241**

Tratamiento (nivel)	Observaciones					Totales	Promedios
	1	2	3	4	5		
1	0.2900	0.3102	0.3201	0.3148	0.3171	1.5521	0.3104
2	0.3430	0.2932	0.3009	0.2978	0.3375	1.5724	0.3145
3	0.2874	0.3802	0.3159	0.2659	0.2981	1.5475	0.3095
4	0.2707	0.3283	0.2943	0.2848	0.3321	1.5102	0.3020
5	0.3375	0.2891	0.2720	0.2533	0.3221	1.4740	0.2948
6	0.2844	0.3016	0.2602	0.3016	0.3148	1.4627	0.2925

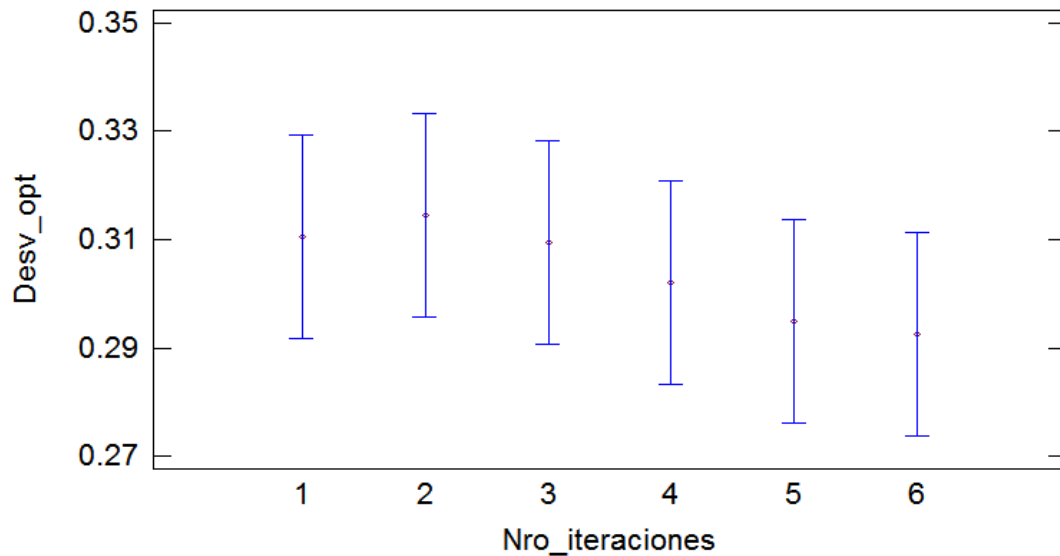
**Tabla 25. ANOVA para Desv\_opt por Nro de iteraciones (R241)**

<i>Fuente</i>	<i>Suma de Cuadrados</i>	<i>Gl</i>	<i>Cuadrado Medio</i>	<i>Razón-F</i>	<i>Valor-P</i>
Entre tratamientos	0.00200909	5	0.000401818	0.48	0.7850
Intra tratamientos	0.0199396	24	0.000830816		
Total (Corr.)	0.0219487	29			

La ANOVA en la tabla 25 descompone la varianza de Desv\_opt en dos componentes: un componente entre-tratamientos y un componente dentro-de-tratamientos. La razón-F, que en este caso es igual a 0.483643, es el cociente entre el estimado entre-grupos y el estimado dentro-de-grupos. Puesto que el valor-P de la razón-F es mayor o igual que 0.05, no existe una diferencia estadísticamente significativa entre la media de Desv\_opt entre un nivel de Nro\_iteraciones y otro, con un nivel del 95.0% de confianza. En la ilustración 27 se observan las medias obtenidas con los diferentes número de iteraciones. Se puede concluir que no existe ningún efecto de los diferentes valores del número de iteraciones en la obtención de un valor objetivo cercana a la mejor solución conocida para la instancia.

**Ilustración 27. Gráfico de medias R241**

Medias y 95.0% de Fisher LSD



**Instancia C241 (400 clientes)**

Tratamiento (nivel)	Observaciones					Totales	Promedios
	1	2	3	4	5		
1	0.31356362	0.369858	0.38219881	0.44432548	0.43156204	1.94150795	0.38830159
2	0.3760284	0.369858	0.34193621	0.3705342	0.38991885	1.84827567	0.36965513
3	0.35238927	0.40727488	0.33080694	0.40087907	0.40611969	1.89746985	0.37949397
4	0.39245464	0.39245464	0.40175251	0.33779443	0.36394117	1.88839739	0.37767948
5	0.39296179	0.36033472	0.33570946	0.34630339	0.38479094	1.8201003	0.36402006
6	0.35968669	0.40513355	0.35839062	0.38687592	0.39659642	1.9066832	0.38133664

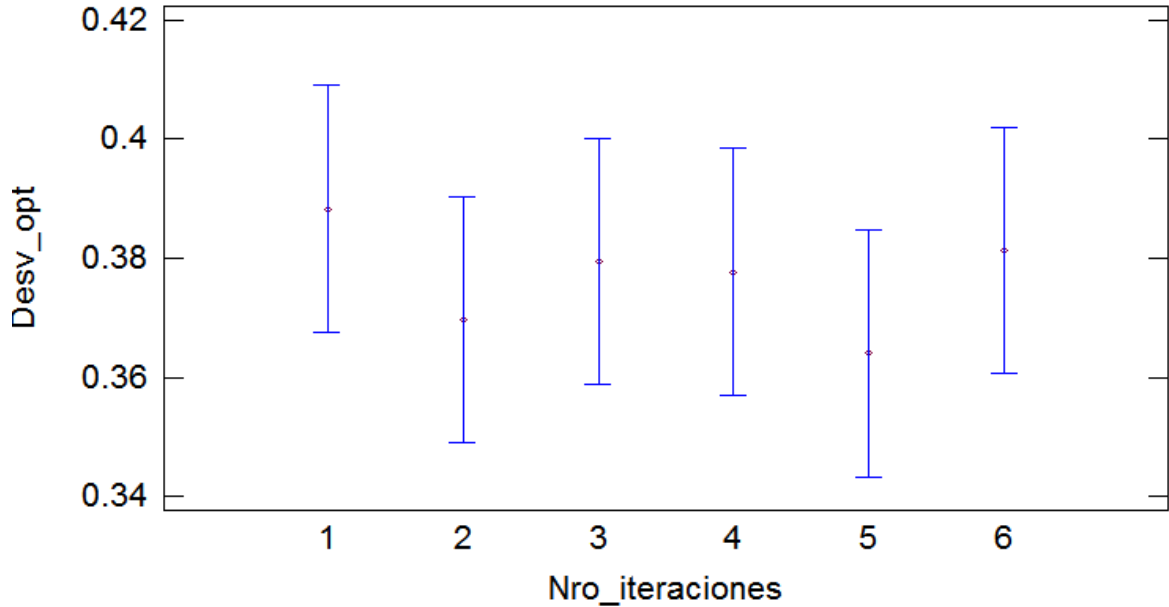
**Tabla 26. Tabla ANOVA para Desv\_opt por Nro\_iteraciones (C241)**

Fuente	Suma de Cuadrados	Gl	Cuadrado Medio	Razón-F	Valor-P
Entre tratamientos	0.00187629	5	0.000375258	0.37	0.8632
Intra tratamientos	0.0242556	24	0.00101065		
Total (Corr.)	0.0261319	29			

La ANOVA de la tabla 26 descompone la varianza de Desv\_opt en dos componentes: un componente entre-tratamientos y un componente dentro-de-tratamientos. La razón-F, que en este caso es igual a 0.371304, es el cociente entre el estimado entre-grupos y el estimado dentro-de-grupos. Puesto que el valor-P de la razón-F es mayor o igual que 0.05, no existe una diferencia estadísticamente significativa entre la media de Desv\_opt entre un nivel de Nro\_iteraciones y otro, con un nivel del 95.0% de confianza. En la ilustración 28 se observan las medias obtenidas con los diferentes números de iteraciones. Se puede concluir que no existe ningún efecto de los diferentes valores del número de iteraciones en la obtención de un valor objetivo cercana a la mejor solución conocida para la instancia.

### Ilustración 28. Gráfico de medias C241

#### Medias y 95.0% de Fisher LSD



## 5.2 ANÁLISIS FACTORIAL DE EFECTOS FIJOS

Tras obtener los resultados de las iteraciones, se tuvo la sospecha de la existencia de un efecto en las soluciones arrojadas por el algoritmo causado por la variación en la capacidad de los vehículos y el número de clientes del problema en sus diferentes instancias. Para verificar esta hipótesis se efectuó un análisis estadístico diseñado para establecer si existe un efecto significativo de los factores *Capacidad de vehículos* y *número de clientes* en las soluciones obtenidas con el algoritmo, medidas por las desviaciones de las dos variables de salida con las mejores soluciones conocidas.

Para este análisis se seleccionaron las instancias representativas de 100,200 y 400 clientes, como se muestra en la tabla 27.

**Tabla 27. Instancias para experimentación (capacidad vehículos)**

<i>Instancias</i>	<i>Capacidad</i>	<i>Clientes</i>
r201	1000	100
c101	200	100
C221	1000	200
R121	200	200
R241	1000	400
R141	200	400

Para todas las instancias se tienen 20 observaciones. Los niveles del factor *Capacidad de vehículos* están dados por las instancias del problema que se probaron. En la tabla 28 se muestran los niveles del factor.

**Tabla 28. Niveles de Capacidad de vehículos**

<i>Niveles</i>		<i>Capacidad</i>
1	bajo	200
2	alto	1000

El modelo de efectos que describe las observaciones del experimento es:

$$y_{ij} = \mu + \tau_i + \beta_j + (\tau\beta)_{ij} + \varepsilon_{ij} \left\{ \begin{array}{l} i = 1000, 200 \\ j = 100, 200, 400 \\ k = 1, \dots, 20 \end{array} \right.$$

Donde:

$y_{ijk}$ : Respuesta observada cuando la capacidad de vehículos está en el nivel  $i$ , hay  $j$  número de clientes en la  $k$ -ésima réplica

$\mu$ : Media *global*

$\tau_i$ : Efecto del  $i$ -ésimo nivel del *número de vehículos*

$\beta_j$ : Efecto del  $j$ -ésimo nivel del *número de clientes*

$(\tau\beta)_{ij}$ : Efecto de la interacción entre el *número de vehículos* y el *número de clientes*

$\mathcal{E}_{ij}$ : Error aleatorio

Se quiere verificar el efecto que tienen los factores *Capacidad de vehículos* y *número de clientes* en la media global de la variable salida *Desv\_opt* y *Desv\_K*, así como de si existe interacción entre ambos que afecte a la solución. Por lo tanto, el conjunto de hipótesis apropiadas son las siguientes:

1) No existe diferencia significativa entre los efectos generados por la *capacidad de vehículos*.

$$H_0 : \tau_1 = \tau_2 = 0$$

$$H_1 : \tau_i \neq 0 \quad \text{para al menos una } i$$

2) No existe diferencia significativa entre los efectos generados por la *el número de clientes*.

$$H_0 : \beta_1 = \beta_2 = \beta_3 = 0$$

$$H_1 : \beta_j \neq 0 \quad \text{para al menos una } j$$

3) No existe interacción entre los efectos de los factores *capacidad de vehículos* y *número de clientes*.

$$H_0 : (\tau\beta)_{ij} = 0$$

$$H_1 : \text{Al menos una } (\tau\beta)_{ij} \neq 0$$

Se ejecutó un análisis de varianza de varios factores para cada una de las variables *Desv opt* y *Desv K*. Mediante varias pruebas y gráficas se determinó qué factores tienen un efecto estadísticamente significativo sobre las medidas de desempeño del algoritmo. También evalúa la significancia de las interacciones entre los factores.

**ANOVA para la desviación en la función de costos *Desv\_opt*.**

**Tabla 29. Análisis de Varianza para Desv opt - Suma de Cuadrados Tipo III**

<i>Fuente</i>	<i>Suma de Cuadrados</i>	<i>Gl</i>	<i>Cuadrado Medio</i>	<i>Razón-F</i>	<i>Valor-P</i>
<b>EFFECTOS PRINCIPALES</b>					
<i>A:Nro Clientes</i>	0.466203	2	0.233102	270.71	0.0000
<i>B:Capacidad vehiculos</i>	0.218219	1	0.218219	253.43	0.0000
<b>INTERACCIONES</b>					
<i>AB</i>	0.327232	2	0.163616	190.02	0.0000
<b>RESIDUOS</b>	0.103328	120	0.000861068		
<b>TOTAL (CORREGIDO)</b>	1.09209	125			

Todas las razones-F se basan en el cuadrado medio del error residual

La ANOVA de la tabla 29 descompone la variabilidad de Desv opt en contribuciones debidas a varios factores. La contribución de cada factor se mide eliminando los efectos de los demás factores. Los valores-P prueban la significancia estadística de cada uno de los factores. Puesto que 3 valores-P son menores que 0.05, estos factores tienen un efecto estadísticamente significativo sobre Desv opt con un 95.0% de nivel de confianza. Esto significa que tanto el número de clientes, la capacidad de los vehículos y la interacción entre ambos tienen efecto sobre la solución del problema con el algoritmo propuesto.

**Pruebas de Rangos Múltiples para DesvOpt por Nro de clientes**

Se aplica un procedimiento de comparación múltiple para determinar cuáles medias son significativamente diferentes de otras. La tabla 30 muestra la identificación de 3 grupos homogéneos según la alineación de las X's en columnas. No existen diferencias estadísticamente significativas entre aquellos niveles que compartan una misma columna de X's.

**Tabla 30. Prueba Rangos múltiples Desv Opt - Nro Clientes**

<i>Nro Clientes</i>	<i>Casos</i>	<i>Media LS</i>	<i>Sigma LS</i>	<i>Grupos Homogéneos</i>
100	39	0.133463	0.00470034	X
200	47	0.250708	0.00432853	X
400	40	0.277424	0.00463969	X

Método Diferencia mínima Significativa (LSD) de Fisher

En la tabla 31 se observan las diferencias estimadas entre cada par de medias. El asterisco que se encuentra al lado de los 3 pares indica que estos pares muestran diferencias estadísticamente significativas con un nivel del 95.0% de confianza. Esto significa que la variación en el *número de clientes* tiene un impacto significativo sobre la variable de respuesta. Con la ayuda gráfica de la ilustración 19, se observa que se obtienen resultados con menor desviación respecto a la mejor solución con 100 clientes, y esta desviación aumenta conforme aumenta el número de clientes.

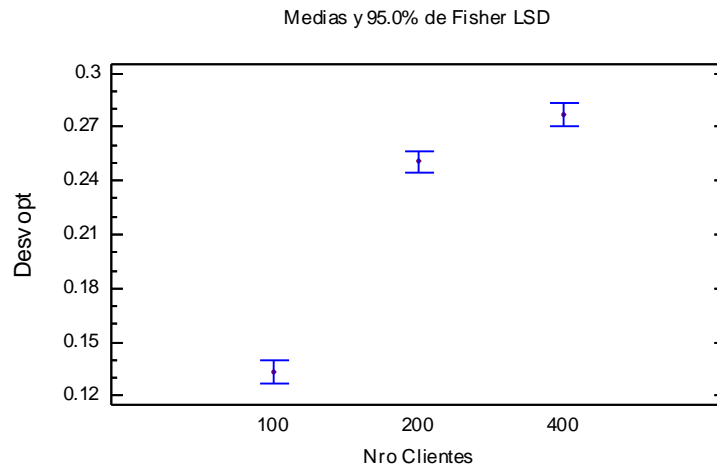
Con esto, podría deducirse que el algoritmo presenta mejores soluciones con problemas pequeños, de 100 clientes, y disminuye su eficiencia a medida que aumenta el tamaño del problema.

**Tabla 31. Contrastes DesvOpt - Nro clientes**

<b>Contraste</b>	<b>Sig.</b>	<b>Diferencia</b>	<b>+/- Límites</b>
100 - 200	*	-0.117245	0.0126514
100 - 400	*	-0.143961	0.0130765
200 - 400	*	-0.0267164	0.0125633

\* indica una diferencia significativa.

**Ilustración 29. Gráfico de Medias - Nro clientes**



## Pruebas de Rangos Múltiples para DesvOpt por Capacidad Vehículos

**Tabla 32. Rangos múltiples para DesvOpt-Capacidad**

Capacidad_vehiculos	Casos	Media LS	Sigma LS	Grupos Homogéneos
1000	59	0.178628	0.00382137	X
200	67	0.262434	0.0036209	X

Método: 95.0 porcentaje LSD

**Tabla 33. Contrastes DesvOpt - Capacidad**

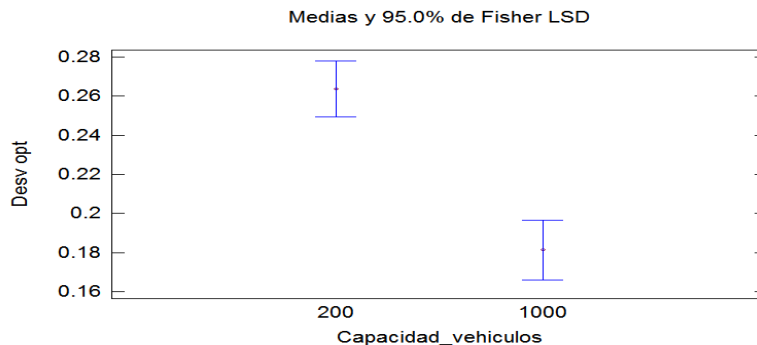
Contraste	Sig.	Diferencia	+/- Límites
200 - 1000	*	0.0838062	0.0104231

\* indica una diferencia significativa.

Para el caso de la *capacidad de los vehículos*, se ha identificado que hay diferencia estadísticamente significativa entre los niveles de la capacidad en la solución del problema con un nivel del 95% de confianza.

En la ilustración 30 se observa el gráfico de medias de *Desv opt* para capacidad de vehículos. Del análisis de varianza y de las pruebas de rangos múltiples se puede concluir que el algoritmo arroja soluciones más cercanas al mejor valor conocido cuando las instancias tienen vehículos con capacidad grande (1000 u.) que cuando tienen capacidad pequeña (200 u.). Esto podría deberse a que con una flota de gran capacidad se requerirán menos vehículos para atender a los clientes, y esto representa un menor costo computacional para el algoritmo.

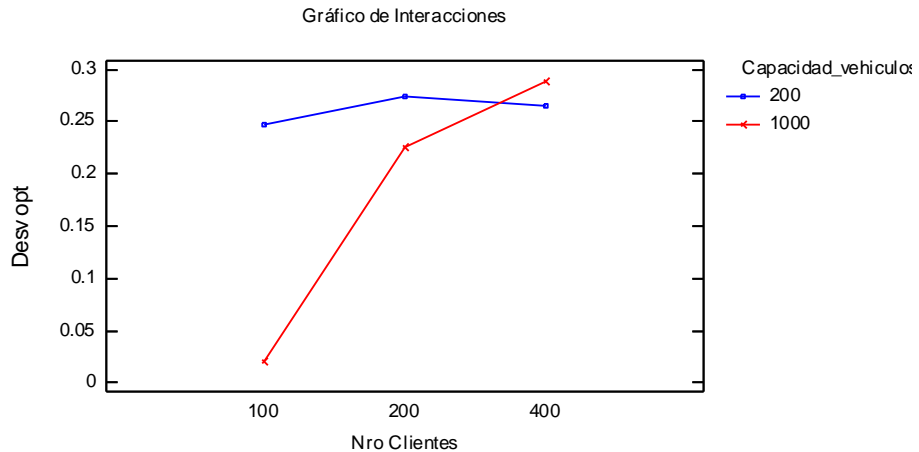
**Ilustración 30. Gráfico de medias Desv Opt - capacidad**



## Interacción de los factores

Como ayuda para interpretar los resultados del experimento, se muestra la gráfica de interacciones en la ilustración 31. El hecho de que las rectas no sean paralelas indica que la interacción es significativa. En general se observa que la desviación de la solución para problemas con vehículos de capacidad de 200 clientes no varía con los diferentes niveles de número de clientes. Sin embargo, para problemas con vehículos de capacidad 1000 se logran encontrar mejores soluciones cuando hay 100 clientes que cuando hay 200 o 400 clientes.

**Ilustración 31. Gráfico de Interacciones para Desv\_opt**



## ANOVA para desviación del número de vehículos, *Desv\_K*.

**Tabla 34. Análisis de Varianza para Desv k - Suma de Cuadrados Tipo III**

<i>Fuente</i>	<i>Suma de Cuadrados</i>	<i>Gl</i>	<i>Cuadrado Medio</i>	<i>Razón-F</i>	<i>Valor-P</i>
<b>EFFECTOS PRINCIPALES</b>					
<b>A:Nro Clientes</b>	0.252979	2	0.12649	38.77	0.0000
<b>B:Capacidad_vehiculos</b>	0.0000303251	1	0.0000303251	0.01	0.9234
<b>INTERACCIONES</b>					
<b>AB</b>	0.211932	2	0.105966	32.48	0.0000
<b>RESIDUOS</b>	0.391475	120	0.00326229		
<b>TOTAL (CORREGIDO)</b>	0.851788	125			

Todas las razones-F se basan en el cuadrado medio del error residual

La ANOVA de la tabla 34 descompone la variabilidad de Desv\_k en contribuciones debidas a varios factores. La contribución de cada factor se mide eliminando los efectos de los demás factores. Los valores-P prueban la significancia estadística de cada uno de los factores. Puesto que 2 valores-P son menores que 0.05, estos factores tienen un efecto estadísticamente significativo sobre Desv\_K con un 95.0% de nivel de confianza. Esto significa que el *número de clientes* y la interacción entre *Nro de clientes* y *capacidad del vehículo* tiene efecto significativo sobre la solución del problema con el algoritmo propuesto. Por su parte, la *capacidad del vehículo* no tiene influencia significativa sobre la solución del algoritmo.

### Pruebas de Rangos Múltiples para DesvK por Nro de clientes

Se aplica un procedimiento de comparación múltiple para determinar cuáles medias son significativamente diferentes de otras. La tabla 35 muestra la identificación de 3 grupos homogéneos según la alineación de las X's en columnas. No existen diferencias estadísticamente significativas entre aquellos niveles que compartan una misma columna de X's.

**Tabla 35. Prueba Rangos Múltiples Desv K - Nro clientes**

<i>Nro Clientes</i>	<i>Casos</i>	<i>Media LS</i>	<i>Sigma LS</i>	<i>Grupos Homogéneos</i>
100	39	0.0390625	0.00914897	X
200	47	0.0924718	0.00842526	X
400	40	0.15217	0.00903091	X

Método: 95.0 porcentaje LSD

En la tabla 36 se observan las diferencias estimadas entre cada par de medias. El asterisco que se encuentra al lado de los 3 pares indica que estos pares muestran diferencias estadísticamente significativas con un nivel del 95.0% de confianza, es decir, que todas las medias de *Desv\_k* con los diferentes tamaños del problema, son diferentes. Esto significa que el número de clientes tiene un impacto significativo sobre la variable de respuesta. Con la ayuda gráfica de la ilustración

32, se observa que se obtienen resultados con menor desviación respecto a la mejor solución con 100 clientes, y esta desviación aumenta conforme aumenta el número de clientes.

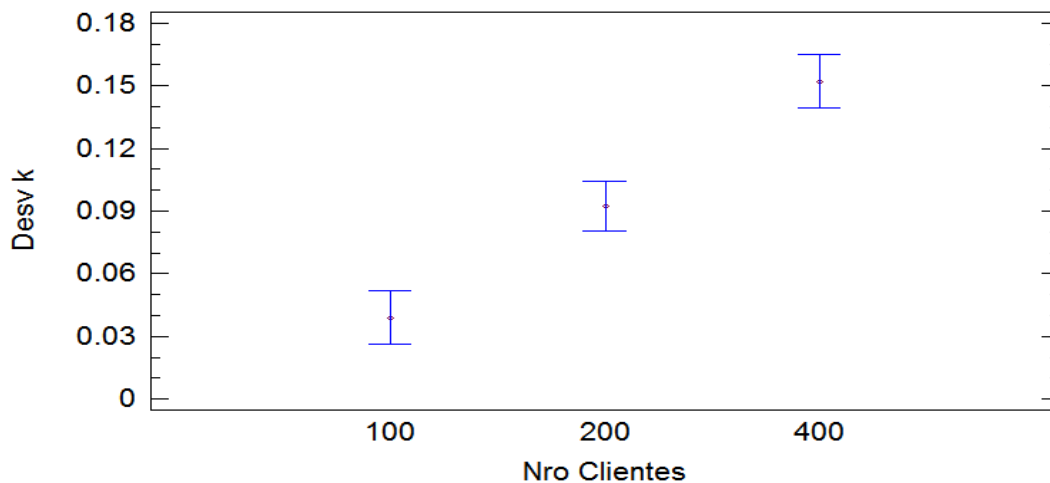
**Tabla 36. Contrastes DesvK - nro clientes**

Contraste	Sig.	Diferencia	+/- Límites
100 - 200	*	-0.0534093	0.0246252
100 - 400	*	-0.113107	0.0254528
200 - 400	*	-0.059698	0.0244538

\* indica una diferencia significativa.

**Ilustración 32. Gráfico de Medias - Nro clientes**

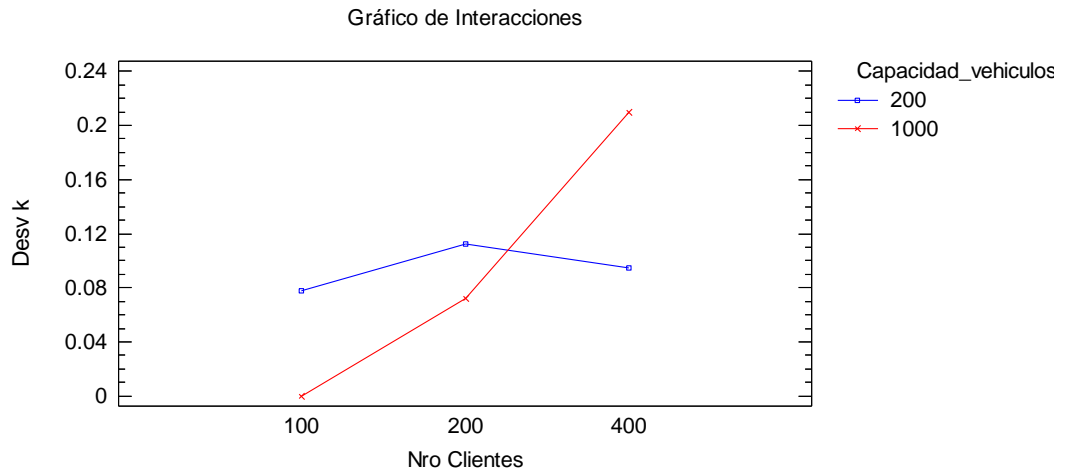
Medias y 95.0% de Fisher LSD



### Interacción de los factores

Como ayuda para interpretar los resultados del experimento, se muestra la gráfica de interacciones en la ilustración 33. El hecho de que las rectas no sean paralelas indica que la interacción es significativa. En general se observa que la desviación de la solución para problemas con vehículos de capacidad de 200, no varía con los diferentes niveles de número de clientes. Sin embargo, para problemas con vehículos de capacidad 1000 se logran encontrar mejores soluciones para el número de vehículos cuando hay 100 clientes, pero desmejoran cuando el tamaño del problema aumenta a 200 y 400 clientes.

**Ilustración 33. Gráfico de interacciones Desv\_K**



## CONCLUSIONES

- Se realizó una recopilación exhaustiva de la literatura concerniente al método metaheurístico PSO (Optimización de enjambre de partículas) aplicado a problemas de programación lineal entera. Se pudo establecer que el método PSO ha sido utilizado satisfactoriamente en una gran variedad de problemas de optimización combinatoria y, específicamente, aplicado a diferentes problemas presentes en los sistemas logísticos de diferentes sectores económicos. Ha sido utilizado para resolver muchas variaciones del Problema de Ruteo de vehículos y en problemas de localización, produciendo buenas soluciones.
- Se desarrolló un algoritmo aplicado a la resolución del problema VRPSPD (Problema de ruteo de vehículos con entrega y recolección simultánea) basado en la metaheurística PSO que arroja soluciones factibles para problemas de hasta 400 clientes. Los resultados son cercanos a los valores de referencia y fueron obtenidos en tiempos de cómputo razonables.
- Los códigos de programación del algoritmo desarrollado, implementados en MatLab, se dejan a disposición del lector, como herramienta para estudiar posibles variaciones en la programación, codificación, implementación o aplicación en trabajos futuros.
- A partir de los experimentos numéricos realizados al algoritmo propuesto, se puede concluir que:
- En la mayoría de las instancias en las cuales se hizo el análisis estadístico, se encontró que la utilización de diferentes números de iteraciones definidos como criterio de parada influyen en la solución que arroja el algoritmo. En promedio,

se obtuvo que con 50 iteraciones, el programa arroja soluciones con menor desviación respecto a la mejor solución conocida del problema.

- Del análisis factorial de los efectos fijos, se pudo comprobar que el tamaño del problema influye significativamente en la eficiencia del algoritmo para encontrar una buena solución. Para problemas de gran número de clientes, el algoritmo tuvo una mayor dificultad en encontrar soluciones que tuvieran menor desviación en su costo, respecto a las mejores soluciones conocidas.
- También, se pudo concluir que para todos los tamaños del problema, la capacidad de la flota de vehículos disponible, es una variable que tiene un impacto significativo en la obtención de buenas soluciones. Dado que la capacidad de los vehículos es una restricción del problema, la complejidad aumenta cuando ésta se hace menor.
- La interacción entre tamaño del problema y capacidad de vehículos implica una variación significativa en la obtención de resultados con el algoritmo propuesto. Se puede concluir que el algoritmo obtiene mejores soluciones con problemas pequeños de hasta 200 clientes con una flota de vehículos de mayor capacidad. A medida que el problema aumenta de tamaño y se reduce el umbral de la restricción de capacidad, el algoritmo se hace menos eficiente.
- Para un número grande de iteraciones usadas como criterio de parada, el algoritmo desarrollado no muestra convergencia para obtener soluciones en algunas de las instancias.
- Se adquirió valiosa habilidad en programación en lenguaje MatLab, que permite desarrollar investigaciones futuras en otros campos de optimización y de la ingeniería industrial en general. Así mismo, esta experiencia da la

capacidad de utilizar otros lenguajes de programación para desarrollar futuras investigaciones en este campo.

- Se entrega un artículo publicable en el que se resumen los resultados obtenidos en este trabajo de investigación.

## RECOMENDACIONES

- Para futuras investigaciones enfocadas a encontrar métodos de solución para el VRPSPD, se recomienda tener en cuenta el valor de costo fijo relacionado con la utilización de los vehículos ya que, aunque para las instancias estudiadas en este proyecto se fijó este valor en cero, previos estudios en torno a otras variables del VRP han mostrado que éste tiene un alto impacto en la función de costos para aplicaciones prácticas del problema. El método propuesto puede manejar fuertemente esta situación puesto que el modelo matemático y la programación del algoritmo consideran el costo fijo en la función objetivo.
- Es importante considerar otros procedimientos para establecer el número inicial de vehículos necesarios para que la rutina usada para calcular el número apropiado de vehículos sea más eficiente y se encuentren mejores soluciones al problema.
- De acuerdo a la formulación matemática del VRPSPD, es conveniente considerar los tiempos de servicio y las restricciones que estos representan, de manera que las soluciones que se obtengan con métodos Metaheurísticos sean más próximas a problemas reales.
- Algoritmos adicionales dentro del método de decodificación que mejoren las rutinas de construcción de rutas pueden tener un impacto significativo en la búsqueda de nuevos algoritmos para resolver el VRPSPD. Dado que las rutas son fluctuantes, no se construyen progresivamente una a una y podrían tener un número diferente de clientes en cada una de ellas, es posible aplicar heurísticas de intercambio entre-rutas y no únicamente intra-rutas para mejorar las secuencias emergentes.

- El desarrollo de aplicaciones prácticas que puedan apoyar los procesos logísticos de la industria requiere partir de herramientas computacionales como la propuesta en el presente trabajo. Por esta razón es importante divulgar y aprovechar este tipo de investigaciones teóricas.
- Se recomienda la realización de un análisis de sensibilidad para determinar otras combinaciones de parámetros del PSO, con el fin de encontrar mejores soluciones al problema planteado.

## REFERENCIAS BIBLIOGRÁFICAS

Angelelli E., Mansini R., The Vehicle Routing Problem with time Windows and simultaneous pick-up and delivery. Lecture Notes in Economics and Mathematical Systems, A. Klose et al (eds), 249 – 267, Springer, 2002.

Applegate, D. L.; Bixty, R.M.; Chvátal, V; Cook, W.J. The Travelling Salesman Problem, 2006. ISBN: 0-691-12993-2

Beasley JE. Route first-cluster second methods for vehicle routing. Omega 1983; 11:403 – 8.

Bianchessi, N., Righini, G. (2007). Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery. Computers & Operations Research 34 (2), 578 – 594.

BLUM, C., ROLI, A. Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. ACM Computing Surveys, Vol. 35, No. 3, September 2003, pp. 268–308.

Bräysy, O., Gendreau, M. Vehicle Routing Problem with Time Windows, Parti I: Route Construction and Local Search Algorithms. Chemical Engineering PasAmerican Collaboration CEPAC. Tomado de internet el 2 de Julio de 2013 [<http://cepac.cheme.cmu.edu/pasi2011/library/cerda/braysy-gendreau-vrp-review.pdf>]

C.W. Reynolds. "Flocks, herds and schools: a distributed behavioral model". Computer Graphics. 1987, 21(4) P. 25-34

Chen AL, Yang GK, Wu ZM. Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem. Journal of Zhejiang University Science A 2006;7(4):607--14.

CORDEAU, J.-F., GENDREAU, M., LAPORTE, G., POTVIN, J.Y. and SEMET, F. "A guide to vehicle routing heuristics," Journal of the Operational Research Society, vol. 53, no. 5, pp. 512–522, May 2002.

Crispim, J., & Brandao, J. (2005). Metaheuristics applied to mixed and simultaneous extensions of vehicle routing problems with backhauls. Journal of the Operational Research Society, 56, 1296 – 1302.

Dantzig, G. B., RAMSER, J. H. The Truck Dispatching Problem. Management Science, Vol 6. No 1, Oct. 1959, pp. 80 – 91

Dell'Amico M, Girghini G, Salani M. A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection. Transport Science 2006;40(2): 235-47

Dethloff, J. (2001). Vehicle routing and reverse logistics: The vehicle routing problem with simultaneous delivery and pick-up. OR Spektrum, 23, 79–96.

DETHLOFF, J. Vehicle Routing and reverse logistics: the vehicle routing problems with simultaneous delivery and pick-up. 2001. Or Spektrum, Vol 23, p 79-96

DETHLOFF, J. Vehicle Routing and reverse logistics: the vehicle routing problems with simultaneous delivery and pick-up. 2001. Or Spektrum, Vol 23, p 79-96

Engelbrecht, A. Computational intelligence An Introduction. Editorial John Wiley. 2° Edición. Chippenham, Great Britain. 2007 - ISBN: 978-0-470-03561-0 (HB)

FLEISCHMANN, M. Quantitative models for Reverse Logistics. Thesis to obtain the degree of Doctor from the Erasmus University Rotterdam, Rotterdam. July 2000

Gajpal, Y., & Abad, P. (2012). Saving-based algorithms for the vehicle routing problem with simultaneous delivery and pickup. *Computers and Operations Research Society*, 61(1), 1498-1509.

Gajpal, Y., Abad, P. (2009). An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup. *Computers & Operations Research*. Doi:10.1016/j.cor.2009.02.017.

GARCÉS RAMIREZ, Carlos Duván. Modelo de entregas directas para la reducción de costos logísticos de distribución en una empresa de consumo masivo. Aplicación en una empresa piloto de Caldas. Trabajo de grado para Optar al título de Magíster en Administración de Negocios, Director William Ariel Sarache Castro. Universidad Nacional de Colombia, Facultad de Administración, Manizales. 2010.

Garey, M.R., Johnson, D.S., 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York.

Gehring, H., & Homberger, J. A parallel hybrid evolutionary metaheuristics for the vehicle routing problem with time windows. In K. Miettinen, M. Mäkelä, & J. Toivanen (Ed.). *Proceedings for EUROGEN99 (Vol. A2(s), pp. 57 – 47 )*. Berlin : Springer, 1999.

Gillet BE, Miller LR. A heuristic algorithm for the vehicle dispatch problem. *Operations Research* 1974;22:340 – 9.

Goksal, F.P., Karaoglan, I. , Altiparmak, F. A Hybrid Discrete Particle Swarm Optimization for Vehicle Routing Problem with Simultaneous Pickup and Delivery. Jan. 2012, Computers & Industrial Engineering, 2012.

Hebert W. Davis y William H. Drumm. Logistics Costs and Service Database. Annual Conference Proceedings .San Francisco, CA, 2002: Council of Logistics Management.

Heppner, F., Grenander, U. "A stochastic nonlinear model for coordinated bird flocks. Editorial In S. Krasner. "The Ubiquity of Chaos". 1990. Publicaciones AAAS, Washington D.C.

HILLIER. F.S., LIEBERMAN. G.J. Introducción a la investigación de operaciones. Novena Edición. Editorial Mc. Graw-Hill. México, 2010. ISBN: 0073376299

KACHITVICHYANUKUL, V. The jin Ai, Voratas Kachitvichyanukul, A particle Swarm Optimization for the vehicle routing problem with simultaneous pickup and delivery. Computers & Operations Research, vol 36, 2009, p. 1693 -1702.

Lin S. (1965). Computer solutions for the travelling salesman problem. Bell System Technology Journal; 44:2245-69.

MAREK L. A note on robustness tolerances for combinatorial optimization problems. Information Processing Letters. Vol 110. 2010. p.725-729.

MARTÍ, R. Procedimientos Metaheurísticos en Optimización combinatoria. Departamento de Estadística e Investigación Operativa. Facultad de Matemáticas. Universidad de Valencia. 2003.

MIN, H. The multiple vehicle routing problem with simultaneous delivery and pickup points. 1989. Transportation Research A. Vol 23, p. 337-386.

MINISTERIO DE TRANSPORTE. Transporte en cifras. Edición 2011. Bogotá: MINISTERIO DE TRANSPORTE, 2012. 43 p.

Moghaddam, B. F., Ruiz, R., Sadjadi, S. J. Vehicle Routing Problem with uncertain demands: An advanced particle swarm algorithm. Computers & Industrial Engineering 2012; Vol 62. P: 306-317.

Nagy G, Salhi S. Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. European Journal of Operational Research 2005; 162(1):126--41.

Particle Trajectories. Information Sciences, 176(8):937–971, 2006.

PAU COS, Jordi. y IAVASCUES Y GASCA, Ricardo. Manual de logística integral [online]. Ediciones Díaz de Santos. Madrid, España: Google Books, 2013 [Citado el 30 de Marzo de 2013]. Disponible en: <http://books.google.com.co/Books>.

Pongchairerks, P. , Kachitvichyanukul, V. A non-homogeneous particle swarm optimization with multiple social structures. Proceedings of the 2005 International Conference on Simulation and Modeling

Salhi S, Nagy G. A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. Journal of the Operational Research Society 1999;50(10):1034--42.

SÁNCHEZ, M. G. Las matemáticas del siglo XX: una mirada en 101 artículos. Págs 115-120. Universidad de la Laguna, Sociedad Canaria Isaac Newton de profesores de matemáticas: Nivola. 2000.

Solomon, M. M. Algorithms for the vehicles routing and scheduling problems with time window constraints. *Operations Research* 1987; vol 35, 254 – 265

SUNGUR, Ilgaz. The Robust Vehicle Routing Problem. Ann Arbor, MI – USA, 2007, 24 h. PhD Thesis (Doctor of Philosophy, Industrial and systems engineering). University of Southern California, Faculty of the Graduate School.

T. WEISE., M. ZAPF, R. CHIONG, A. NEBRO. Nature-Inspired Algorithms for Optimization. Series: Studies in Computational Intelligence, Vol 193. Springer, 2009.

Tang FA, Galvao RD. A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Computers and Operations Research* 2006;33(3):595--619.

Tang FA, Galvão RD. Vehicle routing problems with simultaneous pick-up and delivery service. *Journal of the Operational Research Society of India (OPSERACH)* 2002; 39; 19 – 33.

TOTH, P., VIGO “An exact algorithm for the capacitated shortest spanning arborescence,” *Annals of Operations Research*, vol. 61, Dec. 1995, pp. 121-141.

TOTH, P., VIGO “Exact solutions of the vehicle routing problem”. *Fleet Management and Logistics*. 1998. Kluwer: Boston, p 1-31.

TOTH, P., VIGO, D. THE VEHICLE ROUTING PROBLEM. Vol 9 of Monographs on Discrete Maths and Application Series. SIAM, 2002. ISBN: 978 – 089 – 87 – 1579 - 8

TREVISAN, L. “Lecture Notes on Computational Complexity”. Computer Science Division. Berkeley, University of California. 2004.

Van Den Bergh, F. and Engelbrecht, A.P. A Study of Particle Swarm Optimization  
W. Chu, X. Gao, S. Sorooshian. 'Handling boundary constraints for particle swarm optimization in high-dimensional search space. Information Sciences. 2011, Vol. 181, p. 4569-4581

Wassan, N.A., Wassan, A. H. & Nagy, G. (2007). A reactive tabu search algorithm for the vehicle routing problem with simultaneous pickups and deliveries. Journal of combinatorial Optimization 15. 368 – 386.

Zachariadis, E., Kiranoudis, C. A local search metaheuristic algorithm for the vehicle routing problem with simultaneous pick-ups and deliveries. Expert Systems with Applications 2011; vol 38 (3): p. 2717-2726.

Zachariadis, E.E., & Kiranoudis, C.T. A local Search metaheuristic algorithm for the vehicle routing problema with simultaneous pick-ups and deliveries. Expert Systems with Applications 2009; vol. 38 (3)), pp. 2717-2726.

Zachariadis, E.E., Tarantilis, C.D., & Kiranoudis, C.T. A Hybrid metaheuristic solution approach for the Vehicle Routing Problem with simultaneous delivery and pick-up service. Expert Systems with Applications 2009; vol. 36 (2), pp. 1070 – 1081.

Zachariadis, E.E., Tarantilis, C.D., & Kiranoudis, C.T. An adaptive memory methodology for the vehicle routing problem with simultaneous pick-ups and deliveries. European Journal of Operational Research 009; doi: 10.1016/j.ejor.2009.05.015.

## ANEXOS

### Anexo A. Código de programación

En el anexo presentado a continuación se encuentra detallado el Toolbox de Matlab elaborado para la construcción del algoritmo GLNPSO de resolución del problema VRPSPD, los métodos heurísticos de mejoramiento y el método de decodificación propuesto.

#### **FUNCIÓN ‘distancias’**

A partir de las coordenadas de los clientes disponibles para cada instancia, construye la matriz de distancias euclidianas entre clientes y depósito.

#### **Entradas:**

- depCor: Coordenada del depósito
- clieCor: Coordenadas de clientes
- vehVector: Vector que contiene la información de la flota de vehículos.

#### **Salidas:**

- distMat: Matriz de distancias euclidianas entre clientes y depósito.
- tieMat: Matriz de tiempo de viaje entre clientes y depósito.

```
function [distMat,tieMat] = distancias(depCor,clieCor,vehVector)
% Busca coordenadas de clientes para construir Matriz de distancias
s=size(clieCor,1); % s no contiene el depósito

% Se agrega la coordenada del depósito en la última posición
for i=1:2
    clieCor(s+1,i)=depCor(i);%Agrega al depósito en la última posición de
clieCor
end

M = size(clieCor,1); %Numero de nodos(incluye deposito)

distMat = zeros(M); % Crea matriz de distancias

% Calcula la distancia entre dos clientes, uno por uno.
% La distancia calculada es distancia euclideana
% Incluye al depósito en la última posición
```

```

for i=1:M
    for j=i+1:M
        distMat(i,j)=sqrt( ( clieCor (j,1) - clieCor (i,1) ) .^2)...
            +( (clieCor (j,2) - clieCor (i,2) ).^2 ) );
        distMat(j,i) = distMat (i,j);
        %Calcula Distancias del taxista. Matriz simétrica
    end
end
end
%N=M-1;%numero de clientes
velocidadVeh=vehVector(6);
tieMat=distMat./velocidadVeh; %Calcular tiempos de viajes entre clientes.

```

## **FUNCIÓN ‘vehículos’**

Calcula el número apropiado de vehículos para construir las rutas del problema.

### **Entradas:**

- N: Número de clientes del problema.
- numVeh: Número de vehículos disponibles inicialmente.
- rangoMin: Límite inferior del espacio de búsqueda.
- rangoMax: Límite superior del espacio de búsqueda.
- clieDem: Matriz de demandas de entrega y recogida de cada cliente.
- distMat: Matriz de distancias entre clientes y depósito.
- tieMat: Matriz de tiempos de viaje entre clientes y depósito.
- vehVector: Vector que contiene la información de la flota de vehículos, tales como capacidad, límite de duración del servicio, costo fijo asociado a la utilización de cada vehículo y costo variable por unidad de distancia.
- clieCor: Coordenadas entre clientes y depósito

### **Salidas:**

- numVeh: Número apropiado de vehículos para la construcción de las rutas.

```

function [numVeh] =
vehiculos(N,numVeh,rangoMin,rangoMax,clieDem,distMat,tieMat,vehVector,clieCor)

%% Generar una partícula
posicion = rangoMin + ( rangoMax-rangoMin ).* rand ( 1 , N+2*numVeh );
v = numVeh;

%% Penalización al costo por clientes no servidos
penalty = 1000;
%% Decodificar la solución de la partícula

[priorClientes] = priorClienteINICIALIZAR(N,posicion);
[priorVeh] = prioriVehINICIALIZAR(posicion,N,clieCor);
[rutaEnjambre] = celdaenjambreINICIALIZAR(N,numVeh);
[rutaEnjambre]=constructRouteINICIALIZAR(N,clieDem,
priorClientes,priorVeh,distMat,tieMat,vehVector,rutaEnjambre,numVeh);
[fitness] = performanceINICIALIZAR(rutaEnjambre,vehVector,numVeh);
z = 0;
%% Mientras haya minimización
while (z==0)

%% Calcular numero de clientes servidos por cada vehiculo
z = 1;
clientes = zeros;

for i = 1:numVeh
    clientes(i) = length(rutaEnjambre{i,1}) - 1 ;
end

%% Eliminar las dimensiones correspondientes al vehiculo con menor numero
de
%%clientes. Se reduce el tamaño de la partícula en dos dimensiones.

[~,I] = min(clientes);
posicion(N+2*I-1:N+2*I) = [];
v = v - 1;
%% DecodificaR la partícula con la partícula actualizada (un vehículo
menos)
[priorClientes] = priorClienteINICIALIZAR(N,posicion);
[priorVeh,numVeh] = prioriVehINICIALIZAR(posicion,N,clieCor);
clear rutaEnjambre
[rutaEnjambre] = celdaenjambreINICIALIZAR(N,numVeh);
[rutaEnjambre,noAsignados]=constructRouteINICIALIZAR(N,clieDem,
priorClientes,priorVeh,distMat,tieMat,vehVector,rutaEnjambre,numVeh);
[fitness_primo] = performanceINICIALIZAR(rutaEnjambre,vehVector,numVeh);
fitness_primo = fitness_primo + penalty * noAsignados;

% Si hay disminución del costo
if (fitness_primo <= fitness)

    fitness = fitness_primo;
    z=0;

```

```

else
    v = v + 1;
end
end
%%
numVeh = v;

```

## **FUNCIÓN ‘inicializar’**

Inicializa el enjambre de partículas del algoritmo. Se define la velocidad inicial, la posición, los comportamientos personal y global, y los límites del espacio de búsqueda.

### **Entradas:**

- N: Número de clientes del problema.
- numPar: Número de partículas del enjambre.
- numVeh: Número de vehículos disponibles.
- vecinos: Número de vecinos correspondientes al comportamiento social ‘Local best’ del enjambre.
- rangoMin: Límite inferior del espacio de búsqueda.
- rangoMax: Límite superior del espacio de búsqueda.

### **Salidas:**

- curSwarm: Matriz que contiene la posición actual del enjambre.
- velocidad: Vector de velocidad del enjambre.
- perBest: Matriz que contiene la mejor posición personal de cada partícula del enjambre.

```

function [curSwarm, velocidad, perBest, globalBest, localBest,
solucion, rangoMin, rangoMax]=inicializar(N, numPar, numVeh, vecinos, rangoMin,
rangoMax)
%Preallocating
perBest = {zeros};

%% Inicializar particulas del enjambre
curSwarm = rangoMin + ( rangoMax-rangoMin ).* rand ( numPar , N+2*numVeh
);
velocidad = zeros( numPar , N + 2*numVeh );

for i=1:numPar
    %% Inicializa Personal Best

```

```

    perBest{i, :, 1} = curSwarm(i, :);
    perBest{i, 1, 2} = Inf; %Se define como infinito inicialmente
end

%% Inicializa Global Best
globalBest{1, 1} = zeros(1, N+2*numVeh);
globalBest{1, 2} = Inf; %Se define como infinito inicialmente

% La solución del problema que arroja globalBest será la solución
% definitiva del algoritmo por lo tanto, se deberá guardar en
"solucion".

    %Preallocating
solucion = {zeros};

for i = 1 : numVeh
    solucion{i, 1} = [];
    solucion{i, 2} = [];
    solucion{i, 3} = [];
end

%% Inicializa Local Best
localBest = {zeros};
for i = 1 : numPar/vecinos
    localBest{i, 1} = zeros(1, N+2*numVeh);
    localBest{i, 2} = Inf; %Se define como infinito inicialmente
end

```

## MÉTODO DE DECODIFICACIÓN

Este conjunto de funciones se utiliza tanto para la totalidad del enjambre durante la ejecución del algoritmo, como para la partícula que se encarga del cálculo apropiado de vehículos para cada instancia del problema.

### FUNCIÓN 'priorCliente'

Construye la lista de prioridades de clientes del problema.

#### Entradas:

- N: Número de clientes del problema.
- numPar: Número de partículas del enjambre.
- curSwarm: Matriz que contiene la posición actual del enjambre.

#### Salidas:

- priorClientes: Lista de prioridad de clientes a ser atendidos en el problema.

```
function [priorClientes] = priorCliente(N,numPar,curSwarm)
priorClientes = [zeros];
swarmEv=curSwarm; %matriz de evaluación para buscar el menor
for i=1:numPar
    %clientes(i,:)=(1:N); % Conjunto S de algoritmo 2
    for j=1:N
        [maxPrior,indCliente]=min(swarmEv(i,1:N)); %busca el indice del
menor valor de las posiciones de S
        priorClientes(i,j)=indCliente; %Agrega el indice encontrado a U
        swarmEv(i,indCliente)=Inf; %Borra el cliente encontrado de S
    end
end
```

### **FUNCIÓN 'prioriVeh'**

Calcula la matriz de prioridad de vehículos de acuerdo a la distancia entre los clientes y los puntos de referencia de cada vehículo.

#### **Entradas:**

- N: Número de clientes del problema
- numPar: Número de partículas del enjambre
- curSwarm: Matriz que almacena la posición actual del enjambre en todas las dimensiones del problema.

#### **Salidas:**

- priorVeh: Matriz de prioridad de vehículos para cada cliente del problema.

```
function [priorVeh] = prioriVeh(vehVector,numPar,curSwarm,N,clieCor)
% Preallocating
xref = [zeros];
yref = [zeros];
eucDist = [zeros];

priorVeh = [zeros];
numVeh=vehVector(1);
for i=1:numPar %para cada particula
    for j=1:numVeh %para cada vehiculo
        xref(i,j)=curSwarm(i,N+(2*j)-1); %calcule puntos de ref en X y Y
        yref(i,j)=curSwarm(i,N+(2*j));
    end
end
for j=1:numVeh
```

```

    for k = 1:N %Para cada cliente calcule distancia euclidiana a los ref
        eucDist(i,j,k)= sqrt((clieCor(k,1)-xref(i,j)).^2+(clieCor(k,2)-
yref(i,j)).^2); %calculo de dis euclidiana

    end
    end
    euclidianas(i,[:, :]) = eucDist(i,[:, :]); %matriz de evaluación para
buscar el menor
    for k=1:N
    for j=1:numVeh
        [maxPriorVeh,indVehiculo]=min(euclidianas(i,1:numVeh,k)); %busca el
indice del menor valor de las posiciones de S. vehiculos
        priorVeh(k,j,i)=indVehiculo; %Agrega el indice encontrado a W(matriz
prior Veh)
        euclidianas(i,indVehiculo,k)=Inf; %Borra el vehiculo encontrado de la
matriz de evaluación
    end
    end
end
end

```

### **FUNCIÓN 'celdaenjambre'**

Crea el arreglo de celdas que contiene la información de la solución que genera el enjambre.

#### **Entradas:**

- N: Número de clientes del problema.
- numPar: Número de partículas del enjambre.
- vehVector: Vector que contiene información sobre la flota de vehículos.

#### **Salidas:**

- rutaEnjambre: Arreglo de celdas que contiene la información de las rutas construidas para el problema.

```

function [rutaEnjambre] = celdaenjambre(N,numPar,vehVector)
%% Preallocating
rutaEnjambre = {zeros};

numVeh=vehVector(1);
for i=1:numPar
    %--> En la primera capa del arreglo se encuentra la secuencia de cada
vehículo.
    for j=1:numVeh
        for k=1 %Almacena secuencias de rutas en el arreglo celda.
            rutaEnjambre{i,j,k}=N+1;
        end
    end
end

```

```

end
for k=2 %Almacena costos de las secuencias
    rutaEnjambre{i,j,k}=(0);
end
for k=3 %Almacena Carga de cada arco de la secuencia
    rutaEnjambre{i,j,k}= [0];
end
for k=4 %Almacena tiempos de la secuencia
    rutaEnjambre{i,j,k}= (0);
end
end
end

```

### **FUNCIÓN 'constructRoute'**

Construye las rutas para la flota de vehículos del problema correspondiente y la información asociada a cada una de ellas.

#### **Entradas:**

- N: Número de clientes del problema.
- numPar: Número de partículas del enjambre.
- clieDem: Matriz de demandas de entrega y recogida de cada cliente del problema.
- priorClientes: Lista de prioridad de clientes por atender.
- priorVeh: Matriz de prioridad de vehículos asignada a cada cliente de acuerdo a los puntos de referencia.
- distMat: Matriz de distancias euclidianas entre clientes y depósito.
- vehVector: Vector que contiene toda la información relacionada con la flota de vehículos.
- rutaEnjambre: Arreglo de celdas que contiene la información asociada a las rutas construidas por el algoritmo.

#### **Salidas:**

- rutaEnjambre: Arreglo de celdas que contiene la información de las rutas construidas para el problema.

```

function [rutaEnjambre]=constructRoute(N,numPar,clieDem,
priorClientes,priorVeh,distMat,tieMat,vehVector,rutaEnjambre)

```

```

%Calcula número de vehículos
numVeh=vehVector(1);

%%      2. Se construyen las rutas

for i=1:numPar %Construcción de rutas para cada partícula.

    for k = 1:N; %Recorre matriz de prioridad de clientes.

        for prioridad = 1: numVeh %Asigna el candidato a su vehículo en
orden de prioridad

            % Selecciona al candidato de la lista de prioridad
candidato = priorClientes(i,k);

            % Asigna al candidato a la ruta del vehículo que está en su
% prioridad
vehiculoasignado = priorVeh(candidato,prioridad,i);

            % Selecciona la secuencia del vehículo respectivo y el costo
secuenciactual=rutaEnjambre{i,vehiculoasignado,1};
costoactual=rutaEnjambre{i,vehiculoasignado,2};

            cont=length(secuenciactual); %contador de posiciones en la
secuencia

            nuevasecuencia = secuenciactual;%Crea una secuencia para
evaluación de restricciones
            nuevasecuencia( cont + 1 ) = candidato;%Ingresa al candidato a
la siguiente posición disponible de la secuencia
            nuevocosto = costoactual +
distMat(candidato,nuvasecuencia(cont))...
            + distMat(candidato,nuvasecuencia(1))-
distMat(nuvasecuencia(cont),nuvasecuencia(1)); %Calcula el costo de la
nueva secuencia con el candidato seleccionado.
            % Se actualiza el costo de inicio y fin de la ruta, de acuerdo
al
            % cambio de secuencia tras agregar a candidato.

            % Evaluar el menor costo de inserción en nueva secuencia
secuenciaPrueba = nuevasecuencia;
indice = cont;

            if cont>1
                %La heurística de inserción se hace cuando haya al menos un
%cliente asignado a la ruta.

                %% ---> Cheapest Insertion Heuristic
                for j = 2:cont          %Realiza cont-1 intercambios posibles.

                    %Hace el intercambio de nodos

```

```

a = secuenciaPrueba(indice);
secuenciaPrueba(indice) = secuenciaPrueba(indice+1);
secuenciaPrueba(indice+1) = a;
costoPrueba = 0; % Comienza en cero para empezar a
evaluar
    %Calcula el costo de la secuencia de prueba
    for h=1:length(secuenciaPrueba)-1 % Recorre la
secuencia de prueba
        costoPrueba = costoPrueba +
distMat(secuenciaPrueba(h),secuenciaPrueba(h+1));
        end
        costoPrueba = costoPrueba +
distMat(secuenciaPrueba(length(secuenciaPrueba)),N+1); %Agrega el costo
de volver al depósito.

        if (costoPrueba < nuevocosto) % Si el costo de
prueba es menor al anterior,
            nuevasecuencia = secuenciaPrueba; %Se actualiza
como mejor inserción a la secuencia, y,
            nuevocosto = costoPrueba; % se actualiza su
costo
        end

        indice=indice-1; % Continúa evaluando hasta analizar
todas las posibles inserciones
    end

end
% Al terminar la Heurística (Cheapest insertion heuristic)
se
    % tendrá el candidato insertado en la mejor posición de la
% secuencia factible.

%% Evalúa la factibilidad de la mejor secuencia encontrada.
capacidad = vehVector (2); % Q = capacidad del vehículo

% vector con Información sobre la carga de los arcos de la secuencia.
vectorCarga = rutaEnjambre{i,vehiculoasignado,3};

% El vector tiene las cargas de cada arco de la ruta.
% 1. Carga del vehículo al salir del depósito
%cargaDeposito = vectorCarga(1);

% 2. Carga del vehículo después del último arco asignado.
%ultimaCarga = vectorCarga(2);

%Calcular carga de cada arco y evaluar su factibilidad.
vectorCarga(1) = vectorCarga(1) + clieDem(candidato,1);
for j = 1 : length (nuevasecuencia) - 1

```

```

        vectorCarga (j + 1) = vectorCarga(j) -
        clieDem(nuevasecuencia(j+1),1) ...
            + clieDem (nuevasecuencia(j+1),2);

    end

% Si la carga en ningún arco excede la capacidad
if all(vectorCarga < capacidad)

    % Evalúa factibilidad de tiempo de la misma secuencia

    tiempoMax = vehVector(3);
    tiempoPrueba = tieMat ( length( distMat ) , nuevasecuencia( 2
))...
        + clieDem( nuevasecuencia ( 2 ) , 3 );
    for j=2:length(nuevasecuencia)-1
        tiempoPrueba = tiempoPrueba +...
            tieMat(nuevasecuencia(j),nuevasecuencia(j+1)) +
        clieDem(nuevasecuencia(j+1),3);
    end
    tiempoPrueba = tiempoPrueba + ...
        tieMat(nuevasecuencia( length(nuevasecuencia) ) ,
nuevasecuencia(1) );

    if (tiempoPrueba <= tiempoMax)

        %Actualice la secuencia
        secuenciactual = nuevasecuencia;
        rutaEnjambre{i,vehiculoasignado,1} = secuenciactual;

        %Actualice el costo
        costoactual = nuevocosto;
        rutaEnjambre{i,vehiculoasignado,2} = costoactual;

        %Actualice la carga de la secuencia
        rutaEnjambre{i,vehiculoasignado,3} = vectorCarga;

        %Actualice los tiempos de la secuencia
        rutaEnjambre{i,vehiculoasignado,4} = tiempoPrueba;

        % Reoptimizar ruta con 2-opt, sólo cuando hay al menos
        %4 nodos en la ruta
        if length(secuenciactual) >= 4
            [costoactual , secuenciactual] =
exchange_2opt(secuenciactual,distMat);
            rutaEnjambre{i,vehiculoasignado,1} = secuenciactual;
            rutaEnjambre{i,vehiculoasignado,2} = costoactual;
        end
    end
end

```

```

                break %for... Para continuar con el siguiente vehículo en
prioridad.
            end
        end

        end
    end
end

```

## **FUNCIÓN 'performance'**

Calcula el desempeño de la función fitness para cada partícula del enjambre.

### **Entradas:**

- rutaEnjambre: Arreglo de celdas que contiene la información de las rutas construidas hasta el momento.
- numPar: Número de partículas que conforman el enjambre.
- vehVector: Vector que contiene la información de la flota de vehículos disponibles para el problema.

### **Salidas:**

- fitness: matriz que contiene el valor fitness para cada partícula del enjambre.

```

function [fitness] = performance(rutaEnjambre,numPar,vehVector)
%% Preallocating
    fitness(:,1) = zeros(1,numPar);
    fitness(:,2) = (1:numPar);

%% calcula número de vehículos
numVeh = vehVector(1);

% Calcula valor fitness de cada partícula
for i=1:numPar
    for j=1:numVeh
        if (rutaEnjambre{i,j,2} ~= 0)

                fitness(i,1) = fitness(i,1) + vehVector(4) +
(vehVector(5)*rutaEnjambre{i,j,2});
            end
        end
    end
end

```

## **FUNCIÓN 'behaviorINDICES'**

Calcula y/o actualiza los comportamiento cognitivos y sociales del enjambre de partículas.

### **Entradas:**

- fitness: Matriz que almacena el valor de la función fitness para cada partícula.
- perBest: Matriz que contiene los vectores de mejor posición personal para cada partícula.
- globalBest: Mejor posición obtenida en el enjambre. Corresponde a la solución del problema.
- localBest: Mejor posición para cada vecindario de partículas.
- curSwarm: Matriz que almacena la posición actual del enjambre en todas las dimensiones del problema.
- numPar: Número de partículas que conforman el enjambre.
- N: Número de clientes del problema.
- vehVector: Vector que contiene la información de la flota de vehículos disponibles para el problema.
- rutaEnjambre: Arreglo de celdas que contiene la información de las rutas construidas hasta el momento.
- solución: Matriz que almacena las secuencias de las rutas y los costos para cada vehículo de la flota disponible.
- vecinos: Valor k que representa el tamaño de las vecindades utilizadas para calcular el comportamiento Nbest del enjambre.

### **Salidas:**

- fitness: Matriz que almacena el valor de la función fitness para cada partícula.
- perBest: Vector que contiene la mejor posición personal para cada partícula.
- globalBest: Mejor posición obtenida en el enjambre. Corresponde a la solución del problema.
- localBest: Mejor posición para cada vecindario de partículas.
- Nbest: Matriz que almacena los vectores de mejor posición que maximiza la relación fitness-distance-ratio.
- solución: Matriz que almacena las secuencias de las rutas y los costos para cada vehículo de la flota disponible.

```

function [perBest,globalBest,localBest,Nbest,solucion,fitness] =
behaviorINDICES(fitness,perBest,globalBest,localBest,curSwarm, numPar,N,
vehVector, rutaEnjambre,solucion,vecinos)

%% Actualizar perBest para cada partícula

for i=1:size(perBest,1)

    if fitness(i)< perBest{i,1,2}
        perBest{i,1,2} = fitness( i );
        perBest{i,:,1} = curSwarm(i,:);
    end
end

%% Actualizar globalBest de todo el enjambre.

%En este paso se obtiene la información de la solución que arroja la
mejor
%posición global, que será, al finalizar el algoritmo, la mejor solución
%obtenida para el problema VRPSPD.

[C,I] = min(fitness(:,1));

if C < globalBest{1,2}
    globalBest{1,2} = C;
    globalBest{1,1} = curSwarm(I,:);

    for i = 1 : vehVector(1)
        solucion{i,1} = rutaEnjambre{I,i,1}; %Rutas de la solución de la
partícula globalBest
        solucion{i,2} = rutaEnjambre{I,i,3}; %Carga de los vehiculos de
la solución de la partícula globalBest
        solucion{i,3} = rutaEnjambre{I,i,4}; %Tiempo de las secuencia de
la solución de la partícula globalBest
        solucion{i,4} = rutaEnjambre{I,i,2}; %Costo de las secuencia de
la solución de la partícula globalBest
    end
end

%% Actualizar local Best
% Se debe comenzar por definir las vecindades de las partículas, que
serán
% tomadas a partir del índice de cada una. Se tomarán grupos de
"vecinos"
%partículas.

%Comienza por el primer vecindario
i = 1;

```

```

%Recorre cada uno de los vecindarios. En cada uno de ellos calcula
%localBest
for j = 1 : length(fitness)/vecinos

    %En el vector grupo estarán las partículas que pertenecen al
    %vecindario i y se marcarán como tal.
    grupo = fitness(i:i+vecinos-1,1);
    fitness(i:i+vecinos-1,3) = j;

    [C,I] = min(grupo);
    particula = fitness(I,2);

    if C < localBest{j,2}
        localBest{j,2} = C;
        localBest{j,1} = curSwarm(particula,:);
    end
    i=i+vecinos;
end

%% Generar Nbest (Fitness Distance Ratio - FDR)
% Numero de dimensiones
dimensiones = N + 2 * vehVector(1);

%Crear matriz FDR
FDR = zeros(numPar,dimensiones);

%Se debe sacar la matriz de las posiciones de perBest
matrizperBest = zeros ( numPar , dimensiones );

for i = 1:numPar
    matrizperBest ( i,: ) = perBest { i , 1 , 1 };
end

% calcular Función FDR de las dimensiones de cada partícula
for i = 1: numPar
    for j = 1 : dimensiones
        FDR(i , j) = ( fitness(i) - perBest{i,1,2}) /...
            abs( curSwarm(i,j) - matrizperBest(i,j));
    end
end

%Generar Nbest de la partícula que maximiza FDR para cada dimensión
%Preallocating
Nbest = (zeros);

for j = 1 : dimensiones
    [~,I] = max(FDR(:,j));
    Nbest(j) = matrizperBest(I,j);
end

```

## **FUNCIÓN ‘actualizar’**

Calcula la nueva velocidad y posición de las partículas para la siguiente iteración del algoritmo y evalúa el criterio de parada.

### **Entradas:**

- numPar: Número de partículas que conforman el enjambre.
- N: Número de clientes del problema
- vehVector: Vector que contiene la información de la flota de vehículos disponibles para el problema.
- curSwarm: Matriz que almacena la posición actual del enjambre en todas las dimensiones del problema.
- fitness: Matriz que almacena el valor de la función fitness para cada partícula.
- velocidad: Matriz que almacena los vectores de velocidad de cada partícula.
- perBest: Matriz que contiene los vectores de mejor posición personal para cada partícula.
- globalBest: Mejor posición obtenida en el enjambre. Corresponde a la solución del problema.
- localBest: Mejor posición para cada vecindario de partículas.
- Nbest: Matriz que almacena los vectores de mejor posición que maximiza la relación fitness-distance-ratio.
- inertiaInicial: Parámetro de peso inercial de las partículas definido antes de iniciar la ejecución del algoritmo.
- inertiaFinal: Parámetro de peso inercial de las partículas al final del algoritmo.
- iter: Iteración actual
- iterMax: Criterio de parada (máximo número de iteraciones).
- Cpersonal: Constante de aceleración para el comportamiento personal de las partículas. Definida como parámetro del algoritmo.
- Cglobal: Constante de aceleración asociada al comportamiento social ‘Global Best’ de las partículas. Definida como parámetro del algoritmo.
- Cneighbor: Constante de aceleración para el comportamiento ‘Near neighbor best’ del enjambre de partículas. Definida como parámetro del algoritmo.

- Clocal: Constante de aceleración para el comportamiento 'Local best' del enjambre de partículas Definida como parámetro del algoritmo.
- rangoMin: Límite inferior del espacio de búsqueda.
- rangoMax: Límite superior del espacio de búsqueda.

### **Salidas:**

- curSwarm: Matriz que almacena la posición actual del enjambre en todas las dimensiones del problema.
- velocidad: Matriz que almacena los vectores de velocidad de cada partícula.

```
function [velocidad,curSwarm] = actualizar(numPar, N, vehVector,
curSwarm,fitness, velocidad, perBest, globalBest,localBest,Nbest,
inertialInicial, inertialFinal, iter,iterMax, Cpersonal,
Cglobal,Cneighbor,Clocal,rangoMax,rangoMin)

% Numero de dimensiones
dimensiones = N + 2 * vehVector(1);

%% Actualiza Peso inercial
inertialActual = inertialFinal + (( iter - iterMax ) / ( 1 - iterMax
))...
    * (inertialInicial - inertialFinal);

%% Actualiza Velocidad
for i = 1 : numPar

    %Toma el vector perBest de cada particula
    vectorperBest = perBest{i,1,1};

    %Toma el vector globalBest del enjambre
    vectorgloBest = globalBest{1,1};

    %Toma el vector localBest
    vectorlocalBest = localBest{fitness(i,3),1};

    for j = 1 : dimensiones

        %Actualiza velocidad para cada dimensión de cada partícula
        velocidad(i,j) = inertialActual * velocidad(i,j) +...
            Cpersonal * rand(1) * (vectorperBest(j) - curSwarm(i,j))+...
            Cglobal * rand(1) * (vectorgloBest(j) - curSwarm(i,j))+...
            Clocal * rand(1) * (vectorlocalBest(j) - curSwarm(i,j))+...
            Cneighbor *rand(1) * (Nbest(j) - curSwarm(i,j));
    end
end
```

```

    end
end
%% Actualiza posición del enjambre
for i = 1 : numPar
    for j = 1 : dimensiones
        curSwarm (i,j) = curSwarm (i,j) + velocidad (i,j) ;
    end
end

%% Evalúa límites del espacio de búsqueda.
for i = 1 : numPar
    for j = 1 : dimensiones

        if curSwarm(i,j) > rangoMax
            curSwarm(i,j) = rangoMax;
            velocidad (i,j) = 0;
        end

        if curSwarm(i,j) < rangoMin
            curSwarm(i,j) = rangoMin;
            velocidad (i,j) = 0;
        end
    end
end
end

```

## Anexo B. Instancias 'VRPSPD'

### Instancia r101

	Nro Clientes	Capacidad vehículos		
	100	200		
	<i>Cord X</i>	<i>Cord Y</i>	<i>Entrega</i>	<i>Recogida</i>
<b>Depósito</b>	35	35	0	0
<b>1</b>	41	49	10	41
<b>2</b>	35	17	7	12
<b>3</b>	55	45	13	30
<b>4</b>	55	20	19	7
<b>5</b>	15	30	26	7
<b>6</b>	25	30	3	37
<b>7</b>	20	50	5	28
<b>8</b>	10	43	9	40
<b>9</b>	55	60	16	31
<b>10</b>	30	60	16	20
<b>11</b>	20	65	12	27
<b>12</b>	50	35	19	39
<b>13</b>	30	25	23	41
<b>14</b>	15	10	20	24
<b>15</b>	30	5	8	5
<b>16</b>	10	20	19	28
<b>17</b>	5	30	2	10
<b>18</b>	20	40	12	4
<b>19</b>	15	60	17	11
<b>20</b>	45	65	9	7
<b>21</b>	45	20	11	34
<b>22</b>	45	10	18	32
<b>23</b>	55	5	29	15
<b>24</b>	65	35	3	19
<b>25</b>	65	20	6	22
<b>26</b>	45	30	17	8
<b>27</b>	35	40	16	12
<b>28</b>	41	37	16	28
<b>29</b>	64	42	9	26
<b>30</b>	40	60	21	15

<b>31</b>	31	52	27	31
<b>32</b>	35	69	23	38
<b>33</b>	53	52	11	28
<b>34</b>	65	55	14	16
<b>35</b>	63	65	8	7
<b>36</b>	2	60	5	27
<b>37</b>	20	20	8	24
<b>38</b>	5	5	16	1
<b>39</b>	60	12	31	24
<b>40</b>	40	25	9	10
<b>41</b>	42	7	5	1
<b>42</b>	24	12	5	36
<b>43</b>	23	3	7	22
<b>44</b>	11	14	18	40
<b>45</b>	6	38	16	38
<b>46</b>	2	48	1	33
<b>47</b>	8	56	27	10
<b>48</b>	13	52	36	32
<b>49</b>	6	68	30	24
<b>50</b>	47	47	13	32
<b>51</b>	49	58	10	16
<b>52</b>	27	43	9	15
<b>53</b>	37	31	14	25
<b>54</b>	57	29	18	40
<b>55</b>	63	23	2	3
<b>56</b>	53	12	6	22
<b>57</b>	32	12	7	41
<b>58</b>	36	26	18	8
<b>59</b>	21	24	28	10
<b>60</b>	17	34	3	34
<b>61</b>	12	24	13	34
<b>62</b>	24	58	19	27
<b>63</b>	27	69	10	5
<b>64</b>	15	77	9	36
<b>65</b>	62	77	20	19
<b>66</b>	49	73	25	35
<b>67</b>	67	5	25	40
<b>68</b>	56	39	36	41
<b>69</b>	37	47	6	5
<b>70</b>	37	56	5	39

<b>71</b>	57	68	15	29
<b>72</b>	47	16	25	5
<b>73</b>	44	17	9	13
<b>74</b>	46	13	8	5
<b>75</b>	49	11	18	21
<b>76</b>	49	42	13	29
<b>77</b>	53	43	14	27
<b>78</b>	61	52	3	21
<b>79</b>	57	48	23	27
<b>80</b>	56	37	6	18
<b>81</b>	55	54	26	28
<b>82</b>	15	47	16	32
<b>83</b>	14	37	11	16
<b>84</b>	11	31	7	12
<b>85</b>	16	22	41	39
<b>86</b>	4	18	35	33
<b>87</b>	28	18	26	25
<b>88</b>	26	52	9	18
<b>89</b>	26	35	15	24
<b>90</b>	31	67	3	12
<b>91</b>	15	19	1	41
<b>92</b>	22	22	2	19
<b>93</b>	18	24	22	32
<b>94</b>	26	27	27	41
<b>95</b>	25	24	20	38
<b>96</b>	22	27	11	9
<b>97</b>	25	21	12	10
<b>98</b>	19	21	10	25
<b>99</b>	20	26	9	30
<b>100</b>	18	18	17	31

### Instancia r201

	<b>Nro Clientes</b>	<b>Capacidad vehículos</b>		
	100	1000		
	<b>Cord X</b>	<b>Cord Y</b>	<b>Entrega</b>	<b>Recogida</b>
<b>Depósito</b>	35	35	0	0
<b>1</b>	41	49	10	18

<b>2</b>	35	17	7	23
<b>3</b>	55	45	13	13
<b>4</b>	55	20	19	37
<b>5</b>	15	30	26	1
<b>6</b>	25	30	3	3
<b>7</b>	20	50	5	17
<b>8</b>	10	43	9	5
<b>9</b>	55	60	16	26
<b>10</b>	30	60	16	28
<b>11</b>	20	65	12	35
<b>12</b>	50	35	19	20
<b>13</b>	30	25	23	30
<b>14</b>	15	10	20	27
<b>15</b>	30	5	8	33
<b>16</b>	10	20	19	13
<b>17</b>	5	30	2	34
<b>18</b>	20	40	12	35
<b>19</b>	15	60	17	5
<b>20</b>	45	65	9	35
<b>21</b>	45	20	11	38
<b>22</b>	45	10	18	37
<b>23</b>	55	5	29	5
<b>24</b>	65	35	3	26
<b>25</b>	65	20	6	37
<b>26</b>	45	30	17	9
<b>27</b>	35	40	16	7
<b>28</b>	41	37	16	15
<b>29</b>	64	42	9	21
<b>30</b>	40	60	21	27
<b>31</b>	31	52	27	7
<b>32</b>	35	69	23	29
<b>33</b>	53	52	11	5
<b>34</b>	65	55	14	33
<b>35</b>	63	65	8	24
<b>36</b>	2	60	5	18
<b>37</b>	20	20	8	3
<b>38</b>	5	5	16	14
<b>39</b>	60	12	31	33
<b>40</b>	40	25	9	22
<b>41</b>	42	7	5	12

<b>42</b>	24	12	5	7
<b>43</b>	23	3	7	34
<b>44</b>	11	14	18	33
<b>45</b>	6	38	16	32
<b>46</b>	2	48	1	38
<b>47</b>	8	56	27	26
<b>48</b>	13	52	36	39
<b>49</b>	6	68	30	30
<b>50</b>	47	47	13	8
<b>51</b>	49	58	10	30
<b>52</b>	27	43	9	25
<b>53</b>	37	31	14	37
<b>54</b>	57	29	18	35
<b>55</b>	63	23	2	34
<b>56</b>	53	12	6	21
<b>57</b>	32	12	7	41
<b>58</b>	36	26	18	31
<b>59</b>	21	24	28	16
<b>60</b>	17	34	3	31
<b>61</b>	12	24	13	14
<b>62</b>	24	58	19	8
<b>63</b>	27	69	10	26
<b>64</b>	15	77	9	40
<b>65</b>	62	77	20	18
<b>66</b>	49	73	25	26
<b>67</b>	67	5	25	18
<b>68</b>	56	39	36	38
<b>69</b>	37	47	6	28
<b>70</b>	37	56	5	29
<b>71</b>	57	68	15	37
<b>72</b>	47	16	25	20
<b>73</b>	44	17	9	26
<b>74</b>	46	13	8	17
<b>75</b>	49	11	18	23
<b>76</b>	49	42	13	16
<b>77</b>	53	43	14	17
<b>78</b>	61	52	3	2
<b>79</b>	57	48	23	39
<b>80</b>	56	37	6	6
<b>81</b>	55	54	26	21

<b>82</b>	15	47	16	18
<b>83</b>	14	37	11	11
<b>84</b>	11	31	7	38
<b>85</b>	16	22	41	26
<b>86</b>	4	18	35	2
<b>87</b>	28	18	26	38
<b>88</b>	26	52	9	8
<b>89</b>	26	35	15	19
<b>90</b>	31	67	3	33
<b>91</b>	15	19	1	4
<b>92</b>	22	22	2	28
<b>93</b>	18	24	22	33
<b>94</b>	26	27	27	38
<b>95</b>	25	24	20	11
<b>96</b>	22	27	11	33
<b>97</b>	25	21	12	1
<b>98</b>	19	21	10	22
<b>99</b>	20	26	9	1
<b>100</b>	18	18	17	21

### Instancia c101

	<b>Nro Clientes</b>	<b>Capacidad vehículos</b>		
	100	200		
	<b>Cord X</b>	<b>Cord Y</b>	<b>Entrega</b>	<b>Recogida</b>
<b>Depósito</b>	40	50	0	0
<b>1</b>	45	68	10	40
<b>2</b>	45	70	30	20
<b>3</b>	42	66	10	40
<b>4</b>	42	68	10	20
<b>5</b>	42	65	10	20
<b>6</b>	40	69	20	50
<b>7</b>	40	66	20	20
<b>8</b>	38	68	20	10
<b>9</b>	38	70	10	40
<b>10</b>	35	66	10	50
<b>11</b>	35	69	10	50

<b>12</b>	25	85	20	40
<b>13</b>	22	75	30	10
<b>14</b>	22	85	10	20
<b>15</b>	20	80	40	20
<b>16</b>	20	85	40	30
<b>17</b>	18	75	20	40
<b>18</b>	15	75	20	50
<b>19</b>	15	80	10	10
<b>20</b>	30	50	10	20
<b>21</b>	30	52	20	50
<b>22</b>	28	52	20	20
<b>23</b>	28	55	10	20
<b>24</b>	25	50	10	50
<b>25</b>	25	52	40	10
<b>26</b>	25	55	10	50
<b>27</b>	23	52	10	30
<b>28</b>	23	55	20	20
<b>29</b>	20	50	10	30
<b>30</b>	20	55	10	50
<b>31</b>	10	35	20	20
<b>32</b>	10	40	30	40
<b>33</b>	8	40	40	30
<b>34</b>	8	45	20	50
<b>35</b>	5	35	10	30
<b>36</b>	5	45	10	50
<b>37</b>	2	40	20	20
<b>38</b>	0	40	30	20
<b>39</b>	0	45	20	40
<b>40</b>	35	30	10	40
<b>41</b>	35	32	10	30
<b>42</b>	33	32	20	10
<b>43</b>	33	35	10	50
<b>44</b>	32	30	10	50
<b>45</b>	30	30	10	10
<b>46</b>	30	32	30	40
<b>47</b>	30	35	10	20
<b>48</b>	28	30	10	20
<b>49</b>	28	35	10	10
<b>50</b>	26	32	10	20
<b>51</b>	25	30	10	10

<b>52</b>	25	35	10	30
<b>53</b>	44	5	20	40
<b>54</b>	42	10	40	20
<b>55</b>	42	15	10	10
<b>56</b>	40	5	30	20
<b>57</b>	40	15	40	20
<b>58</b>	38	5	30	40
<b>59</b>	38	15	10	50
<b>60</b>	35	5	20	30
<b>61</b>	50	30	10	10
<b>62</b>	50	35	20	40
<b>63</b>	50	40	50	30
<b>64</b>	48	30	10	10
<b>65</b>	48	40	10	40
<b>66</b>	47	35	10	20
<b>67</b>	47	40	10	30
<b>68</b>	45	30	10	50
<b>69</b>	45	35	10	20
<b>70</b>	95	30	30	20
<b>71</b>	95	35	20	30
<b>72</b>	53	30	10	40
<b>73</b>	92	30	10	30
<b>74</b>	53	35	50	50
<b>75</b>	45	65	20	40
<b>76</b>	90	35	10	40
<b>77</b>	88	30	10	40
<b>78</b>	88	35	20	40
<b>79</b>	87	30	10	50
<b>80</b>	85	25	10	20
<b>81</b>	85	35	30	50
<b>82</b>	75	55	20	40
<b>83</b>	72	55	10	20
<b>84</b>	70	58	20	30
<b>85</b>	68	60	30	30
<b>86</b>	66	55	10	30
<b>87</b>	65	55	20	30
<b>88</b>	65	60	30	30
<b>89</b>	63	58	10	40
<b>90</b>	60	55	10	50
<b>91</b>	60	60	10	50

<b>92</b>	67	85	20	30
<b>93</b>	65	85	40	10
<b>94</b>	65	82	10	10
<b>95</b>	62	80	30	20
<b>96</b>	60	80	10	30
<b>97</b>	60	85	30	10
<b>98</b>	58	75	20	30
<b>99</b>	55	80	10	40
<b>100</b>	55	85	20	40

### Instancia c201

	Nro Clientes	Capacidad vehículos		
	100	700		
	<i>Cord X</i>	<i>Cord Y</i>	<i>Entrega</i>	<i>Recogida</i>
<b>Depósito</b>	40	50	0	0
<b>1</b>	52	75	10	10
<b>2</b>	45	70	30	30
<b>3</b>	62	69	10	20
<b>4</b>	60	66	10	50
<b>5</b>	42	65	10	20
<b>6</b>	16	42	20	10
<b>7</b>	58	70	20	10
<b>8</b>	34	60	20	20
<b>9</b>	28	70	10	30
<b>10</b>	35	66	10	10
<b>11</b>	35	69	10	10
<b>12</b>	25	85	20	10
<b>13</b>	22	75	30	40
<b>14</b>	22	85	10	30
<b>15</b>	20	80	40	10
<b>16</b>	20	85	40	20
<b>17</b>	18	75	20	20
<b>18</b>	15	75	20	40
<b>19</b>	15	80	10	10
<b>20</b>	30	50	10	50
<b>21</b>	30	56	20	10
<b>22</b>	28	52	20	20

<b>23</b>	14	66	10	10
<b>24</b>	25	50	10	10
<b>25</b>	22	66	40	30
<b>26</b>	8	62	10	50
<b>27</b>	23	52	10	30
<b>28</b>	4	55	20	50
<b>29</b>	20	50	10	30
<b>30</b>	20	55	10	10
<b>31</b>	10	35	20	40
<b>32</b>	10	40	30	30
<b>33</b>	8	40	40	50
<b>34</b>	8	45	20	20
<b>35</b>	5	35	10	50
<b>36</b>	5	45	10	40
<b>37</b>	2	40	20	50
<b>38</b>	0	40	30	40
<b>39</b>	0	45	20	50
<b>40</b>	36	18	10	50
<b>41</b>	35	32	10	50
<b>42</b>	33	32	20	30
<b>43</b>	33	35	10	20
<b>44</b>	32	20	10	40
<b>45</b>	30	30	10	50
<b>46</b>	34	25	30	50
<b>47</b>	30	35	10	40
<b>48</b>	36	40	10	30
<b>49</b>	48	20	10	20
<b>50</b>	26	32	10	40
<b>51</b>	25	30	10	30
<b>52</b>	25	35	10	50
<b>53</b>	44	5	20	50
<b>54</b>	42	10	40	10
<b>55</b>	42	15	10	40
<b>56</b>	40	5	30	20
<b>57</b>	38	15	40	20
<b>58</b>	38	5	30	20
<b>59</b>	38	10	10	10
<b>60</b>	35	5	20	20
<b>61</b>	50	30	10	40
<b>62</b>	50	35	20	20

<b>63</b>	50	40	50	50
<b>64</b>	48	30	10	10
<b>65</b>	44	25	10	40
<b>66</b>	47	35	10	10
<b>67</b>	47	40	10	10
<b>68</b>	42	30	10	40
<b>69</b>	45	35	10	50
<b>70</b>	95	30	30	10
<b>71</b>	95	35	20	40
<b>72</b>	53	30	10	10
<b>73</b>	92	30	10	40
<b>74</b>	53	35	50	20
<b>75</b>	45	65	20	40
<b>76</b>	90	35	10	30
<b>77</b>	72	45	10	30
<b>78</b>	78	40	20	10
<b>79</b>	87	30	10	10
<b>80</b>	85	25	10	50
<b>81</b>	85	35	30	40
<b>82</b>	75	55	20	30
<b>83</b>	72	55	10	20
<b>84</b>	70	58	20	20
<b>85</b>	86	46	30	10
<b>86</b>	66	55	10	40
<b>87</b>	64	46	20	50
<b>88</b>	65	60	30	10
<b>89</b>	56	64	10	30
<b>90</b>	60	55	10	30
<b>91</b>	60	60	10	10
<b>92</b>	67	85	20	40
<b>93</b>	42	58	40	20
<b>94</b>	65	82	10	10
<b>95</b>	62	80	30	40
<b>96</b>	62	40	10	50
<b>97</b>	60	85	30	50
<b>98</b>	58	75	20	20
<b>99</b>	55	80	10	50
<b>100</b>	55	85	20	20

**Instancia rc101**

	<b>Nro Clientes</b>	<b>Capacidad vehículos</b>		
	100	200		
	<b><i>Cord X</i></b>	<b><i>Cord Y</i></b>	<b><i>Entrega</i></b>	<b><i>Recogida</i></b>
<b><i>Depósito</i></b>	40	50	0	0
<b>1</b>	25	85	20	2
<b>2</b>	22	75	30	2
<b>3</b>	22	85	10	16
<b>4</b>	20	80	40	18
<b>5</b>	20	85	20	38
<b>6</b>	18	75	20	18
<b>7</b>	15	75	20	24
<b>8</b>	15	80	10	26
<b>9</b>	10	35	20	12
<b>10</b>	10	40	30	20
<b>11</b>	8	40	40	20
<b>12</b>	8	45	20	12
<b>13</b>	5	35	10	10
<b>14</b>	5	45	10	24
<b>15</b>	2	40	20	20
<b>16</b>	0	40	20	22
<b>17</b>	0	45	20	14
<b>18</b>	44	5	20	8
<b>19</b>	42	10	40	32
<b>20</b>	42	15	10	14
<b>21</b>	40	5	10	10
<b>22</b>	40	15	40	24
<b>23</b>	38	5	30	22
<b>24</b>	38	15	10	18
<b>25</b>	35	5	20	10
<b>26</b>	95	30	30	2
<b>27</b>	95	35	20	8
<b>28</b>	92	30	10	14
<b>29</b>	90	35	10	8
<b>30</b>	88	30	10	32
<b>31</b>	88	35	20	22
<b>32</b>	87	30	10	38
<b>33</b>	85	25	10	32

<b>34</b>	85	35	30	4
<b>35</b>	67	85	20	32
<b>36</b>	65	85	40	10
<b>37</b>	65	82	10	34
<b>38</b>	62	80	30	14
<b>39</b>	60	80	10	20
<b>40</b>	60	85	30	24
<b>41</b>	58	75	20	18
<b>42</b>	55	80	10	2
<b>43</b>	55	85	20	36
<b>44</b>	55	82	10	16
<b>45</b>	20	82	10	24
<b>46</b>	18	80	10	8
<b>47</b>	2	45	10	18
<b>48</b>	42	5	10	38
<b>49</b>	42	12	10	26
<b>50</b>	72	35	30	28
<b>51</b>	55	20	19	6
<b>52</b>	25	30	3	16
<b>53</b>	20	50	5	20
<b>54</b>	55	60	16	40
<b>55</b>	30	60	16	10
<b>56</b>	50	35	19	10
<b>57</b>	30	25	23	10
<b>58</b>	15	10	20	16
<b>59</b>	10	20	19	14
<b>60</b>	15	60	17	10
<b>61</b>	45	65	9	26
<b>62</b>	65	35	3	14
<b>63</b>	65	20	6	38
<b>64</b>	45	30	17	16
<b>65</b>	35	40	16	24
<b>66</b>	41	37	16	26
<b>67</b>	64	42	9	2
<b>68</b>	40	60	21	26
<b>69</b>	31	52	27	6
<b>70</b>	35	69	23	24
<b>71</b>	65	55	14	4
<b>72</b>	63	65	8	30
<b>73</b>	2	60	5	16

<b>74</b>	20	20	8	34
<b>75</b>	5	5	16	36
<b>76</b>	60	12	31	36
<b>77</b>	23	3	7	38
<b>78</b>	8	56	27	14
<b>79</b>	6	68	30	30
<b>80</b>	47	47	13	16
<b>81</b>	49	58	10	18
<b>82</b>	27	43	9	30
<b>83</b>	37	31	14	2
<b>84</b>	57	29	18	20
<b>85</b>	63	23	2	20
<b>86</b>	21	24	28	10
<b>87</b>	12	24	13	22
<b>88</b>	24	58	19	30
<b>89</b>	67	5	25	14
<b>90</b>	37	47	6	20
<b>91</b>	49	42	13	10
<b>92</b>	53	43	14	10
<b>93</b>	61	52	3	12
<b>94</b>	57	48	23	36
<b>95</b>	56	37	6	14
<b>96</b>	55	54	26	6
<b>97</b>	4	18	35	22
<b>98</b>	26	52	9	10
<b>99</b>	26	35	15	20
<b>100</b>	31	67	3	34

### Instancia rc201

	Nro Clientes	Capacidad vehículos		
	100	1000		
	<i>Cord X</i>	<i>Cord Y</i>	<i>Entrega</i>	<i>Recogida</i>
<b>Depósito</b>	35	35	0	0
<b>1</b>	41	49	10	18
<b>2</b>	35	17	7	23
<b>3</b>	55	45	13	13
<b>4</b>	55	20	19	37

<b>5</b>	15	30	26	1
<b>6</b>	25	30	3	3
<b>7</b>	20	50	5	17
<b>8</b>	10	43	9	5
<b>9</b>	55	60	16	26
<b>10</b>	30	60	16	28
<b>11</b>	20	65	12	35
<b>12</b>	50	35	19	20
<b>13</b>	30	25	23	30
<b>14</b>	15	10	20	27
<b>15</b>	30	5	8	33
<b>16</b>	10	20	19	13
<b>17</b>	5	30	2	34
<b>18</b>	20	40	12	35
<b>19</b>	15	60	17	5
<b>20</b>	45	65	9	35
<b>21</b>	45	20	11	38
<b>22</b>	45	10	18	37
<b>23</b>	55	5	29	5
<b>24</b>	65	35	3	26
<b>25</b>	65	20	6	37
<b>26</b>	45	30	17	9
<b>27</b>	35	40	16	7
<b>28</b>	41	37	16	15
<b>29</b>	64	42	9	21
<b>30</b>	40	60	21	27
<b>31</b>	31	52	27	7
<b>32</b>	35	69	23	29
<b>33</b>	53	52	11	5
<b>34</b>	65	55	14	33
<b>35</b>	63	65	8	24
<b>36</b>	2	60	5	18
<b>37</b>	20	20	8	3
<b>38</b>	5	5	16	14
<b>39</b>	60	12	31	33
<b>40</b>	40	25	9	22
<b>41</b>	42	7	5	12
<b>42</b>	24	12	5	7
<b>43</b>	23	3	7	34
<b>44</b>	11	14	18	33

<b>45</b>	6	38	16	32
<b>46</b>	2	48	1	38
<b>47</b>	8	56	27	26
<b>48</b>	13	52	36	39
<b>49</b>	6	68	30	30
<b>50</b>	47	47	13	8
<b>51</b>	49	58	10	30
<b>52</b>	27	43	9	25
<b>53</b>	37	31	14	37
<b>54</b>	57	29	18	35
<b>55</b>	63	23	2	34
<b>56</b>	53	12	6	21
<b>57</b>	32	12	7	41
<b>58</b>	36	26	18	31
<b>59</b>	21	24	28	16
<b>60</b>	17	34	3	31
<b>61</b>	12	24	13	14
<b>62</b>	24	58	19	8
<b>63</b>	27	69	10	26
<b>64</b>	15	77	9	40
<b>65</b>	62	77	20	18
<b>66</b>	49	73	25	26
<b>67</b>	67	5	25	18
<b>68</b>	56	39	36	38
<b>69</b>	37	47	6	28
<b>70</b>	37	56	5	29
<b>71</b>	57	68	15	37
<b>72</b>	47	16	25	20
<b>73</b>	44	17	9	26
<b>74</b>	46	13	8	17
<b>75</b>	49	11	18	23
<b>76</b>	49	42	13	16
<b>77</b>	53	43	14	17
<b>78</b>	61	52	3	2
<b>79</b>	57	48	23	39
<b>80</b>	56	37	6	6
<b>81</b>	55	54	26	21
<b>82</b>	15	47	16	18
<b>83</b>	14	37	11	11
<b>84</b>	11	31	7	38

85	16	22	41	26
86	4	18	35	2
87	28	18	26	38
88	26	52	9	8
89	26	35	15	19
90	31	67	3	33
91	15	19	1	4
92	22	22	2	28
93	18	24	22	33
94	26	27	27	38
95	25	24	20	11
96	22	27	11	33
97	25	21	12	1
98	19	21	10	22
99	20	26	9	1
100	18	18	17	21

### Instancia R121

	Nro Clientes	Capacidad vehículos		
	200	200		
	<i>Cord X</i>	<i>Cord Y</i>	<i>Entrega</i>	<i>Recogida</i>
<i>Depósito</i>	70	70	0	0
<b>1</b>	107	77	34	5
<b>2</b>	109	139	8	22
<b>3</b>	120	22	39	10
<b>4</b>	48	47	19	41
<b>5</b>	116	22	32	10
<b>6</b>	12	138	14	4
<b>7</b>	86	40	22	4
<b>8</b>	121	124	21	15
<b>9</b>	61	57	35	23
<b>10</b>	40	113	23	4
<b>11</b>	129	24	18	4
<b>12</b>	12	84	16	9
<b>13</b>	44	116	27	33
<b>14</b>	102	52	15	19
<b>15</b>	41	36	13	4

<b>16</b>	132	133	24	10
<b>17</b>	104	139	29	9
<b>18</b>	104	54	13	30
<b>19</b>	22	104	11	3
<b>20</b>	46	133	7	43
<b>21</b>	138	78	43	1
<b>22</b>	16	92	10	15
<b>23</b>	18	104	18	2
<b>24</b>	66	82	27	5
<b>25</b>	107	25	13	23
<b>26</b>	139	73	10	43
<b>27</b>	101	0	9	19
<b>28</b>	90	14	31	40
<b>29</b>	20	69	13	19
<b>30</b>	64	132	8	8
<b>31</b>	115	82	30	30
<b>32</b>	54	106	9	18
<b>33</b>	30	21	12	39
<b>34</b>	63	129	7	12
<b>35</b>	82	100	18	41
<b>36</b>	108	30	18	30
<b>37</b>	37	73	33	37
<b>38</b>	50	112	18	28
<b>39</b>	47	83	1	43
<b>40</b>	92	138	12	43
<b>41</b>	81	36	17	38
<b>42</b>	115	124	11	19
<b>43</b>	13	48	15	12
<b>44</b>	113	35	9	30
<b>45</b>	60	84	23	40
<b>46</b>	44	58	30	39
<b>47</b>	8	87	17	29
<b>48</b>	116	105	23	25
<b>49</b>	117	4	9	15
<b>50</b>	140	42	13	35
<b>51</b>	139	68	25	20
<b>52</b>	58	95	35	37
<b>53</b>	103	119	18	39
<b>54</b>	32	52	32	9
<b>55</b>	89	39	23	33

<b>56</b>	54	41	9	12
<b>57</b>	39	100	31	12
<b>58</b>	5	99	24	11
<b>59</b>	67	96	26	3
<b>60</b>	73	96	9	18
<b>61</b>	112	13	27	28
<b>62</b>	55	127	17	12
<b>63</b>	103	44	12	41
<b>64</b>	97	12	10	7
<b>65</b>	16	62	15	28
<b>66</b>	140	36	1	3
<b>67</b>	90	44	6	3
<b>68</b>	113	32	36	33
<b>69</b>	104	73	17	42
<b>70</b>	36	38	5	4
<b>71</b>	15	4	23	29
<b>72</b>	54	33	20	1
<b>73</b>	70	94	32	34
<b>74</b>	35	90	20	10
<b>75</b>	88	29	33	33
<b>76</b>	109	131	16	21
<b>77</b>	10	111	21	18
<b>78</b>	138	105	24	8
<b>79</b>	113	45	4	5
<b>80</b>	92	22	18	41
<b>81</b>	85	75	37	33
<b>82</b>	10	86	14	19
<b>83</b>	51	92	9	12
<b>84</b>	46	112	10	10
<b>85</b>	12	48	4	5
<b>86</b>	137	7	40	33
<b>87</b>	79	80	4	38
<b>88</b>	69	116	26	9
<b>89</b>	28	62	23	24
<b>90</b>	50	106	25	21
<b>91</b>	83	38	22	1
<b>92</b>	120	102	18	35
<b>93</b>	65	0	20	9
<b>94</b>	4	71	19	2
<b>95</b>	19	120	16	28

<b>96</b>	18	70	26	43
<b>97</b>	58	80	18	43
<b>98</b>	49	140	3	16
<b>99</b>	69	115	20	40
<b>100</b>	36	124	34	16
<b>101</b>	84	88	16	16
<b>102</b>	17	138	8	20
<b>103</b>	103	82	27	20
<b>104</b>	38	122	13	21
<b>105</b>	70	83	23	20
<b>106</b>	29	60	14	39
<b>107</b>	61	34	7	18
<b>108</b>	54	78	11	23
<b>109</b>	83	12	29	33
<b>110</b>	34	72	21	11
<b>111</b>	80	103	15	1
<b>112</b>	78	111	22	26
<b>113</b>	116	83	30	30
<b>114</b>	29	94	1	35
<b>115</b>	137	137	27	13
<b>116</b>	110	28	20	28
<b>117</b>	116	13	5	15
<b>118</b>	59	115	26	29
<b>119</b>	71	70	9	26
<b>120</b>	120	60	18	10
<b>121</b>	85	111	25	37
<b>122</b>	121	123	9	24
<b>123</b>	137	84	41	21
<b>124</b>	103	86	19	7
<b>125</b>	30	57	20	14
<b>126</b>	139	72	23	31
<b>127</b>	2	106	16	42
<b>128</b>	64	138	15	9
<b>129</b>	23	39	20	4
<b>130</b>	126	23	13	26
<b>131</b>	138	24	12	15
<b>132</b>	78	63	19	38
<b>133</b>	98	4	20	3
<b>134</b>	103	77	17	22
<b>135</b>	58	11	9	38

<b>136</b>	26	130	20	20
<b>137</b>	104	32	18	9
<b>138</b>	55	102	4	9
<b>139</b>	137	101	12	4
<b>140</b>	65	80	20	21
<b>141</b>	26	50	7	13
<b>142</b>	56	139	18	7
<b>143</b>	119	73	15	28
<b>144</b>	9	19	10	14
<b>145</b>	82	103	15	37
<b>146</b>	107	102	12	42
<b>147</b>	60	23	16	36
<b>148</b>	123	84	13	42
<b>149</b>	79	72	22	23
<b>150</b>	99	34	9	10
<b>151</b>	27	46	35	14
<b>152</b>	64	115	15	31
<b>153</b>	32	6	11	34
<b>154</b>	86	80	21	30
<b>155</b>	81	57	8	22
<b>156</b>	135	65	3	15
<b>157</b>	66	40	20	42
<b>158</b>	120	47	18	30
<b>159</b>	37	125	10	35
<b>160</b>	120	14	20	23
<b>161</b>	114	109	16	33
<b>162</b>	137	127	28	43
<b>163</b>	37	138	5	26
<b>164</b>	36	54	15	3
<b>165</b>	109	78	21	21
<b>166</b>	34	20	15	26
<b>167</b>	39	13	20	30
<b>168</b>	137	96	9	2
<b>169</b>	110	99	21	17
<b>170</b>	136	123	31	17
<b>171</b>	133	130	13	28
<b>172</b>	120	52	20	35
<b>173</b>	33	117	20	8
<b>174</b>	37	79	5	7
<b>175</b>	60	54	3	43

<b>176</b>	125	68	20	16
<b>177</b>	13	129	25	23
<b>178</b>	20	106	24	41
<b>179</b>	15	9	2	28
<b>180</b>	106	10	13	12
<b>181</b>	100	99	17	2
<b>182</b>	89	24	15	17
<b>183</b>	67	120	5	28
<b>184</b>	42	29	12	42
<b>185</b>	117	92	25	42
<b>186</b>	15	18	10	25
<b>187</b>	139	110	19	8
<b>188</b>	44	36	9	40
<b>189</b>	8	138	10	1
<b>190</b>	33	0	20	24
<b>191</b>	4	44	24	32
<b>192</b>	97	115	25	13
<b>193</b>	23	19	16	16
<b>194</b>	56	129	21	33
<b>195</b>	105	2	1	17
<b>196</b>	66	91	16	37
<b>197</b>	93	61	10	16
<b>198</b>	27	104	1	33
<b>199</b>	69	6	8	37
<b>200</b>	95	135	27	16

### Instancia R221

	Nro Clientes	Capacidad vehículos		
	200	1000		
	<i>Cord X</i>	<i>Cord Y</i>	<i>Entrega</i>	<i>Recogida</i>
<b>Depósito</b>	70	70	0	0
<b>1</b>	101	0	9	24
<b>2</b>	26	50	7	33
<b>3</b>	30	57	20	35
<b>4</b>	29	60	14	30
<b>5</b>	69	116	26	4
<b>6</b>	137	137	27	11

<b>7</b>	103	119	18	36
<b>8</b>	78	111	22	22
<b>9</b>	107	25	13	18
<b>10</b>	17	138	8	13
<b>11</b>	37	138	5	12
<b>12</b>	20	69	13	32
<b>13</b>	121	123	9	22
<b>14</b>	22	104	11	22
<b>15</b>	137	7	40	34
<b>16</b>	50	106	25	38
<b>17</b>	46	112	10	34
<b>18</b>	79	72	22	19
<b>19</b>	104	32	18	40
<b>20</b>	8	87	17	29
<b>21</b>	109	139	8	6
<b>22</b>	80	103	15	20
<b>23</b>	78	63	19	34
<b>24</b>	95	135	27	12
<b>25</b>	37	79	5	38
<b>26</b>	64	132	8	1
<b>27</b>	133	130	13	14
<b>28</b>	39	13	20	27
<b>29</b>	120	14	20	13
<b>30</b>	10	86	14	21
<b>31</b>	140	42	13	5
<b>32</b>	121	124	21	8
<b>33</b>	42	29	12	14
<b>34</b>	44	58	30	29
<b>35</b>	32	6	11	16
<b>36</b>	41	36	13	20
<b>37</b>	106	10	13	15
<b>38</b>	107	102	12	43
<b>39</b>	83	38	22	10
<b>40</b>	132	133	24	12
<b>41</b>	81	57	8	6
<b>42</b>	120	52	20	32
<b>43</b>	113	32	36	25
<b>44</b>	137	101	12	23
<b>45</b>	82	103	15	33
<b>46</b>	10	111	21	1

<b>47</b>	90	44	6	2
<b>48</b>	139	73	10	33
<b>49</b>	89	24	15	20
<b>50</b>	64	138	15	10
<b>51</b>	98	4	20	35
<b>52</b>	61	34	7	5
<b>53</b>	60	84	23	9
<b>54</b>	13	48	15	4
<b>55</b>	84	88	16	22
<b>56</b>	86	40	22	11
<b>57</b>	69	115	20	11
<b>58</b>	59	115	26	35
<b>59</b>	105	2	1	9
<b>60</b>	107	77	34	14
<b>61</b>	104	54	13	7
<b>62</b>	54	33	20	35
<b>63</b>	32	52	32	19
<b>64</b>	37	125	10	12
<b>65</b>	99	34	9	27
<b>66</b>	38	122	13	37
<b>67</b>	85	75	37	24
<b>68</b>	109	131	16	30
<b>69</b>	103	77	17	24
<b>70</b>	47	83	1	36
<b>71</b>	139	68	25	38
<b>72</b>	88	29	33	17
<b>73</b>	39	100	31	4
<b>74</b>	48	47	19	23
<b>75</b>	104	73	17	35
<b>76</b>	9	19	10	8
<b>77</b>	103	44	12	8
<b>78</b>	73	96	9	32
<b>79</b>	44	116	27	18
<b>80</b>	71	70	9	29
<b>81</b>	83	12	29	26
<b>82</b>	36	54	15	3
<b>83</b>	16	92	10	7
<b>84</b>	56	139	18	37
<b>85</b>	27	104	1	35
<b>86</b>	67	96	26	43

<b>87</b>	34	72	21	9
<b>88</b>	120	22	39	41
<b>89</b>	54	106	9	19
<b>90</b>	63	129	7	42
<b>91</b>	33	117	20	5
<b>92</b>	102	52	15	1
<b>93</b>	13	129	25	10
<b>94</b>	34	20	15	42
<b>95</b>	114	109	16	43
<b>96</b>	86	80	21	25
<b>97</b>	33	0	20	33
<b>98</b>	18	104	18	13
<b>99</b>	70	83	23	9
<b>100</b>	12	48	4	4
<b>101</b>	139	72	23	4
<b>102</b>	109	78	21	1
<b>103</b>	116	22	32	28
<b>104</b>	27	46	35	27
<b>105</b>	19	120	16	11
<b>106</b>	23	39	20	2
<b>107</b>	110	99	21	32
<b>108</b>	15	18	10	38
<b>109</b>	18	70	26	21
<b>110</b>	16	62	15	30
<b>111</b>	61	57	35	37
<b>112</b>	123	84	13	35
<b>113</b>	37	73	33	10
<b>114</b>	115	124	11	26
<b>115</b>	93	61	10	42
<b>116</b>	89	39	23	17
<b>117</b>	65	0	20	19
<b>118</b>	108	30	18	39
<b>119</b>	120	60	18	14
<b>120</b>	92	22	18	40
<b>121</b>	2	106	16	37
<b>122</b>	117	92	25	12
<b>123</b>	26	130	20	32
<b>124</b>	85	111	25	18
<b>125</b>	113	45	4	7
<b>126</b>	40	113	23	10

<b>127</b>	138	105	24	43
<b>128</b>	116	105	23	33
<b>129</b>	110	28	20	2
<b>130</b>	12	138	14	33
<b>131</b>	66	40	20	3
<b>132</b>	140	36	1	21
<b>133</b>	50	112	18	1
<b>134</b>	23	19	16	10
<b>135</b>	4	71	19	43
<b>136</b>	92	138	12	31
<b>137</b>	69	6	8	4
<b>138</b>	100	99	17	39
<b>139</b>	116	13	5	19
<b>140</b>	112	13	27	17
<b>141</b>	49	140	3	26
<b>142</b>	36	124	34	38
<b>143</b>	120	47	18	19
<b>144</b>	60	54	3	36
<b>145</b>	115	82	30	16
<b>146</b>	139	110	19	9
<b>147</b>	126	23	13	6
<b>148</b>	58	80	18	17
<b>149</b>	30	21	12	12
<b>150</b>	64	115	15	2
<b>151</b>	82	100	18	36
<b>152</b>	113	35	9	25
<b>153</b>	35	90	20	21
<b>154</b>	58	95	35	31
<b>155</b>	117	4	9	2
<b>156</b>	97	115	25	41
<b>157</b>	56	129	21	29
<b>158</b>	54	41	9	39
<b>159</b>	136	123	31	36
<b>160</b>	129	24	18	40
<b>161</b>	46	133	7	29
<b>162</b>	58	11	9	35
<b>163</b>	54	78	11	27
<b>164</b>	65	80	20	37
<b>165</b>	28	62	23	5
<b>166</b>	55	127	17	32

<b>167</b>	137	84	41	43
<b>168</b>	79	80	4	12
<b>169</b>	51	92	9	22
<b>170</b>	12	84	16	2
<b>171</b>	66	82	27	15
<b>172</b>	97	12	10	9
<b>173</b>	67	120	5	30
<b>174</b>	55	102	4	17
<b>175</b>	90	14	31	12
<b>176</b>	20	106	24	33
<b>177</b>	15	4	23	40
<b>178</b>	138	78	43	2
<b>179</b>	60	23	16	15
<b>180</b>	29	94	1	29
<b>181</b>	70	94	32	7
<b>182</b>	4	44	24	5
<b>183</b>	66	91	16	16
<b>184</b>	119	73	15	8
<b>185</b>	135	65	3	30
<b>186</b>	103	82	27	27
<b>187</b>	104	139	29	17
<b>188</b>	15	9	2	26
<b>189</b>	5	99	24	34
<b>190</b>	36	38	5	13
<b>191</b>	125	68	20	38
<b>192</b>	44	36	9	21
<b>193</b>	137	127	28	36
<b>194</b>	120	102	18	31
<b>195</b>	138	24	12	10
<b>196</b>	103	86	19	14
<b>197</b>	137	96	9	37
<b>198</b>	8	138	10	11
<b>199</b>	81	36	17	12
<b>200</b>	116	83	30	34

### Instancia C121

Nro Clientes	Capacidad vehículos
--------------	---------------------

	200	200		
	<i>Cord X</i>	<i>Cord Y</i>	<i>Entrega</i>	<i>Recogida</i>
<b>Depósito</b>	70	70	0	0
<b>1</b>	33	78	20	30
<b>2</b>	59	52	20	10
<b>3</b>	10	137	30	30
<b>4</b>	4	28	10	10
<b>5</b>	25	26	20	20
<b>6</b>	86	37	10	40
<b>7</b>	1	109	10	20
<b>8</b>	6	135	40	10
<b>9</b>	32	79	20	30
<b>10</b>	24	26	20	40
<b>11</b>	86	36	20	40
<b>12</b>	95	35	10	30
<b>13</b>	63	50	10	10
<b>14</b>	100	106	10	20
<b>15</b>	99	112	20	20
<b>16</b>	36	135	10	30
<b>17</b>	57	59	10	30
<b>18</b>	8	124	10	40
<b>19</b>	85	106	20	10
<b>20</b>	103	69	30	20
<b>21</b>	109	131	20	40
<b>22</b>	43	140	10	10
<b>23</b>	115	134	30	20
<b>24</b>	98	70	10	40
<b>25</b>	112	67	10	10
<b>26</b>	102	104	20	40
<b>27</b>	93	75	30	30
<b>28</b>	90	104	20	10
<b>29</b>	127	108	10	30
<b>30</b>	84	99	20	40
<b>31</b>	113	69	20	20
<b>32</b>	129	9	10	30
<b>33</b>	18	38	30	20
<b>34</b>	30	27	10	40
<b>35</b>	25	80	20	30
<b>36</b>	17	37	30	40
<b>37</b>	32	106	10	20

<b>38</b>	43	135	10	20
<b>39</b>	61	59	20	40
<b>40</b>	104	106	20	30
<b>41</b>	109	71	20	30
<b>42</b>	121	110	30	10
<b>43</b>	61	48	20	40
<b>44</b>	74	99	20	40
<b>45</b>	89	73	10	10
<b>46</b>	21	25	20	40
<b>47</b>	99	28	30	20
<b>48</b>	101	96	10	20
<b>49</b>	9	114	20	10
<b>50</b>	121	112	20	20
<b>51</b>	137	6	10	10
<b>52</b>	118	131	20	30
<b>53</b>	34	82	10	30
<b>54</b>	4	125	30	20
<b>55</b>	26	105	40	10
<b>56</b>	35	123	20	20
<b>57</b>	21	48	10	20
<b>58</b>	21	115	10	30
<b>59</b>	99	108	20	40
<b>60</b>	1	34	20	20
<b>61</b>	94	70	20	10
<b>62</b>	74	93	20	40
<b>63</b>	32	128	20	20
<b>64</b>	94	73	10	10
<b>65</b>	135	3	20	30
<b>66</b>	97	28	20	20
<b>67</b>	59	56	20	30
<b>68</b>	80	97	10	40
<b>69</b>	134	31	20	20
<b>70</b>	103	34	30	20
<b>71</b>	30	79	20	20
<b>72</b>	4	30	40	30
<b>73</b>	87	39	20	20
<b>74</b>	90	101	10	40
<b>75</b>	119	136	30	30
<b>76</b>	78	92	30	40
<b>77</b>	110	66	10	30

<b>78</b>	62	58	20	30
<b>79</b>	127	112	20	40
<b>80</b>	113	71	10	10
<b>81</b>	34	104	10	40
<b>82</b>	2	28	10	40
<b>83</b>	17	43	10	20
<b>84</b>	1	24	10	30
<b>85</b>	112	71	20	20
<b>86</b>	135	0	10	30
<b>87</b>	125	108	20	30
<b>88</b>	8	138	30	20
<b>89</b>	103	111	20	30
<b>90</b>	57	53	10	40
<b>91</b>	98	31	20	40
<b>92</b>	116	132	10	30
<b>93</b>	28	99	10	10
<b>94</b>	137	3	20	10
<b>95</b>	29	28	10	20
<b>96</b>	94	105	10	20
<b>97</b>	92	107	30	10
<b>98</b>	10	134	10	20
<b>99</b>	34	79	10	30
<b>100</b>	94	72	20	30
<b>101</b>	35	76	10	30
<b>102</b>	76	99	20	40
<b>103</b>	127	28	30	20
<b>104</b>	8	119	20	20
<b>105</b>	102	110	10	30
<b>106</b>	24	24	10	30
<b>107</b>	64	61	20	10
<b>108</b>	22	38	20	10
<b>109</b>	88	72	30	10
<b>110</b>	107	63	30	40
<b>111</b>	33	131	30	10
<b>112</b>	123	112	20	20
<b>113</b>	67	55	10	10
<b>114</b>	39	128	10	40
<b>115</b>	136	0	30	30
<b>116</b>	93	36	20	40
<b>117</b>	7	115	10	10

<b>118</b>	21	44	20	20
<b>119</b>	28	74	20	40
<b>120</b>	86	102	10	30
<b>121</b>	19	37	20	40
<b>122</b>	84	36	10	40
<b>123</b>	118	108	10	10
<b>124</b>	132	24	10	40
<b>125</b>	6	25	10	20
<b>126</b>	28	78	10	10
<b>127</b>	2	132	10	40
<b>128</b>	22	22	10	30
<b>129</b>	88	35	20	10
<b>130</b>	94	104	10	20
<b>131</b>	74	96	30	30
<b>132</b>	0	117	30	20
<b>133</b>	99	96	10	10
<b>134</b>	119	114	20	40
<b>135</b>	28	107	20	40
<b>136</b>	133	12	10	10
<b>137</b>	35	102	10	20
<b>138</b>	36	105	20	30
<b>139</b>	84	39	20	30
<b>140</b>	34	138	20	30
<b>141</b>	136	29	10	40
<b>142</b>	33	133	20	20
<b>143</b>	16	39	20	40
<b>144</b>	32	77	20	40
<b>145</b>	122	131	20	20
<b>146</b>	77	100	20	40
<b>147</b>	96	26	10	30
<b>148</b>	127	29	20	30
<b>149</b>	91	95	20	40
<b>150</b>	42	136	10	20
<b>151</b>	36	132	10	40
<b>152</b>	103	106	30	40
<b>153</b>	107	105	10	10
<b>154</b>	96	74	10	30
<b>155</b>	63	57	10	40
<b>156</b>	122	113	20	30
<b>157</b>	14	131	10	20

<b>158</b>	30	31	20	20
<b>159</b>	40	132	20	40
<b>160</b>	99	26	40	10
<b>161</b>	10	118	10	40
<b>162</b>	102	60	30	10
<b>163</b>	120	136	10	30
<b>164</b>	97	29	30	20
<b>165</b>	19	39	10	40
<b>166</b>	27	77	10	20
<b>167</b>	30	24	20	20
<b>168</b>	126	112	10	40
<b>169</b>	105	110	20	20
<b>170</b>	118	114	10	30
<b>171</b>	135	6	30	40
<b>172</b>	111	66	20	20
<b>173</b>	95	77	10	30
<b>174</b>	139	10	30	40
<b>175</b>	61	54	10	30
<b>176</b>	16	38	20	40
<b>177</b>	16	138	10	10
<b>178</b>	91	78	10	40
<b>179</b>	91	71	20	40
<b>180</b>	0	27	30	40
<b>181</b>	6	113	20	40
<b>182</b>	119	139	20	30
<b>183</b>	35	99	20	40
<b>184</b>	31	112	20	40
<b>185</b>	5	119	20	20
<b>186</b>	3	135	20	30
<b>187</b>	30	136	10	40
<b>188</b>	20	39	10	10
<b>189</b>	131	11	10	10
<b>190</b>	28	33	20	40
<b>191</b>	4	23	10	30
<b>192</b>	87	108	20	10
<b>193</b>	21	26	20	40
<b>194</b>	121	134	20	40
<b>195</b>	119	130	20	30
<b>196</b>	91	108	10	20
<b>197</b>	130	27	30	40

<b>198</b>	101	107	10	20
<b>199</b>	34	108	10	20
<b>200</b>	131	31	20	40

### Instancia C221

	Nro Clientes	Capacidad vehículos		
	200	700		
	<i>Cord X</i>	<i>Cord Y</i>	<i>Entrega</i>	<i>Recogida</i>
<b>Depósito</b>	70	70	0	0
<b>1</b>	38	105	20	30
<b>2</b>	100	5	20	10
<b>3</b>	111	78	10	10
<b>4</b>	96	62	10	30
<b>5</b>	122	11	30	20
<b>6</b>	94	18	30	40
<b>7</b>	60	89	10	50
<b>8</b>	67	123	20	50
<b>9</b>	99	53	20	50
<b>10</b>	26	24	20	20
<b>11</b>	93	61	10	30
<b>12</b>	27	32	20	20
<b>13</b>	25	31	20	20
<b>14</b>	98	36	10	30
<b>15</b>	101	68	10	30
<b>16</b>	49	48	10	10
<b>17</b>	56	95	10	40
<b>18</b>	14	107	20	20
<b>19</b>	98	46	20	20
<b>20</b>	131	4	40	30
<b>21</b>	94	89	20	40
<b>22</b>	70	113	30	40
<b>23</b>	94	59	30	30
<b>24</b>	64	59	10	20
<b>25</b>	112	3	10	10
<b>26</b>	20	108	10	10
<b>27</b>	10	118	20	30
<b>28</b>	76	65	40	50

<b>29</b>	103	12	20	40
<b>30</b>	44	62	30	10
<b>31</b>	96	138	10	20
<b>32</b>	27	44	20	10
<b>33</b>	21	28	40	20
<b>34</b>	26	91	10	50
<b>35</b>	98	75	10	40
<b>36</b>	69	48	20	50
<b>37</b>	19	116	10	10
<b>38</b>	60	111	10	40
<b>39</b>	54	102	20	20
<b>40</b>	63	103	20	50
<b>41</b>	91	91	10	40
<b>42</b>	77	15	10	30
<b>43</b>	77	117	30	20
<b>44</b>	115	41	20	10
<b>45</b>	12	33	10	10
<b>46</b>	44	82	20	40
<b>47</b>	117	55	20	50
<b>48</b>	116	10	20	30
<b>49</b>	57	77	20	30
<b>50</b>	29	36	20	20
<b>51</b>	90	91	10	10
<b>52</b>	62	73	20	10
<b>53</b>	6	121	20	20
<b>54</b>	105	49	10	10
<b>55</b>	28	115	10	20
<b>56</b>	4	137	30	30
<b>57</b>	11	100	20	50
<b>58</b>	67	84	30	40
<b>59</b>	23	22	20	10
<b>60</b>	41	64	20	50
<b>61</b>	56	56	40	20
<b>62</b>	80	40	20	10
<b>63</b>	60	66	30	40
<b>64</b>	16	90	20	40
<b>65</b>	104	82	30	30
<b>66</b>	64	78	10	10
<b>67</b>	47	36	20	30
<b>68</b>	46	80	20	10

<b>69</b>	86	124	20	30
<b>70</b>	48	72	20	10
<b>71</b>	121	9	10	30
<b>72</b>	60	52	20	40
<b>73</b>	25	46	20	50
<b>74</b>	26	15	20	50
<b>75</b>	98	116	30	30
<b>76</b>	66	131	10	50
<b>77</b>	15	30	20	50
<b>78</b>	66	117	30	50
<b>79</b>	111	70	20	20
<b>80</b>	123	19	20	40
<b>81</b>	17	18	30	50
<b>82</b>	57	87	10	10
<b>83</b>	102	33	30	30
<b>84</b>	7	113	10	40
<b>85</b>	37	115	20	20
<b>86</b>	17	25	10	30
<b>87</b>	83	105	30	20
<b>88</b>	51	115	10	40
<b>89</b>	96	55	20	40
<b>90</b>	112	2	20	40
<b>91</b>	116	23	20	50
<b>92</b>	57	70	10	20
<b>93</b>	15	23	30	40
<b>94</b>	45	68	30	10
<b>95</b>	19	41	10	30
<b>96</b>	38	16	10	40
<b>97</b>	81	135	20	30
<b>98</b>	62	128	20	40
<b>99</b>	49	70	20	30
<b>100</b>	24	19	10	10
<b>101</b>	119	20	10	40
<b>102</b>	62	125	30	40
<b>103</b>	29	10	30	20
<b>104</b>	13	80	30	10
<b>105</b>	65	73	10	10
<b>106</b>	109	25	10	20
<b>107</b>	91	71	10	40
<b>108</b>	47	57	30	50

<b>109</b>	9	42	20	30
<b>110</b>	96	23	30	10
<b>111</b>	26	102	40	20
<b>112</b>	4	15	20	20
<b>113</b>	3	98	10	50
<b>114</b>	35	88	20	50
<b>115</b>	18	107	10	10
<b>116</b>	95	68	20	10
<b>117</b>	58	85	10	40
<b>118</b>	11	23	10	50
<b>119</b>	61	107	10	50
<b>120</b>	3	17	10	20
<b>121</b>	26	104	10	10
<b>122</b>	74	56	10	30
<b>123</b>	24	105	10	20
<b>124</b>	30	106	30	30
<b>125</b>	4	33	20	40
<b>126</b>	106	0	10	50
<b>127</b>	120	10	20	50
<b>128</b>	106	5	30	10
<b>129</b>	30	58	20	20
<b>130</b>	9	9	10	10
<b>131</b>	7	92	20	30
<b>132</b>	50	29	10	50
<b>133</b>	46	52	10	10
<b>134</b>	70	110	30	50
<b>135</b>	37	111	20	10
<b>136</b>	75	64	20	50
<b>137</b>	20	93	10	40
<b>138</b>	94	1	10	50
<b>139</b>	41	12	10	30
<b>140</b>	20	109	10	40
<b>141</b>	28	14	20	10
<b>142</b>	87	63	30	40
<b>143</b>	101	4	10	30
<b>144</b>	80	22	10	20
<b>145</b>	117	10	10	20
<b>146</b>	103	33	10	30
<b>147</b>	104	51	20	50
<b>148</b>	103	19	30	20

<b>149</b>	7	52	20	20
<b>150</b>	73	119	30	50
<b>151</b>	79	128	30	20
<b>152</b>	10	50	20	50
<b>153</b>	26	23	50	40
<b>154</b>	95	103	20	30
<b>155</b>	101	59	30	50
<b>156</b>	113	67	20	10
<b>157</b>	25	94	10	20
<b>158</b>	120	20	20	30
<b>159</b>	105	26	30	40
<b>160</b>	59	77	20	50
<b>161</b>	116	34	20	10
<b>162</b>	45	71	10	10
<b>163</b>	86	109	10	40
<b>164</b>	15	96	10	50
<b>165</b>	64	88	30	30
<b>166</b>	60	115	20	40
<b>167</b>	63	100	10	20
<b>168</b>	94	104	30	40
<b>169</b>	8	102	10	10
<b>170</b>	64	89	10	50
<b>171</b>	61	66	10	30
<b>172</b>	95	121	10	40
<b>173</b>	47	63	30	40
<b>174</b>	19	96	10	40
<b>175</b>	103	55	20	20
<b>176</b>	36	94	30	40
<b>177</b>	75	103	20	50
<b>178</b>	65	80	10	20
<b>179</b>	94	56	30	50
<b>180</b>	18	123	20	20
<b>181</b>	92	123	10	40
<b>182</b>	12	31	20	10
<b>183</b>	106	81	20	50
<b>184</b>	98	28	10	40
<b>185</b>	14	133	30	10
<b>186</b>	101	32	20	10
<b>187</b>	25	66	10	10
<b>188</b>	78	61	40	40

<b>189</b>	123	64	20	30
<b>190</b>	126	20	20	50
<b>191</b>	72	117	10	30
<b>192</b>	10	105	20	40
<b>193</b>	128	0	10	10
<b>194</b>	130	16	30	40
<b>195</b>	114	63	10	20
<b>196</b>	80	63	20	40
<b>197</b>	113	36	10	40
<b>198</b>	15	94	30	30
<b>199</b>	129	77	10	20
<b>200</b>	85	85	20	10

### Instancia RC121

	<b>Nro Clientes</b>	<b>Capacidad vehículos</b>		
	200	200		
	<b>Cord X</b>	<b>Cord Y</b>	<b>Entrega</b>	<b>Recogida</b>
<b>Depósito</b>	70	70	0	0
<b>1</b>	10	86	14	19
<b>2</b>	16	62	15	24
<b>3</b>	84	36	10	13
<b>4</b>	79	72	22	39
<b>5</b>	16	39	20	1
<b>6</b>	61	54	10	4
<b>7</b>	96	26	10	18
<b>8</b>	16	92	10	6
<b>9</b>	85	106	20	27
<b>10</b>	6	25	10	29
<b>11</b>	37	125	10	37
<b>12</b>	87	108	20	21
<b>13</b>	18	104	18	31
<b>14</b>	61	48	20	28
<b>15</b>	97	29	30	35
<b>16</b>	112	67	10	14
<b>17</b>	50	106	25	35
<b>18</b>	99	28	30	37
<b>19</b>	15	9	2	5

<b>20</b>	19	37	20	37
<b>21</b>	90	101	10	40
<b>22</b>	125	68	20	39
<b>23</b>	127	108	10	5
<b>24</b>	90	44	6	27
<b>25</b>	4	23	10	39
<b>26</b>	17	37	30	9
<b>27</b>	113	32	36	7
<b>28</b>	63	57	10	16
<b>29</b>	94	104	10	22
<b>30</b>	58	95	35	28
<b>31</b>	137	96	9	7
<b>32</b>	127	112	20	30
<b>33</b>	103	119	18	5
<b>34</b>	20	69	13	34
<b>35</b>	2	28	10	25
<b>36</b>	88	29	33	19
<b>37</b>	86	102	10	3
<b>38</b>	126	112	10	15
<b>39</b>	61	34	7	35
<b>40</b>	113	69	20	23
<b>41</b>	109	78	21	13
<b>42</b>	21	48	10	8
<b>43</b>	86	36	20	36
<b>44</b>	97	12	10	35
<b>45</b>	121	112	20	33
<b>46</b>	19	39	10	40
<b>47</b>	66	40	20	27
<b>48</b>	87	39	20	41
<b>49</b>	63	50	10	31
<b>50</b>	121	123	9	9
<b>51</b>	116	22	32	31
<b>52</b>	137	101	12	27
<b>53</b>	137	137	27	39
<b>54</b>	70	94	32	37
<b>55</b>	117	92	25	35
<b>56</b>	122	113	20	22
<b>57</b>	109	71	20	43
<b>58</b>	35	102	10	33
<b>59</b>	91	108	10	17

<b>60</b>	8	87	17	32
<b>61</b>	140	42	13	15
<b>62</b>	16	38	20	8
<b>63</b>	44	36	9	27
<b>64</b>	60	84	23	42
<b>65</b>	109	139	8	19
<b>66</b>	80	103	15	28
<b>67</b>	66	82	27	19
<b>68</b>	99	26	40	40
<b>69</b>	34	108	10	30
<b>70</b>	135	65	3	30
<b>71</b>	36	124	34	38
<b>72</b>	102	60	30	21
<b>73</b>	30	21	12	27
<b>74</b>	139	110	19	18
<b>75</b>	10	111	21	24
<b>76</b>	18	70	26	17
<b>77</b>	4	28	10	18
<b>78</b>	57	53	10	3
<b>79</b>	54	41	9	41
<b>80</b>	107	25	13	6
<b>81</b>	110	99	21	22
<b>82</b>	84	88	16	18
<b>83</b>	101	0	9	12
<b>84</b>	28	99	10	39
<b>85</b>	119	114	20	28
<b>86</b>	118	108	10	2
<b>87</b>	50	112	18	40
<b>88</b>	44	116	27	8
<b>89</b>	82	100	18	20
<b>90</b>	83	12	29	34
<b>91</b>	126	23	13	4
<b>92</b>	85	75	37	29
<b>93</b>	39	13	20	34
<b>94</b>	120	136	10	40
<b>95</b>	105	2	1	12
<b>96</b>	36	54	15	35
<b>97</b>	35	90	20	1
<b>98</b>	111	66	20	23
<b>99</b>	119	136	30	1

<b>100</b>	84	99	20	22
<b>101</b>	137	127	28	6
<b>102</b>	95	35	10	33
<b>103</b>	69	6	8	28
<b>104</b>	1	24	10	19
<b>105</b>	31	112	20	26
<b>106</b>	138	78	43	28
<b>107</b>	100	99	17	12
<b>108</b>	123	112	20	30
<b>109</b>	119	130	20	42
<b>110</b>	92	22	18	40
<b>111</b>	137	7	40	2
<b>112</b>	121	124	21	28
<b>113</b>	34	104	10	38
<b>114</b>	121	110	30	21
<b>115</b>	56	139	18	19
<b>116</b>	106	10	13	22
<b>117</b>	125	108	20	37
<b>118</b>	21	115	10	21
<b>119</b>	110	66	10	14
<b>120</b>	36	38	5	29
<b>121</b>	56	129	21	6
<b>122</b>	4	30	40	36
<b>123</b>	8	138	10	34
<b>124</b>	88	35	20	5
<b>125</b>	116	132	10	18
<b>126</b>	113	45	4	38
<b>127</b>	28	107	20	1
<b>128</b>	115	134	30	11
<b>129</b>	116	83	30	8
<b>130</b>	57	59	10	18
<b>131</b>	89	39	23	43
<b>132</b>	59	52	20	4
<b>133</b>	112	71	20	4
<b>134</b>	35	99	20	20
<b>135</b>	23	39	20	19
<b>136</b>	84	39	20	18
<b>137</b>	19	120	16	17
<b>138</b>	20	39	10	8
<b>139</b>	90	104	20	11

<b>140</b>	66	91	16	27
<b>141</b>	4	44	24	30
<b>142</b>	78	63	19	9
<b>143</b>	92	107	30	2
<b>144</b>	37	138	5	18
<b>145</b>	120	14	20	12
<b>146</b>	33	0	20	37
<b>147</b>	113	71	10	21
<b>148</b>	112	13	27	4
<b>149</b>	41	36	13	33
<b>150</b>	64	61	20	20
<b>151</b>	121	134	20	13
<b>152</b>	103	44	12	26
<b>153</b>	67	55	10	3
<b>154</b>	86	80	21	20
<b>155</b>	64	132	8	3
<b>156</b>	93	36	20	16
<b>157</b>	115	124	11	30
<b>158</b>	103	82	27	36
<b>159</b>	18	38	30	28
<b>160</b>	122	131	20	25
<b>161</b>	107	63	30	42
<b>162</b>	48	47	19	35
<b>163</b>	62	58	20	31
<b>164</b>	0	27	30	24
<b>165</b>	118	114	10	32
<b>166</b>	91	95	20	39
<b>167</b>	113	35	9	39
<b>168</b>	119	139	20	5
<b>169</b>	104	54	13	17
<b>170</b>	15	4	23	13
<b>171</b>	59	56	20	39
<b>172</b>	21	44	20	18
<b>173</b>	100	106	10	34
<b>174</b>	47	83	1	11
<b>175</b>	34	72	21	37
<b>176</b>	32	106	10	19
<b>177</b>	109	131	20	34
<b>178</b>	89	24	15	24
<b>179</b>	97	28	20	11

<b>180</b>	86	37	10	8
<b>181</b>	1	34	20	41
<b>182</b>	103	69	30	22
<b>183</b>	98	31	20	18
<b>184</b>	103	34	30	10
<b>185</b>	13	48	15	20
<b>186</b>	22	38	20	8
<b>187</b>	61	59	20	25
<b>188</b>	94	105	10	6
<b>189</b>	118	131	20	2
<b>190</b>	108	30	18	14
<b>191</b>	36	105	20	33
<b>192</b>	133	130	13	6
<b>193</b>	15	18	10	4
<b>194</b>	103	86	19	30
<b>195</b>	107	77	34	2
<b>196</b>	99	34	9	28
<b>197</b>	17	43	10	28
<b>198</b>	140	36	1	23
<b>199</b>	26	105	40	2
<b>200</b>	60	23	16	40

### Instancia RC221

	Nro Clientes	Capacidad vehículos		
	200	1000		
	<i>Cord X</i>	<i>Cord Y</i>	<i>Entrega</i>	<i>Recogida</i>
<b>Depósito</b>	10	86	14	39
<b>1</b>	16	62	15	35
<b>2</b>	84	36	10	38
<b>3</b>	79	72	22	28
<b>4</b>	16	39	20	38
<b>5</b>	61	54	10	11
<b>6</b>	96	26	10	7
<b>7</b>	16	92	10	12
<b>8</b>	85	106	20	21
<b>9</b>	6	25	10	37
<b>10</b>	37	125	10	3

<b>11</b>	87	108	20	1
<b>12</b>	18	104	18	20
<b>13</b>	61	48	20	31
<b>14</b>	97	29	30	22
<b>15</b>	112	67	10	41
<b>16</b>	50	106	25	17
<b>17</b>	99	28	30	26
<b>18</b>	15	9	2	42
<b>19</b>	19	37	20	23
<b>20</b>	90	101	10	2
<b>21</b>	125	68	20	1
<b>22</b>	127	108	10	37
<b>23</b>	90	44	6	34
<b>24</b>	4	23	10	11
<b>25</b>	17	37	30	10
<b>26</b>	113	32	36	1
<b>27</b>	63	57	10	3
<b>28</b>	94	104	10	16
<b>29</b>	58	95	35	41
<b>30</b>	137	96	9	25
<b>31</b>	127	112	20	20
<b>32</b>	103	119	18	23
<b>33</b>	20	69	13	9
<b>34</b>	2	28	10	43
<b>35</b>	88	29	33	9
<b>36</b>	86	102	10	24
<b>37</b>	126	112	10	29
<b>38</b>	61	34	7	2
<b>39</b>	113	69	20	35
<b>40</b>	109	78	21	24
<b>41</b>	21	48	10	21
<b>42</b>	86	36	20	6
<b>43</b>	97	12	10	28
<b>44</b>	121	112	20	26
<b>45</b>	19	39	10	2
<b>46</b>	66	40	20	43
<b>47</b>	87	39	20	6
<b>48</b>	63	50	10	37
<b>49</b>	121	123	9	27
<b>50</b>	116	22	32	3

<b>51</b>	137	101	12	38
<b>52</b>	137	137	27	9
<b>53</b>	70	94	32	32
<b>54</b>	117	92	25	24
<b>55</b>	122	113	20	21
<b>56</b>	109	71	20	42
<b>57</b>	35	102	10	14
<b>58</b>	91	108	10	23
<b>59</b>	8	87	17	29
<b>60</b>	140	42	13	36
<b>61</b>	16	38	20	31
<b>62</b>	44	36	9	6
<b>63</b>	60	84	23	4
<b>64</b>	109	139	8	17
<b>65</b>	80	103	15	18
<b>66</b>	66	82	27	40
<b>67</b>	99	26	40	37
<b>68</b>	34	108	10	12
<b>69</b>	135	65	3	20
<b>70</b>	36	124	34	4
<b>71</b>	102	60	30	37
<b>72</b>	30	21	12	40
<b>73</b>	139	110	19	31
<b>74</b>	10	111	21	27
<b>75</b>	18	70	26	4
<b>76</b>	4	28	10	8
<b>77</b>	57	53	10	27
<b>78</b>	54	41	9	10
<b>79</b>	107	25	13	37
<b>80</b>	110	99	21	15
<b>81</b>	84	88	16	3
<b>82</b>	101	0	9	7
<b>83</b>	28	99	10	23
<b>84</b>	119	114	20	14
<b>85</b>	118	108	10	13
<b>86</b>	50	112	18	10
<b>87</b>	44	116	27	41
<b>88</b>	82	100	18	15
<b>89</b>	83	12	29	12
<b>90</b>	126	23	13	8

<b>91</b>	85	75	37	39
<b>92</b>	39	13	20	36
<b>93</b>	120	136	10	37
<b>94</b>	105	2	1	27
<b>95</b>	36	54	15	16
<b>96</b>	35	90	20	35
<b>97</b>	111	66	20	19
<b>98</b>	119	136	30	13
<b>99</b>	84	99	20	10
<b>100</b>	137	127	28	37
<b>101</b>	95	35	10	14
<b>102</b>	69	6	8	36
<b>103</b>	1	24	10	25
<b>104</b>	31	112	20	17
<b>105</b>	138	78	43	35
<b>106</b>	100	99	17	26
<b>107</b>	123	112	20	2
<b>108</b>	119	130	20	31
<b>109</b>	92	22	18	16
<b>110</b>	137	7	40	38
<b>111</b>	121	124	21	38
<b>112</b>	34	104	10	18
<b>113</b>	121	110	30	13
<b>114</b>	56	139	18	5
<b>115</b>	106	10	13	11
<b>116</b>	125	108	20	42
<b>117</b>	21	115	10	32
<b>118</b>	110	66	10	1
<b>119</b>	36	38	5	16
<b>120</b>	56	129	21	6
<b>121</b>	4	30	40	24
<b>122</b>	8	138	10	2
<b>123</b>	88	35	20	16
<b>124</b>	116	132	10	11
<b>125</b>	113	45	4	18
<b>126</b>	28	107	20	3
<b>127</b>	115	134	30	36
<b>128</b>	116	83	30	6
<b>129</b>	57	59	10	26
<b>130</b>	89	39	23	31

<b>131</b>	59	52	20	30
<b>132</b>	112	71	20	2
<b>133</b>	35	99	20	8
<b>134</b>	23	39	20	24
<b>135</b>	84	39	20	29
<b>136</b>	19	120	16	12
<b>137</b>	20	39	10	38
<b>138</b>	90	104	20	26
<b>139</b>	66	91	16	23
<b>140</b>	4	44	24	1
<b>141</b>	78	63	19	40
<b>142</b>	92	107	30	35
<b>143</b>	37	138	5	40
<b>144</b>	120	14	20	34
<b>145</b>	33	0	20	4
<b>146</b>	113	71	10	34
<b>147</b>	112	13	27	22
<b>148</b>	41	36	13	21
<b>149</b>	64	61	20	12
<b>150</b>	121	134	20	35
<b>151</b>	103	44	12	20
<b>152</b>	67	55	10	33
<b>153</b>	86	80	21	20
<b>154</b>	64	132	8	25
<b>155</b>	93	36	20	41
<b>156</b>	115	124	11	17
<b>157</b>	103	82	27	2
<b>158</b>	18	38	30	29
<b>159</b>	122	131	20	42
<b>160</b>	107	63	30	38
<b>161</b>	48	47	19	27
<b>162</b>	62	58	20	32
<b>163</b>	0	27	30	15
<b>164</b>	118	114	10	15
<b>165</b>	91	95	20	1
<b>166</b>	113	35	9	9
<b>167</b>	119	139	20	15
<b>168</b>	104	54	13	23
<b>169</b>	15	4	23	41
<b>170</b>	59	56	20	26

<b>171</b>	21	44	20	34
<b>172</b>	100	106	10	12
<b>173</b>	47	83	1	22
<b>174</b>	34	72	21	6
<b>175</b>	32	106	10	35
<b>176</b>	109	131	20	8
<b>177</b>	89	24	15	28
<b>178</b>	97	28	20	41
<b>179</b>	86	37	10	25
<b>180</b>	1	34	20	3
<b>181</b>	103	69	30	11
<b>182</b>	98	31	20	7
<b>183</b>	103	34	30	19
<b>184</b>	13	48	15	7
<b>185</b>	22	38	20	10
<b>186</b>	61	59	20	34
<b>187</b>	94	105	10	35
<b>188</b>	118	131	20	35
<b>189</b>	108	30	18	3
<b>190</b>	36	105	20	39
<b>191</b>	133	130	13	15
<b>192</b>	15	18	10	24
<b>193</b>	103	86	19	28
<b>194</b>	107	77	34	38
<b>195</b>	99	34	9	5
<b>196</b>	17	43	10	6
<b>197</b>	140	36	1	1
<b>198</b>	26	105	40	20
<b>199</b>	60	23	16	14
<b>200</b>	60	23	16	40

### Instancia R141

	<b>Nro Clientes</b>	<b>Capacidad vehículos</b>		
	400	200		
	<b>Cord X</b>	<b>Cord Y</b>	<b>Entrega</b>	<b>Recogida</b>
<b>Depósito</b>	100	100	0	0
<b>1</b>	20	59	20	48

<b>2</b>	178	61	19	14
<b>3</b>	39	186	5	35
<b>4</b>	182	110	15	8
<b>5</b>	186	65	39	8
<b>6</b>	58	19	10	44
<b>7</b>	60	71	22	32
<b>8</b>	86	150	22	47
<b>9</b>	40	95	25	36
<b>10</b>	123	90	21	23
<b>11</b>	55	100	22	31
<b>12</b>	18	35	24	45
<b>13</b>	88	152	13	48
<b>14</b>	26	41	14	28
<b>15</b>	24	131	31	6
<b>16</b>	97	192	23	33
<b>17</b>	63	44	14	12
<b>18</b>	1	54	14	5
<b>19</b>	134	157	20	12
<b>20</b>	176	132	10	9
<b>21</b>	155	17	32	39
<b>22</b>	159	187	9	37
<b>23</b>	174	194	8	17
<b>24</b>	136	90	18	23
<b>25</b>	1	149	35	26
<b>26</b>	83	180	15	10
<b>27</b>	63	141	14	14
<b>28</b>	39	112	26	33
<b>29</b>	184	140	15	31
<b>30</b>	50	31	20	17
<b>31</b>	28	160	27	36
<b>32</b>	143	179	24	45
<b>33</b>	74	163	21	33
<b>34</b>	39	151	26	19
<b>35</b>	171	69	19	9
<b>36</b>	153	19	3	32
<b>37</b>	111	189	3	28
<b>38</b>	5	122	15	1
<b>39</b>	159	105	20	28
<b>40</b>	49	99	32	12
<b>41</b>	53	175	28	1

<b>42</b>	48	106	23	42
<b>43</b>	101	127	31	26
<b>44</b>	149	78	6	47
<b>45</b>	78	169	10	45
<b>46</b>	157	142	13	38
<b>47</b>	75	170	17	12
<b>48</b>	156	93	25	37
<b>49</b>	5	130	15	28
<b>50</b>	158	81	23	37
<b>51</b>	85	85	14	19
<b>52</b>	189	83	43	17
<b>53</b>	66	3	9	29
<b>54</b>	45	95	15	46
<b>55</b>	162	162	20	4
<b>56</b>	84	188	17	25
<b>57</b>	38	101	10	48
<b>58</b>	55	51	17	10
<b>59</b>	176	58	21	11
<b>60</b>	48	94	15	39
<b>61</b>	173	147	19	40
<b>62</b>	35	9	29	32
<b>63</b>	112	42	33	6
<b>64</b>	11	118	33	42
<b>65</b>	155	22	24	22
<b>66</b>	42	174	13	41
<b>67</b>	154	7	32	46
<b>68</b>	181	39	11	48
<b>69</b>	60	153	11	5
<b>70</b>	171	36	6	46
<b>71</b>	158	132	12	33
<b>72</b>	112	174	20	6
<b>73</b>	97	129	11	15
<b>74</b>	68	191	15	6
<b>75</b>	124	191	24	25
<b>76</b>	86	110	18	33
<b>77</b>	158	20	19	32
<b>78</b>	130	55	28	24
<b>79</b>	180	66	13	31
<b>80</b>	18	59	20	21
<b>81</b>	139	158	9	33

<b>82</b>	37	146	10	38
<b>83</b>	17	32	16	19
<b>84</b>	86	69	14	14
<b>85</b>	67	70	18	46
<b>86</b>	119	113	35	38
<b>87</b>	136	12	21	29
<b>88</b>	49	71	36	21
<b>89</b>	121	48	17	28
<b>90</b>	118	117	20	15
<b>91</b>	104	88	25	48
<b>92</b>	89	176	13	22
<b>93</b>	46	40	15	37
<b>94</b>	78	72	9	48
<b>95</b>	69	150	17	44
<b>96</b>	6	120	25	11
<b>97</b>	154	29	19	11
<b>98</b>	70	130	8	29
<b>99</b>	172	186	35	35
<b>100</b>	23	168	13	37
<b>101</b>	65	42	12	19
<b>102</b>	53	180	4	11
<b>103</b>	90	4	22	22
<b>104</b>	76	102	15	14
<b>105</b>	30	67	7	39
<b>106</b>	138	72	8	25
<b>107</b>	68	115	20	46
<b>108</b>	84	81	11	17
<b>109</b>	104	142	24	12
<b>110</b>	104	5	26	24
<b>111</b>	31	63	11	10
<b>112</b>	30	183	9	22
<b>113</b>	141	146	15	17
<b>114</b>	159	196	6	33
<b>115</b>	20	72	21	37
<b>116</b>	113	100	5	38
<b>117</b>	184	101	17	37
<b>118</b>	82	48	8	12
<b>119</b>	57	157	26	30
<b>120</b>	67	138	11	47
<b>121</b>	181	70	25	7

<b>122</b>	63	149	17	6
<b>123</b>	41	43	30	25
<b>124</b>	74	67	2	41
<b>125</b>	138	61	19	28
<b>126</b>	182	9	30	17
<b>127</b>	104	176	19	48
<b>128</b>	60	184	7	33
<b>129</b>	63	192	12	12
<b>130</b>	77	119	17	12
<b>131</b>	137	41	15	14
<b>132</b>	151	180	14	23
<b>133</b>	63	151	22	6
<b>134</b>	146	3	24	35
<b>135</b>	186	8	31	15
<b>136</b>	51	162	20	7
<b>137</b>	57	118	21	25
<b>138</b>	107	96	23	24
<b>139</b>	154	184	36	44
<b>140</b>	46	5	8	34
<b>141</b>	80	190	13	18
<b>142</b>	100	144	17	23
<b>143</b>	121	46	3	13
<b>144</b>	177	134	26	42
<b>145</b>	155	107	10	37
<b>146</b>	62	152	19	31
<b>147</b>	148	38	6	9
<b>148</b>	178	127	23	32
<b>149</b>	41	16	6	2
<b>150</b>	144	92	23	32
<b>151</b>	122	13	27	38
<b>152</b>	197	7	13	35
<b>153</b>	9	122	17	17
<b>154</b>	27	97	30	21
<b>155</b>	149	176	16	26
<b>156</b>	106	41	13	37
<b>157</b>	19	51	19	48
<b>158</b>	8	89	13	30
<b>159</b>	151	132	6	31
<b>160</b>	56	148	20	10
<b>161</b>	122	38	5	4

<b>162</b>	90	174	18	48
<b>163</b>	114	122	10	34
<b>164</b>	108	148	21	37
<b>165</b>	41	73	21	5
<b>166</b>	149	103	17	37
<b>167</b>	42	148	14	29
<b>168</b>	103	97	9	43
<b>169</b>	107	133	22	13
<b>170</b>	95	50	25	31
<b>171</b>	114	183	35	10
<b>172</b>	113	105	23	1
<b>173</b>	142	27	11	13
<b>174</b>	25	165	24	1
<b>175</b>	168	58	29	29
<b>176</b>	200	162	24	2
<b>177</b>	135	110	7	19
<b>178</b>	66	98	2	22
<b>179</b>	55	87	30	27
<b>180</b>	116	52	10	37
<b>181</b>	125	53	24	40
<b>182</b>	115	50	14	38
<b>183</b>	97	9	13	33
<b>184</b>	184	110	23	48
<b>185</b>	110	110	14	35
<b>186</b>	63	116	21	7
<b>187</b>	152	122	25	18
<b>188</b>	94	119	25	23
<b>189</b>	46	131	21	12
<b>190</b>	81	183	11	29
<b>191</b>	3	101	33	31
<b>192</b>	191	124	7	46
<b>193</b>	135	45	7	30
<b>194</b>	125	49	17	35
<b>195</b>	173	42	3	10
<b>196</b>	172	179	1	9
<b>197</b>	23	123	15	9
<b>198</b>	12	15	16	2
<b>199</b>	118	45	28	30
<b>200</b>	10	20	14	24
<b>201</b>	161	3	22	37

<b>202</b>	19	38	16	33
<b>203</b>	62	138	15	14
<b>204</b>	145	71	10	5
<b>205</b>	32	23	10	37
<b>206</b>	42	72	18	29
<b>207</b>	193	127	7	23
<b>208</b>	183	170	17	26
<b>209</b>	36	104	11	3
<b>210</b>	124	165	24	26
<b>211</b>	198	97	14	47
<b>212</b>	168	146	19	46
<b>213</b>	121	50	24	16
<b>214</b>	92	178	26	37
<b>215</b>	68	156	28	40
<b>216</b>	115	75	26	44
<b>217</b>	59	139	26	29
<b>218</b>	112	78	34	13
<b>219</b>	7	119	30	8
<b>220</b>	123	59	18	37
<b>221</b>	5	162	26	11
<b>222</b>	147	193	29	44
<b>223</b>	186	149	25	10
<b>224</b>	122	58	15	20
<b>225</b>	113	158	19	5
<b>226</b>	76	93	7	48
<b>227</b>	40	115	41	13
<b>228</b>	53	79	14	19
<b>229</b>	172	55	11	15
<b>230</b>	137	10	21	30
<b>231</b>	37	177	21	33
<b>232</b>	131	136	18	33
<b>233</b>	100	53	22	23
<b>234</b>	113	14	20	23
<b>235</b>	159	129	10	17
<b>236</b>	114	72	22	11
<b>237</b>	157	196	30	13
<b>238</b>	148	196	13	28
<b>239</b>	24	197	17	27
<b>240</b>	85	193	25	9
<b>241</b>	68	40	23	43

<b>242</b>	23	74	21	1
<b>243</b>	85	164	23	36
<b>244</b>	111	129	27	5
<b>245</b>	69	64	48	35
<b>246</b>	174	185	15	5
<b>247</b>	165	156	17	36
<b>248</b>	26	45	8	14
<b>249</b>	127	143	7	7
<b>250</b>	75	134	16	48
<b>251</b>	141	132	18	31
<b>252</b>	8	8	11	47
<b>253</b>	12	113	2	18
<b>254</b>	28	187	16	7
<b>255</b>	41	186	27	1
<b>256</b>	62	38	19	16
<b>257</b>	16	141	20	4
<b>258</b>	78	118	28	44
<b>259</b>	113	151	2	25
<b>260</b>	40	130	1	47
<b>261</b>	142	60	13	23
<b>262</b>	116	4	13	47
<b>263</b>	94	31	18	48
<b>264</b>	28	15	18	45
<b>265</b>	152	190	10	35
<b>266</b>	7	108	32	1
<b>267</b>	42	165	22	15
<b>268</b>	182	89	21	12
<b>269</b>	125	34	14	35
<b>270</b>	118	139	13	10
<b>271</b>	148	157	21	1
<b>272</b>	59	24	19	7
<b>273</b>	75	173	30	19
<b>274</b>	143	103	15	29
<b>275</b>	188	70	13	17
<b>276</b>	78	136	31	4
<b>277</b>	146	166	24	17
<b>278</b>	103	25	2	34
<b>279</b>	152	68	9	43
<b>280</b>	137	59	23	36
<b>281</b>	98	77	7	15

<b>282</b>	136	111	19	32
<b>283</b>	198	16	18	17
<b>284</b>	154	59	17	44
<b>285</b>	176	134	14	33
<b>286</b>	179	151	20	37
<b>287</b>	125	78	11	9
<b>288</b>	114	112	17	34
<b>289</b>	172	66	23	21
<b>290</b>	20	113	22	13
<b>291</b>	67	127	21	40
<b>292</b>	70	37	10	29
<b>293</b>	109	197	15	39
<b>294</b>	135	171	27	46
<b>295</b>	181	156	21	34
<b>296</b>	35	160	16	25
<b>297</b>	193	12	3	32
<b>298</b>	98	89	29	22
<b>299</b>	23	176	20	34
<b>300</b>	54	37	21	23
<b>301</b>	151	179	12	33
<b>302</b>	116	118	12	32
<b>303</b>	127	60	23	22
<b>304</b>	103	77	21	38
<b>305</b>	17	166	22	16
<b>306</b>	46	22	10	32
<b>307</b>	42	144	30	25
<b>308</b>	44	10	16	38
<b>309</b>	108	187	22	29
<b>310</b>	64	156	4	44
<b>311</b>	83	81	21	31
<b>312</b>	199	123	20	35
<b>313</b>	120	39	31	46
<b>314</b>	109	45	14	35
<b>315</b>	114	153	23	22
<b>316</b>	87	30	10	45
<b>317</b>	131	92	22	37
<b>318</b>	137	158	3	4
<b>319</b>	65	77	19	15
<b>320</b>	191	138	2	39
<b>321</b>	161	158	24	19

<b>322</b>	164	37	19	26
<b>323</b>	47	118	23	12
<b>324</b>	111	98	16	44
<b>325</b>	144	59	18	23
<b>326</b>	28	127	24	40
<b>327</b>	34	145	21	48
<b>328</b>	165	181	22	22
<b>329</b>	50	135	8	15
<b>330</b>	105	26	24	4
<b>331</b>	70	28	11	43
<b>332</b>	10	60	16	38
<b>333</b>	172	161	6	16
<b>334</b>	51	112	13	45
<b>335</b>	73	81	19	45
<b>336</b>	69	133	8	36
<b>337</b>	71	114	25	41
<b>338</b>	84	180	12	9
<b>339</b>	112	6	39	7
<b>340</b>	31	79	19	8
<b>341</b>	99	45	21	44
<b>342</b>	127	39	13	20
<b>343</b>	62	15	29	18
<b>344</b>	71	131	11	16
<b>345</b>	197	29	16	33
<b>346</b>	138	31	10	39
<b>347</b>	138	69	9	41
<b>348</b>	25	48	11	43
<b>349</b>	59	83	24	38
<b>350</b>	126	199	18	16
<b>351</b>	32	47	27	20
<b>352</b>	188	194	9	8
<b>353</b>	109	85	3	23
<b>354</b>	13	139	34	38
<b>355</b>	74	55	3	1
<b>356</b>	6	193	13	29
<b>357</b>	130	149	17	35
<b>358</b>	45	55	11	12
<b>359</b>	126	100	36	43
<b>360</b>	178	88	23	24
<b>361</b>	129	176	18	21

<b>362</b>	80	167	10	15
<b>363</b>	179	5	9	39
<b>364</b>	12	131	28	25
<b>365</b>	15	197	10	16
<b>366</b>	66	196	23	47
<b>367</b>	110	184	10	45
<b>368</b>	71	145	7	47
<b>369</b>	45	142	15	18
<b>370</b>	56	19	22	41
<b>371</b>	26	197	17	17
<b>372</b>	179	95	19	7
<b>373</b>	196	126	6	41
<b>374</b>	32	190	13	13
<b>375</b>	101	29	7	18
<b>376</b>	83	112	28	24
<b>377</b>	125	180	1	1
<b>378</b>	145	19	20	30
<b>379</b>	150	26	16	7
<b>380</b>	98	193	27	8
<b>381</b>	28	77	12	24
<b>382</b>	143	116	18	31
<b>383</b>	175	60	18	47
<b>384</b>	56	90	15	27
<b>385</b>	88	15	22	28
<b>386</b>	89	153	12	34
<b>387</b>	38	183	10	30
<b>388</b>	109	48	3	18
<b>389</b>	89	39	19	44
<b>390</b>	195	132	6	10
<b>391</b>	108	141	13	23
<b>392</b>	22	61	17	12
<b>393</b>	110	8	26	34
<b>394</b>	64	123	17	35
<b>395</b>	139	97	4	10
<b>396</b>	168	119	14	28
<b>397</b>	151	66	11	47
<b>398</b>	29	121	46	29
<b>399</b>	32	166	17	21
<b>400</b>	88	143	17	29

## Instancia R241

	<b>Nro Clientes</b>	<b>Capacidad vehículos</b>		
	400	1000		
	<b><i>Cord X</i></b>	<b><i>Cord Y</i></b>	<b><i>Entrega</i></b>	<b><i>Recogida</i></b>
<b><i>Depósito</i></b>	100	100	0	0
<b>1</b>	150	26	16	27
<b>2</b>	84	188	17	37
<b>3</b>	83	112	28	39
<b>4</b>	191	124	7	33
<b>5</b>	112	174	20	5
<b>6</b>	146	166	24	12
<b>7</b>	45	95	15	40
<b>8</b>	81	183	11	24
<b>9</b>	10	60	16	20
<b>10</b>	142	27	11	14
<b>11</b>	184	110	23	14
<b>12</b>	28	127	24	35
<b>13</b>	67	127	21	24
<b>14</b>	16	141	20	25
<b>15</b>	148	157	21	38
<b>16</b>	154	184	36	42
<b>17</b>	8	89	13	38
<b>18</b>	74	163	21	21
<b>19</b>	25	48	11	44
<b>20</b>	58	19	10	32
<b>21</b>	95	50	25	7
<b>22</b>	130	149	17	22
<b>23</b>	78	118	28	38
<b>24</b>	86	69	14	13
<b>25</b>	5	122	15	42
<b>26</b>	155	17	32	1
<b>27</b>	146	3	24	15
<b>28</b>	76	93	7	30
<b>29</b>	191	138	2	15
<b>30</b>	28	160	27	24
<b>31</b>	114	153	23	6
<b>32</b>	70	28	11	9
<b>33</b>	90	4	22	16

<b>34</b>	53	175	28	32
<b>35</b>	84	81	11	18
<b>36</b>	118	139	13	22
<b>37</b>	118	45	28	17
<b>38</b>	103	25	2	47
<b>39</b>	107	133	22	11
<b>40</b>	129	176	18	14
<b>41</b>	186	149	25	7
<b>42</b>	73	81	19	36
<b>43</b>	9	122	17	28
<b>44</b>	168	146	19	26
<b>45</b>	172	161	6	37
<b>46</b>	15	197	10	2
<b>47</b>	26	41	14	2
<b>48</b>	78	72	9	37
<b>49</b>	112	6	39	23
<b>50</b>	135	171	27	11
<b>51</b>	39	151	26	39
<b>52</b>	46	40	15	5
<b>53</b>	138	72	8	10
<b>54</b>	53	79	14	5
<b>55</b>	75	134	16	24
<b>56</b>	66	3	9	12
<b>57</b>	153	19	3	13
<b>58</b>	17	32	16	39
<b>59</b>	59	139	26	10
<b>60</b>	100	53	22	16
<b>61</b>	46	131	21	7
<b>62</b>	111	189	3	39
<b>63</b>	62	15	29	22
<b>64</b>	100	144	17	13
<b>65</b>	70	130	8	30
<b>66</b>	97	192	23	41
<b>67</b>	85	193	25	26
<b>68</b>	197	29	16	34
<b>69</b>	113	100	5	27
<b>70</b>	71	114	25	40
<b>71</b>	120	39	31	42
<b>72</b>	125	53	24	19
<b>73</b>	172	186	35	4

<b>74</b>	138	31	10	25
<b>75</b>	37	146	10	39
<b>76</b>	12	131	28	9
<b>77</b>	1	54	14	9
<b>78</b>	101	127	31	36
<b>79</b>	42	165	22	20
<b>80</b>	149	176	16	32
<b>81</b>	126	100	36	29
<b>82</b>	148	196	13	4
<b>83</b>	78	136	31	7
<b>84</b>	145	71	10	41
<b>85</b>	179	95	19	39
<b>86</b>	124	191	24	48
<b>87</b>	38	101	10	10
<b>88</b>	104	88	25	46
<b>89</b>	57	118	21	21
<b>90</b>	182	9	30	47
<b>91</b>	7	119	30	6
<b>92</b>	85	164	23	1
<b>93</b>	6	193	13	11
<b>94</b>	154	7	32	46
<b>95</b>	180	66	13	48
<b>96</b>	83	180	15	28
<b>97</b>	130	55	28	37
<b>98</b>	149	103	17	14
<b>99</b>	54	37	21	10
<b>100</b>	178	88	23	5
<b>101</b>	118	117	20	5
<b>102</b>	82	48	8	1
<b>103</b>	80	190	13	31
<b>104</b>	74	55	3	30
<b>105</b>	108	141	13	12
<b>106</b>	77	119	17	2
<b>107</b>	114	112	17	35
<b>108</b>	156	93	25	42
<b>109</b>	143	103	15	24
<b>110</b>	104	176	19	33
<b>111</b>	63	141	14	41
<b>112</b>	159	187	9	39
<b>113</b>	125	78	11	11

<b>114</b>	136	12	21	29
<b>115</b>	199	123	20	47
<b>116</b>	154	29	19	19
<b>117</b>	177	134	26	21
<b>118</b>	121	50	24	44
<b>119</b>	114	122	10	15
<b>120</b>	161	158	24	45
<b>121</b>	8	8	11	42
<b>122</b>	144	92	23	13
<b>123</b>	41	43	30	35
<b>124</b>	113	151	2	21
<b>125</b>	39	186	5	7
<b>126</b>	60	153	11	11
<b>127</b>	131	92	22	48
<b>128</b>	53	180	4	37
<b>129</b>	28	77	12	2
<b>130</b>	186	8	31	37
<b>131</b>	40	130	1	3
<b>132</b>	71	131	11	23
<b>133</b>	158	20	19	1
<b>134</b>	121	48	17	11
<b>135</b>	45	55	11	48
<b>136</b>	50	135	8	35
<b>137</b>	152	122	25	5
<b>138</b>	46	5	8	43
<b>139</b>	32	47	27	21
<b>140</b>	111	98	16	19
<b>141</b>	138	69	9	29
<b>142</b>	171	69	19	43
<b>143</b>	125	34	14	21
<b>144</b>	94	31	18	41
<b>145</b>	29	121	46	17
<b>146</b>	20	72	21	10
<b>147</b>	179	151	20	7
<b>148</b>	18	59	20	19
<b>149</b>	162	162	20	13
<b>150</b>	152	190	10	2
<b>151</b>	66	196	23	41
<b>152</b>	45	142	15	28
<b>153</b>	36	104	11	24

<b>154</b>	63	44	14	35
<b>155</b>	83	81	21	2
<b>156</b>	172	66	23	45
<b>157</b>	151	180	14	32
<b>158</b>	109	48	3	44
<b>159</b>	64	156	4	40
<b>160</b>	85	85	14	44
<b>161</b>	157	196	30	32
<b>162</b>	135	110	7	39
<b>163</b>	59	24	19	30
<b>164</b>	175	60	18	41
<b>165</b>	127	39	13	6
<b>166</b>	103	97	9	35
<b>167</b>	71	145	7	48
<b>168</b>	40	115	41	13
<b>169</b>	114	72	22	25
<b>170</b>	62	38	19	2
<b>171</b>	110	110	14	16
<b>172</b>	51	162	20	10
<b>173</b>	193	12	3	33
<b>174</b>	19	51	19	19
<b>175</b>	178	61	19	13
<b>176</b>	161	3	22	37
<b>177</b>	63	116	21	45
<b>178</b>	151	179	12	2
<b>179</b>	88	143	17	16
<b>180</b>	32	190	13	33
<b>181</b>	76	102	15	8
<b>182</b>	139	97	4	6
<b>183</b>	101	29	7	18
<b>184</b>	176	134	14	9
<b>185</b>	60	71	22	34
<b>186</b>	137	10	21	30
<b>187</b>	17	166	22	19
<b>188</b>	75	170	17	29
<b>189</b>	90	174	18	38
<b>190</b>	122	58	15	14
<b>191</b>	113	14	20	42
<b>192</b>	98	89	29	24
<b>193</b>	59	83	24	40

<b>194</b>	44	10	16	34
<b>195</b>	97	9	13	11
<b>196</b>	116	52	10	15
<b>197</b>	155	22	24	41
<b>198</b>	193	127	7	13
<b>199</b>	154	59	17	13
<b>200</b>	122	13	27	38
<b>201</b>	70	37	10	42
<b>202</b>	172	55	11	29
<b>203</b>	98	193	27	9
<b>204</b>	46	22	10	19
<b>205</b>	188	70	13	3
<b>206</b>	157	142	13	13
<b>207</b>	63	192	12	27
<b>208</b>	88	15	22	47
<b>209</b>	189	83	43	42
<b>210</b>	110	8	26	2
<b>211</b>	171	36	6	3
<b>212</b>	65	42	12	8
<b>213</b>	113	158	19	33
<b>214</b>	51	112	13	21
<b>215</b>	151	132	6	4
<b>216</b>	50	31	20	47
<b>217</b>	181	70	25	15
<b>218</b>	12	113	2	25
<b>219</b>	20	113	22	20
<b>220</b>	125	49	17	2
<b>221</b>	27	97	30	33
<b>222</b>	116	4	13	35
<b>223</b>	136	111	19	31
<b>224</b>	104	142	24	16
<b>225</b>	112	42	33	33
<b>226</b>	145	19	20	26
<b>227</b>	23	168	13	3
<b>228</b>	168	58	29	6
<b>229</b>	3	101	33	20
<b>230</b>	137	59	23	4
<b>231</b>	184	140	15	1
<b>232</b>	48	94	15	31
<b>233</b>	198	16	18	47

<b>234</b>	26	197	17	3
<b>235</b>	84	180	12	2
<b>236</b>	31	79	19	33
<b>237</b>	136	90	18	12
<b>238</b>	123	90	21	30
<b>239</b>	62	152	19	30
<b>240</b>	10	20	14	15
<b>241</b>	66	98	2	3
<b>242</b>	19	38	16	44
<b>243</b>	69	150	17	32
<b>244</b>	181	156	21	46
<b>245</b>	39	112	26	7
<b>246</b>	104	5	26	36
<b>247</b>	41	16	6	34
<b>248</b>	172	179	1	1
<b>249</b>	122	38	5	31
<b>250</b>	89	176	13	11
<b>251</b>	74	67	2	5
<b>252</b>	197	7	13	28
<b>253</b>	24	197	17	37
<b>254</b>	131	136	18	25
<b>255</b>	26	45	8	35
<b>256</b>	47	118	23	34
<b>257</b>	20	59	20	14
<b>258</b>	151	66	11	5
<b>259</b>	86	110	18	7
<b>260</b>	55	100	22	34
<b>261</b>	186	65	39	32
<b>262</b>	63	151	22	47
<b>263</b>	159	196	6	48
<b>264</b>	179	5	9	6
<b>265</b>	107	96	23	5
<b>266</b>	139	158	9	5
<b>267</b>	5	162	26	15
<b>268</b>	173	147	19	23
<b>269</b>	121	46	3	1
<b>270</b>	144	59	18	24
<b>271</b>	196	126	6	37
<b>272</b>	134	157	20	43
<b>273</b>	135	45	7	13

<b>274</b>	88	152	13	5
<b>275</b>	41	186	27	16
<b>276</b>	183	170	17	11
<b>277</b>	103	77	21	24
<b>278</b>	97	129	11	16
<b>279</b>	147	193	29	27
<b>280</b>	158	81	23	13
<b>281</b>	48	106	23	20
<b>282</b>	78	169	10	42
<b>283</b>	174	194	8	38
<b>284</b>	152	68	9	8
<b>285</b>	1	149	35	31
<b>286</b>	18	35	24	3
<b>287</b>	123	59	18	36
<b>288</b>	137	41	15	11
<b>289</b>	115	75	26	34
<b>290</b>	124	165	24	48
<b>291</b>	112	78	34	14
<b>292</b>	24	131	31	28
<b>293</b>	116	118	12	10
<b>294</b>	80	167	10	12
<b>295</b>	62	138	15	28
<b>296</b>	195	132	6	19
<b>297</b>	149	78	6	47
<b>298</b>	38	183	10	34
<b>299</b>	174	185	15	35
<b>300</b>	35	160	16	37
<b>301</b>	32	23	10	16
<b>302</b>	111	129	27	3
<b>303</b>	32	166	17	23
<b>304</b>	13	139	34	18
<b>305</b>	158	132	12	15
<b>306</b>	188	194	9	40
<b>307</b>	92	178	26	1
<b>308</b>	55	87	30	19
<b>309</b>	86	150	22	22
<b>310</b>	114	183	35	48
<b>311</b>	42	72	18	26
<b>312</b>	56	148	20	29
<b>313</b>	115	50	14	39

<b>314</b>	200	162	24	37
<b>315</b>	69	64	48	5
<b>316</b>	108	148	21	17
<b>317</b>	35	9	29	4
<b>318</b>	99	45	21	34
<b>319</b>	125	180	1	21
<b>320</b>	142	60	13	44
<b>321</b>	181	39	11	45
<b>322</b>	165	181	22	28
<b>323</b>	87	30	10	7
<b>324</b>	176	58	21	21
<b>325</b>	168	119	14	15
<b>326</b>	198	97	14	33
<b>327</b>	89	153	12	35
<b>328</b>	67	70	18	25
<b>329</b>	28	187	16	36
<b>330</b>	42	144	30	5
<b>331</b>	138	61	19	16
<b>332</b>	11	118	33	11
<b>333</b>	143	179	24	15
<b>334</b>	182	89	21	20
<b>335</b>	141	146	15	13
<b>336</b>	12	15	16	32
<b>337</b>	184	101	17	42
<b>338</b>	127	143	7	4
<b>339</b>	137	158	3	28
<b>340</b>	25	165	24	45
<b>341</b>	22	61	17	18
<b>342</b>	41	73	21	5
<b>343</b>	23	74	21	6
<b>344</b>	159	129	10	38
<b>345</b>	68	115	20	25
<b>346</b>	98	77	7	27
<b>347</b>	126	199	18	48
<b>348</b>	28	15	18	20
<b>349</b>	68	156	28	4
<b>350</b>	94	119	25	40
<b>351</b>	31	63	11	30
<b>352</b>	23	123	15	9
<b>353</b>	105	26	24	45

<b>354</b>	60	184	7	47
<b>355</b>	69	133	8	12
<b>356</b>	141	132	18	4
<b>357</b>	56	19	22	47
<b>358</b>	110	184	10	32
<b>359</b>	178	127	23	44
<b>360</b>	75	173	30	22
<b>361</b>	113	105	23	2
<b>362</b>	165	156	17	37
<b>363</b>	109	197	15	25
<b>364</b>	55	51	17	7
<b>365</b>	37	177	21	15
<b>366</b>	30	183	9	34
<b>367</b>	49	99	32	35
<b>368</b>	143	116	18	8
<b>369</b>	30	67	7	32
<b>370</b>	89	39	19	31
<b>371</b>	23	176	20	33
<b>372</b>	127	60	23	23
<b>373</b>	6	120	25	18
<b>374</b>	65	77	19	20
<b>375</b>	108	187	22	17
<b>376</b>	159	105	20	22
<b>377</b>	176	132	10	29
<b>378</b>	7	108	32	5
<b>379</b>	34	145	21	38
<b>380</b>	42	174	13	9
<b>381</b>	63	149	17	42
<b>382</b>	57	157	26	41
<b>383</b>	56	90	15	8
<b>384</b>	148	38	6	42
<b>385</b>	119	113	35	28
<b>386</b>	164	37	19	5
<b>387</b>	173	42	3	42
<b>388</b>	106	41	13	25
<b>389</b>	155	107	10	44
<b>390</b>	49	71	36	47
<b>391</b>	40	95	25	13
<b>392</b>	5	130	15	1
<b>393</b>	182	110	15	47

<b>394</b>	68	191	15	25
<b>395</b>	68	40	23	48
<b>396</b>	64	123	17	21
<b>397</b>	109	45	14	43
<b>398</b>	109	85	3	5
<b>399</b>	42	148	14	25
<b>400</b>	67	138	11	29

### Instancia C141

	<b>Nro Clientes</b>	<b>Capacidad vehículos</b>		
	400	200		
	<b>Cord X</b>	<b>Cord Y</b>	<b>Entrega</b>	<b>Recogida</b>
<b>Depósito</b>	100	100	0	0
<b>1</b>	181	172	10	40
<b>2</b>	154	31	10	20
<b>3</b>	123	106	20	40
<b>4</b>	23	137	20	20
<b>5</b>	190	28	20	20
<b>6</b>	8	71	10	50
<b>7</b>	103	130	20	20
<b>8</b>	62	50	20	10
<b>9</b>	89	161	10	40
<b>10</b>	117	91	20	50
<b>11</b>	12	156	30	50
<b>12</b>	88	165	30	40
<b>13</b>	107	125	10	10
<b>14</b>	8	74	20	20
<b>15</b>	105	125	10	20
<b>16</b>	118	192	30	30
<b>17</b>	146	11	10	40
<b>18</b>	93	13	30	50
<b>19</b>	188	125	10	10
<b>20</b>	87	117	10	20
<b>21</b>	123	31	20	50
<b>22</b>	121	189	20	20
<b>23</b>	174	128	20	20
<b>24</b>	197	80	10	50

<b>25</b>	151	165	10	10
<b>26</b>	140	4	10	50
<b>27</b>	78	184	20	30
<b>28</b>	96	150	20	20
<b>29</b>	96	147	10	30
<b>30</b>	149	182	20	50
<b>31</b>	87	191	10	20
<b>32</b>	189	151	30	40
<b>33</b>	99	144	10	30
<b>34</b>	64	52	20	50
<b>35</b>	7	151	20	30
<b>36</b>	151	99	10	50
<b>37</b>	87	118	10	20
<b>38</b>	179	141	20	20
<b>39</b>	192	151	10	40
<b>40</b>	143	0	10	40
<b>41</b>	66	58	20	30
<b>42</b>	122	14	20	10
<b>43</b>	66	52	10	50
<b>44</b>	173	55	20	50
<b>45</b>	119	37	20	10
<b>46</b>	199	28	20	40
<b>47</b>	90	161	10	20
<b>48</b>	36	193	20	20
<b>49</b>	159	160	20	10
<b>50</b>	194	112	10	20
<b>51</b>	176	173	20	10
<b>52</b>	117	193	30	30
<b>53</b>	186	96	20	40
<b>54</b>	39	56	10	20
<b>55</b>	189	120	10	10
<b>56</b>	29	139	10	20
<b>57</b>	37	190	10	20
<b>58</b>	165	117	10	40
<b>59</b>	16	84	20	50
<b>60</b>	187	97	10	30
<b>61</b>	10	153	20	10
<b>62</b>	194	149	20	40
<b>63</b>	187	112	30	30
<b>64</b>	103	70	10	10

<b>65</b>	193	159	20	40
<b>66</b>	77	115	20	20
<b>67</b>	190	154	30	30
<b>68</b>	3	75	10	50
<b>69</b>	17	93	20	20
<b>70</b>	151	167	10	20
<b>71</b>	12	85	10	30
<b>72</b>	120	11	10	40
<b>73</b>	140	181	20	30
<b>74</b>	6	67	10	50
<b>75</b>	137	4	10	40
<b>76</b>	103	124	20	40
<b>77</b>	10	67	30	40
<b>78</b>	48	30	50	40
<b>79</b>	53	67	20	50
<b>80</b>	164	113	20	20
<b>81</b>	47	49	20	50
<b>82</b>	187	92	20	40
<b>83</b>	121	115	20	20
<b>84</b>	46	51	20	30
<b>85</b>	144	189	30	30
<b>86</b>	180	93	10	30
<b>87</b>	152	168	20	30
<b>88</b>	117	95	10	30
<b>89</b>	195	25	10	40
<b>90</b>	108	168	20	50
<b>91</b>	149	103	20	50
<b>92</b>	60	58	20	30
<b>93</b>	197	79	10	10
<b>94</b>	39	58	30	10
<b>95</b>	147	186	30	20
<b>96</b>	154	163	10	30
<b>97</b>	96	128	30	10
<b>98</b>	30	187	10	30
<b>99</b>	123	121	20	40
<b>100</b>	104	68	10	40
<b>101</b>	102	128	20	40
<b>102</b>	18	89	10	50
<b>103</b>	197	77	20	20
<b>104</b>	156	11	20	20

<b>105</b>	144	33	30	40
<b>106</b>	101	67	20	40
<b>107</b>	159	121	20	10
<b>108</b>	84	192	10	10
<b>109</b>	139	7	20	10
<b>110</b>	7	152	10	50
<b>111</b>	181	147	20	10
<b>112</b>	104	170	20	20
<b>113</b>	13	81	30	10
<b>114</b>	172	57	10	50
<b>115</b>	80	108	20	40
<b>116</b>	180	171	20	50
<b>117</b>	146	3	10	20
<b>118</b>	123	33	20	30
<b>119</b>	127	8	20	50
<b>120</b>	185	115	20	30
<b>121</b>	192	113	20	50
<b>122</b>	194	75	10	50
<b>123</b>	135	0	10	20
<b>124</b>	157	30	10	50
<b>125</b>	53	29	30	30
<b>126</b>	85	110	10	10
<b>127</b>	182	171	30	50
<b>128</b>	79	193	30	40
<b>129</b>	181	52	10	10
<b>130</b>	148	103	30	30
<b>131</b>	190	109	10	40
<b>132</b>	119	188	20	20
<b>133</b>	197	140	10	10
<b>134</b>	162	123	40	40
<b>135</b>	102	170	40	40
<b>136</b>	120	198	10	10
<b>137</b>	175	50	20	20
<b>138</b>	23	142	10	30
<b>139</b>	149	99	20	40
<b>140</b>	102	174	20	30
<b>141</b>	30	142	10	50
<b>142</b>	175	143	20	30
<b>143</b>	193	27	10	50
<b>144</b>	193	82	10	40

<b>145</b>	103	67	20	20
<b>146</b>	55	64	30	40
<b>147</b>	197	25	10	30
<b>148</b>	42	105	20	30
<b>149</b>	100	129	30	50
<b>150</b>	120	115	20	30
<b>151</b>	183	120	10	50
<b>152</b>	115	89	20	50
<b>153</b>	196	27	10	10
<b>154</b>	131	38	20	40
<b>155</b>	145	31	20	50
<b>156</b>	196	137	20	40
<b>157</b>	98	150	10	20
<b>158</b>	159	8	10	30
<b>159</b>	15	88	10	40
<b>160</b>	191	27	10	10
<b>161</b>	196	111	30	50
<b>162</b>	150	6	20	10
<b>163</b>	95	74	20	30
<b>164</b>	42	101	10	20
<b>165</b>	53	59	20	50
<b>166</b>	193	23	20	30
<b>167</b>	40	105	10	20
<b>168</b>	152	4	20	50
<b>169</b>	100	151	10	20
<b>170</b>	24	140	20	40
<b>171</b>	87	195	10	50
<b>172</b>	140	6	30	30
<b>173</b>	197	113	20	40
<b>174</b>	40	183	20	50
<b>175</b>	127	114	20	40
<b>176</b>	182	46	10	50
<b>177</b>	151	190	40	10
<b>178</b>	8	69	20	50
<b>179</b>	44	52	20	50
<b>180</b>	43	54	10	50
<b>181</b>	179	174	10	50
<b>182</b>	13	152	20	40
<b>183</b>	137	1	10	50
<b>184</b>	127	118	20	40

<b>185</b>	118	92	30	20
<b>186</b>	129	40	20	30
<b>187</b>	123	4	10	50
<b>188</b>	55	24	10	20
<b>189</b>	41	182	30	20
<b>190</b>	22	133	10	50
<b>191</b>	151	101	20	40
<b>192</b>	82	194	10	10
<b>193</b>	185	124	10	50
<b>194</b>	180	147	10	50
<b>195</b>	97	7	20	30
<b>196</b>	108	165	10	20
<b>197</b>	47	50	10	50
<b>198</b>	93	167	40	20
<b>199</b>	186	91	10	30
<b>200</b>	124	6	20	50
<b>201</b>	149	167	40	50
<b>202</b>	125	38	30	20
<b>203</b>	123	8	10	30
<b>204</b>	199	76	30	10
<b>205</b>	194	129	20	50
<b>206</b>	124	117	30	50
<b>207</b>	115	4	20	50
<b>208</b>	105	70	10	10
<b>209</b>	120	92	20	30
<b>210</b>	200	136	20	10
<b>211</b>	14	84	20	40
<b>212</b>	100	150	10	40
<b>213</b>	190	112	20	20
<b>214</b>	146	7	10	10
<b>215</b>	196	28	20	30
<b>216</b>	100	6	30	50
<b>217</b>	118	97	20	50
<b>218</b>	12	154	10	50
<b>219</b>	97	162	10	30
<b>220</b>	196	78	30	30
<b>221</b>	172	114	20	30
<b>222</b>	14	92	30	10
<b>223</b>	7	142	20	30
<b>224</b>	63	55	10	20

<b>225</b>	54	30	10	50
<b>226</b>	141	8	20	20
<b>227</b>	156	12	20	40
<b>228</b>	134	6	20	50
<b>229</b>	198	78	20	10
<b>230</b>	65	53	10	20
<b>231</b>	52	28	10	50
<b>232</b>	125	15	10	30
<b>233</b>	199	115	20	40
<b>234</b>	32	178	10	50
<b>235</b>	179	55	30	40
<b>236</b>	11	151	20	50
<b>237</b>	99	154	10	30
<b>238</b>	95	16	20	20
<b>239</b>	83	197	20	50
<b>240</b>	27	135	10	20
<b>241</b>	102	121	10	20
<b>242</b>	179	165	20	20
<b>243</b>	87	196	30	50
<b>244</b>	38	190	10	10
<b>245</b>	183	167	10	30
<b>246</b>	46	49	20	20
<b>247</b>	149	5	10	10
<b>248</b>	84	194	10	40
<b>249</b>	93	150	10	40
<b>250</b>	35	186	10	40
<b>251</b>	177	146	10	30
<b>252</b>	26	147	20	20
<b>253</b>	21	85	20	40
<b>254</b>	195	97	20	50
<b>255</b>	99	146	10	50
<b>256</b>	195	129	40	10
<b>257</b>	105	127	10	50
<b>258</b>	190	24	20	50
<b>259</b>	178	165	30	30
<b>260</b>	68	50	10	20
<b>261</b>	190	110	20	30
<b>262</b>	60	51	20	10
<b>263</b>	180	50	20	50
<b>264</b>	181	151	10	20

<b>265</b>	143	189	10	30
<b>266</b>	56	59	30	20
<b>267</b>	48	108	20	30
<b>268</b>	50	67	10	50
<b>269</b>	101	70	20	20
<b>270</b>	185	123	30	20
<b>271</b>	38	188	20	40
<b>272</b>	151	28	20	30
<b>273</b>	121	8	10	10
<b>274</b>	25	143	20	10
<b>275</b>	26	138	10	20
<b>276</b>	176	144	20	30
<b>277</b>	78	119	30	20
<b>278</b>	51	31	20	50
<b>279</b>	109	175	20	40
<b>280</b>	150	98	10	30
<b>281</b>	180	52	10	10
<b>282</b>	141	4	20	20
<b>283</b>	152	3	20	30
<b>284</b>	166	117	20	40
<b>285</b>	93	7	20	40
<b>286</b>	113	94	30	30
<b>287</b>	83	188	30	20
<b>288</b>	8	73	20	30
<b>289</b>	154	2	20	10
<b>290</b>	17	150	20	30
<b>291</b>	116	189	30	50
<b>292</b>	140	11	10	40
<b>293</b>	189	97	20	40
<b>294</b>	184	116	20	30
<b>295</b>	198	76	10	10
<b>296</b>	186	94	20	40
<b>297</b>	97	65	20	40
<b>298</b>	154	172	30	20
<b>299</b>	90	160	20	30
<b>300</b>	109	126	10	10
<b>301</b>	148	98	20	10
<b>302</b>	144	170	10	20
<b>303</b>	193	83	20	10
<b>304</b>	15	87	20	30

<b>305</b>	89	158	30	30
<b>306</b>	153	29	20	40
<b>307</b>	183	152	30	20
<b>308</b>	53	61	20	10
<b>309</b>	123	36	10	10
<b>310</b>	59	57	10	50
<b>311</b>	102	122	10	10
<b>312</b>	45	48	10	40
<b>313</b>	185	96	30	20
<b>314</b>	120	192	10	20
<b>315</b>	165	113	10	30
<b>316</b>	177	145	10	20
<b>317</b>	126	192	10	10
<b>318</b>	103	66	20	50
<b>319</b>	69	47	10	10
<b>320</b>	150	3	10	30
<b>321</b>	110	164	20	40
<b>322</b>	178	123	30	20
<b>323</b>	53	31	10	30
<b>324</b>	193	131	20	20
<b>325</b>	83	118	20	40
<b>326</b>	145	9	10	50
<b>327</b>	12	69	10	10
<b>328</b>	145	189	30	50
<b>329</b>	97	67	10	50
<b>330</b>	193	28	20	40
<b>331</b>	79	115	30	40
<b>332</b>	63	58	20	30
<b>333</b>	187	142	10	40
<b>334</b>	99	12	30	10
<b>335</b>	181	89	10	50
<b>336</b>	111	93	10	30
<b>337</b>	192	28	30	40
<b>338</b>	198	132	10	40
<b>339</b>	57	54	20	50
<b>340</b>	39	112	20	10
<b>341</b>	118	5	20	30
<b>342</b>	122	37	30	20
<b>343</b>	100	70	30	10
<b>344</b>	150	103	20	50

<b>345</b>	28	193	30	40
<b>346</b>	121	36	20	40
<b>347</b>	152	31	10	10
<b>348</b>	175	57	30	40
<b>349</b>	149	101	20	50
<b>350</b>	140	3	10	20
<b>351</b>	165	116	10	30
<b>352</b>	143	5	10	10
<b>353</b>	153	160	40	20
<b>354</b>	121	116	10	20
<b>355</b>	147	6	10	50
<b>356</b>	162	118	20	10
<b>357</b>	12	66	20	10
<b>358</b>	98	4	20	20
<b>359</b>	189	95	10	10
<b>360</b>	91	160	30	30
<b>361</b>	147	26	30	50
<b>362</b>	85	192	10	10
<b>363</b>	178	147	10	10
<b>364</b>	40	113	10	50
<b>365</b>	122	8	20	30
<b>366</b>	119	113	30	40
<b>367</b>	45	104	20	50
<b>368</b>	84	118	20	10
<b>369</b>	8	154	20	10
<b>370</b>	25	146	30	10
<b>371</b>	178	140	10	20
<b>372</b>	146	103	10	20
<b>373</b>	40	108	30	10
<b>374</b>	183	148	10	50
<b>375</b>	99	128	20	20
<b>376</b>	151	35	20	50
<b>377</b>	154	27	20	20
<b>378</b>	42	102	30	40
<b>379</b>	45	46	20	40
<b>380</b>	144	190	10	30
<b>381</b>	129	38	10	30
<b>382</b>	199	132	10	40
<b>383</b>	193	111	20	20
<b>384</b>	172	51	10	20

<b>385</b>	188	88	20	40
<b>386</b>	40	49	20	50
<b>387</b>	52	38	30	20
<b>388</b>	178	54	20	20
<b>389</b>	181	175	30	40
<b>390</b>	41	187	20	20
<b>391</b>	49	104	30	40
<b>392</b>	150	99	20	50
<b>393</b>	106	166	10	40
<b>394</b>	5	73	10	40
<b>395</b>	148	32	10	30
<b>396</b>	182	154	10	50
<b>397</b>	116	12	20	30
<b>398</b>	78	115	10	20
<b>399</b>	121	188	20	50
<b>400</b>	61	50	10	30

### Instancia C241

	<b>Nro Clientes</b>	<b>Capacidad vehículos</b>		
	400	700		
	<b>Cord X</b>	<b>Cord Y</b>	<b>Entrega</b>	<b>Recogida</b>
<b>Depósito</b>	100	100	0	0
<b>1</b>	37	48	10	30
<b>2</b>	42	100	30	10
<b>3</b>	131	31	10	10
<b>4</b>	108	13	20	20
<b>5</b>	17	166	10	20
<b>6</b>	147	134	20	30
<b>7</b>	48	103	10	40
<b>8</b>	127	166	20	40
<b>9</b>	50	181	20	40
<b>10</b>	19	156	10	20
<b>11</b>	146	157	40	20
<b>12</b>	93	166	10	20
<b>13</b>	170	83	30	10
<b>14</b>	79	13	20	20
<b>15</b>	73	112	10	20

<b>16</b>	87	87	10	10
<b>17</b>	118	50	10	30
<b>18</b>	79	70	20	20
<b>19</b>	67	102	40	20
<b>20</b>	48	166	30	20
<b>21</b>	45	88	30	30
<b>22</b>	61	101	10	30
<b>23</b>	54	175	20	30
<b>24</b>	126	39	10	20
<b>25</b>	107	73	30	10
<b>26</b>	42	75	20	10
<b>27</b>	74	102	10	20
<b>28</b>	72	118	20	40
<b>29</b>	65	135	20	40
<b>30</b>	27	89	10	10
<b>31</b>	84	133	10	20
<b>32</b>	178	69	40	10
<b>33</b>	127	46	20	20
<b>34</b>	28	96	20	40
<b>35</b>	100	47	20	40
<b>36</b>	69	167	20	40
<b>37</b>	98	59	10	10
<b>38</b>	79	84	10	30
<b>39</b>	61	174	10	20
<b>40</b>	138	135	10	40
<b>41</b>	170	156	10	30
<b>42</b>	119	169	20	30
<b>43</b>	66	192	20	10
<b>44</b>	120	76	30	10
<b>45</b>	29	171	20	10
<b>46</b>	89	11	10	30
<b>47</b>	170	64	20	40
<b>48</b>	135	146	10	30
<b>49</b>	85	134	20	20
<b>50</b>	19	179	30	20
<b>51</b>	95	137	10	10
<b>52</b>	163	125	30	10
<b>53</b>	30	54	30	20
<b>54</b>	16	77	20	10
<b>55</b>	28	168	20	20

<b>56</b>	78	188	20	30
<b>57</b>	27	83	20	40
<b>58</b>	136	57	30	30
<b>59</b>	59	98	10	10
<b>60</b>	170	49	30	40
<b>61</b>	114	138	20	20
<b>62</b>	78	154	10	10
<b>63</b>	69	190	30	30
<b>64</b>	60	171	20	30
<b>65</b>	112	53	10	20
<b>66</b>	99	123	10	10
<b>67</b>	28	50	10	20
<b>68</b>	97	163	20	10
<b>69</b>	150	135	20	30
<b>70</b>	136	134	10	10
<b>71</b>	120	153	10	20
<b>72</b>	79	89	10	40
<b>73</b>	162	44	20	40
<b>74</b>	163	55	20	40
<b>75</b>	103	91	20	20
<b>76</b>	74	91	20	40
<b>77</b>	126	31	20	40
<b>78</b>	16	146	10	40
<b>79</b>	84	20	10	20
<b>80</b>	21	91	20	30
<b>81</b>	37	162	20	40
<b>82</b>	89	168	20	10
<b>83</b>	72	10	20	30
<b>84</b>	55	160	10	30
<b>85</b>	63	189	30	20
<b>86</b>	164	124	20	30
<b>87</b>	20	96	20	20
<b>88</b>	82	88	20	30
<b>89</b>	76	157	20	30
<b>90</b>	74	196	30	40
<b>91</b>	85	14	10	40
<b>92</b>	77	86	20	10
<b>93</b>	142	141	20	30
<b>94</b>	119	66	10	10
<b>95</b>	7	176	10	30

<b>96</b>	92	55	20	30
<b>97</b>	168	43	10	20
<b>98</b>	89	113	10	30
<b>99</b>	98	85	30	20
<b>100</b>	46	99	10	10
<b>101</b>	62	120	30	30
<b>102</b>	93	1	20	30
<b>103</b>	144	58	30	20
<b>104</b>	20	54	20	10
<b>105</b>	32	191	10	10
<b>106</b>	143	72	20	20
<b>107</b>	68	190	20	30
<b>108</b>	140	51	10	40
<b>109</b>	119	130	20	30
<b>110</b>	37	195	10	10
<b>111</b>	29	163	30	20
<b>112</b>	120	70	10	10
<b>113</b>	12	186	10	40
<b>114</b>	118	59	10	40
<b>115</b>	90	74	10	10
<b>116</b>	165	151	20	10
<b>117</b>	125	72	20	30
<b>118</b>	177	122	30	40
<b>119</b>	64	78	10	40
<b>120</b>	67	60	10	20
<b>121</b>	167	55	30	10
<b>122</b>	173	56	20	20
<b>123</b>	179	78	20	20
<b>124</b>	83	74	40	30
<b>125</b>	140	70	30	30
<b>126</b>	81	191	20	40
<b>127</b>	101	37	30	40
<b>128</b>	34	182	20	10
<b>129</b>	99	61	20	20
<b>130</b>	168	126	20	10
<b>131</b>	139	131	20	30
<b>132</b>	134	46	20	40
<b>133</b>	66	78	30	10
<b>134</b>	71	189	10	40
<b>135</b>	70	185	20	10

<b>136</b>	16	78	30	40
<b>137</b>	57	73	30	30
<b>138</b>	92	39	10	40
<b>139</b>	5	58	20	30
<b>140</b>	75	61	20	40
<b>141</b>	186	50	20	10
<b>142</b>	14	199	30	30
<b>143</b>	80	139	30	20
<b>144</b>	110	72	10	20
<b>145</b>	40	87	20	10
<b>146</b>	155	38	10	20
<b>147</b>	155	108	20	40
<b>148</b>	152	149	20	10
<b>149</b>	55	195	20	20
<b>150</b>	86	100	10	40
<b>151</b>	36	107	20	20
<b>152</b>	103	148	20	40
<b>153</b>	29	88	10	30
<b>154</b>	32	93	20	20
<b>155</b>	79	132	20	40
<b>156</b>	25	182	10	10
<b>157</b>	25	187	30	20
<b>158</b>	23	102	30	20
<b>159</b>	128	80	10	30
<b>160</b>	165	62	20	40
<b>161</b>	34	84	10	10
<b>162</b>	82	100	10	10
<b>163</b>	162	37	40	30
<b>164</b>	138	132	10	40
<b>165</b>	65	182	20	20
<b>166</b>	53	168	20	30
<b>167</b>	114	55	10	20
<b>168</b>	24	164	30	30
<b>169</b>	74	5	20	10
<b>170</b>	109	73	10	40
<b>171</b>	102	146	10	20
<b>172</b>	144	38	30	30
<b>173</b>	93	11	20	40
<b>174</b>	73	106	10	40
<b>175</b>	51	190	30	20

<b>176</b>	106	113	30	30
<b>177</b>	81	80	10	40
<b>178</b>	138	22	30	20
<b>179</b>	69	173	40	40
<b>180</b>	75	18	30	20
<b>181</b>	166	55	10	30
<b>182</b>	89	68	10	10
<b>183</b>	126	59	20	40
<b>184</b>	88	137	10	40
<b>185</b>	99	152	40	10
<b>186</b>	178	54	20	10
<b>187</b>	132	51	20	10
<b>188</b>	100	147	10	40
<b>189</b>	36	96	10	20
<b>190</b>	6	59	20	40
<b>191</b>	102	62	30	30
<b>192</b>	55	188	30	30
<b>193</b>	135	54	30	10
<b>194</b>	6	66	20	40
<b>195</b>	88	116	30	20
<b>196</b>	29	75	20	30
<b>197</b>	143	140	10	30
<b>198</b>	15	181	20	30
<b>199</b>	130	42	30	10
<b>200</b>	17	99	20	10
<b>201</b>	182	68	10	20
<b>202</b>	27	187	30	10
<b>203</b>	40	34	10	40
<b>204</b>	57	183	10	20
<b>205</b>	64	140	20	10
<b>206</b>	2	178	10	40
<b>207</b>	34	89	20	30
<b>208</b>	149	149	20	10
<b>209</b>	147	158	10	40
<b>210</b>	110	150	20	10
<b>211</b>	83	149	20	40
<b>212</b>	42	105	20	30
<b>213</b>	83	108	10	10
<b>214</b>	20	78	20	10
<b>215</b>	38	85	20	10

<b>216</b>	101	74	30	20
<b>217</b>	82	128	20	20
<b>218</b>	73	110	20	40
<b>219</b>	81	136	20	10
<b>220</b>	91	14	20	40
<b>221</b>	96	62	10	30
<b>222</b>	130	47	10	10
<b>223</b>	14	163	30	30
<b>224</b>	21	56	40	20
<b>225</b>	88	41	10	10
<b>226</b>	20	92	20	20
<b>227</b>	73	20	10	10
<b>228</b>	95	18	30	30
<b>229</b>	17	87	10	30
<b>230</b>	123	32	30	10
<b>231</b>	124	170	10	10
<b>232</b>	84	141	30	30
<b>233</b>	116	44	20	20
<b>234</b>	133	162	10	20
<b>235</b>	74	31	10	30
<b>236</b>	107	23	20	40
<b>237</b>	25	102	20	40
<b>238</b>	97	71	10	40
<b>239</b>	83	147	20	10
<b>240</b>	38	94	10	10
<b>241</b>	149	146	10	30
<b>242</b>	93	117	40	30
<b>243</b>	124	154	10	20
<b>244</b>	115	40	20	30
<b>245</b>	84	136	20	20
<b>246</b>	88	111	20	40
<b>247</b>	143	135	10	20
<b>248</b>	143	30	30	30
<b>249</b>	148	139	20	30
<b>250</b>	68	69	30	30
<b>251</b>	97	157	20	40
<b>252</b>	146	113	20	40
<b>253</b>	118	9	10	20
<b>254</b>	125	16	10	20
<b>255</b>	81	92	20	10

<b>256</b>	148	57	10	30
<b>257</b>	89	27	20	20
<b>258</b>	165	157	20	20
<b>259</b>	70	89	20	10
<b>260</b>	25	101	10	40
<b>261</b>	6	95	10	40
<b>262</b>	89	80	20	30
<b>263</b>	76	15	20	40
<b>264</b>	3	56	20	30
<b>265</b>	82	48	30	40
<b>266</b>	81	89	20	30
<b>267</b>	20	79	30	40
<b>268</b>	30	75	10	40
<b>269</b>	160	137	10	40
<b>270</b>	150	130	20	30
<b>271</b>	156	137	20	40
<b>272</b>	169	54	10	20
<b>273</b>	13	90	20	40
<b>274</b>	49	64	20	10
<b>275</b>	39	99	20	40
<b>276</b>	91	43	10	30
<b>277</b>	60	197	20	10
<b>278</b>	15	69	20	40
<b>279</b>	101	164	10	20
<b>280</b>	153	161	20	40
<b>281</b>	11	83	10	30
<b>282</b>	127	55	20	20
<b>283</b>	33	103	10	10
<b>284</b>	43	101	30	10
<b>285</b>	175	70	20	10
<b>286</b>	122	144	20	20
<b>287</b>	68	16	10	40
<b>288</b>	93	32	20	20
<b>289</b>	164	61	10	30
<b>290</b>	79	31	10	30
<b>291</b>	69	178	20	30
<b>292</b>	86	155	30	30
<b>293</b>	87	67	30	30
<b>294</b>	99	48	10	10
<b>295</b>	72	8	20	40

<b>296</b>	108	82	30	10
<b>297</b>	69	121	20	10
<b>298</b>	67	13	30	30
<b>299</b>	33	73	20	10
<b>300</b>	68	176	10	30
<b>301</b>	106	127	30	10
<b>302</b>	106	60	10	40
<b>303</b>	35	90	10	40
<b>304</b>	84	13	10	20
<b>305</b>	133	168	10	20
<b>306</b>	21	76	20	10
<b>307</b>	67	89	30	30
<b>308</b>	67	97	10	30
<b>309</b>	69	40	10	40
<b>310</b>	13	120	20	20
<b>311</b>	68	173	30	40
<b>312</b>	190	122	10	40
<b>313</b>	104	59	10	10
<b>314</b>	117	53	20	20
<b>315</b>	84	25	10	30
<b>316</b>	109	169	10	20
<b>317</b>	120	50	10	20
<b>318</b>	52	79	20	30
<b>319</b>	69	99	10	30
<b>320</b>	91	39	10	30
<b>321</b>	108	26	10	40
<b>322</b>	116	171	20	40
<b>323</b>	65	164	20	20
<b>324</b>	106	83	20	20
<b>325</b>	55	97	20	30
<b>326</b>	39	79	10	40
<b>327</b>	13	173	30	10
<b>328</b>	95	29	10	30
<b>329</b>	160	136	20	20
<b>330</b>	105	55	20	40
<b>331</b>	146	62	20	10
<b>332</b>	73	196	20	20
<b>333</b>	95	150	30	40
<b>334</b>	26	169	20	20
<b>335</b>	161	139	20	40

<b>336</b>	77	126	20	10
<b>337</b>	66	81	10	40
<b>338</b>	132	36	20	10
<b>339</b>	101	152	20	40
<b>340</b>	94	75	20	20
<b>341</b>	158	122	30	20
<b>342</b>	81	20	20	20
<b>343</b>	180	126	10	20
<b>344</b>	47	76	30	20
<b>345</b>	60	37	30	20
<b>346</b>	35	185	10	20
<b>347</b>	36	177	30	40
<b>348</b>	48	191	10	30
<b>349</b>	169	36	10	30
<b>350</b>	177	45	30	10
<b>351</b>	72	119	20	20
<b>352</b>	26	107	20	10
<b>353</b>	15	57	10	30
<b>354</b>	54	73	30	40
<b>355</b>	68	152	30	20
<b>356</b>	93	26	20	10
<b>357</b>	40	101	10	40
<b>358</b>	111	24	10	30
<b>359</b>	46	128	30	10
<b>360</b>	177	55	10	10
<b>361</b>	132	139	10	40
<b>362</b>	101	62	40	20
<b>363</b>	97	116	30	10
<b>364</b>	81	83	20	20
<b>365</b>	157	59	30	40
<b>366</b>	91	49	10	40
<b>367</b>	81	32	10	40
<b>368</b>	47	80	10	40
<b>369</b>	119	62	10	30
<b>370</b>	43	98	30	10
<b>371</b>	101	55	30	10
<b>372</b>	31	63	30	30
<b>373</b>	145	144	10	30
<b>374</b>	56	192	20	30
<b>375</b>	101	158	20	40

<b>376</b>	30	86	20	10
<b>377</b>	33	188	30	20
<b>378</b>	156	64	10	40
<b>379</b>	154	44	30	10
<b>380</b>	171	63	20	20
<b>381</b>	139	169	30	40
<b>382</b>	121	52	40	30
<b>383</b>	139	102	10	30
<b>384</b>	84	10	40	20
<b>385</b>	7	63	10	10
<b>386</b>	184	55	10	10
<b>387</b>	58	168	20	40
<b>388</b>	156	135	20	20
<b>389</b>	165	52	20	20
<b>390</b>	128	141	10	40
<b>391</b>	79	106	20	30
<b>392</b>	102	56	30	10
<b>393</b>	87	132	20	10
<b>394</b>	23	172	10	20
<b>395</b>	64	172	10	20
<b>396</b>	82	104	10	20
<b>397</b>	99	156	30	30
<b>398</b>	146	142	10	30
<b>399</b>	14	81	30	30
<b>400</b>	79	19	10	10

### Instancia RC141

	<b>Nro Clientes</b>	<b>Capacidad vehículos</b>		
	400	200		
	<b>Cord X</b>	<b>Cord Y</b>	<b>Entrega</b>	<b>Recogida</b>
<b>Depósito</b>	100	100	0	0
<b>1</b>	134	6	20	1
<b>2</b>	63	58	20	2
<b>3</b>	64	52	20	19
<b>4</b>	139	7	20	21
<b>5</b>	110	8	26	48
<b>6</b>	79	193	30	22

<b>7</b>	80	190	13	29
<b>8</b>	158	81	23	33
<b>9</b>	60	153	11	15
<b>10</b>	141	8	20	24
<b>11</b>	13	81	30	25
<b>12</b>	68	115	20	13
<b>13</b>	151	132	6	12
<b>14</b>	88	143	17	30
<b>15</b>	16	84	20	24
<b>16</b>	155	17	32	26
<b>17</b>	69	150	17	17
<b>18</b>	55	87	30	9
<b>19</b>	22	133	10	39
<b>20</b>	10	20	14	17
<b>21</b>	187	142	10	13
<b>22</b>	114	122	10	29
<b>23</b>	78	119	30	27
<b>24</b>	199	115	20	21
<b>25</b>	185	124	10	12
<b>26</b>	46	22	10	2
<b>27</b>	7	152	10	10
<b>28</b>	144	92	23	16
<b>29</b>	5	162	26	9
<b>30</b>	197	113	20	40
<b>31</b>	189	95	10	27
<b>32</b>	84	118	20	48
<b>33</b>	28	77	12	38
<b>34</b>	105	26	24	4
<b>35</b>	113	100	5	39
<b>36</b>	139	97	4	12
<b>37</b>	143	179	24	42
<b>38</b>	111	189	3	16
<b>39</b>	119	37	20	23
<b>40</b>	87	191	10	29
<b>41</b>	80	167	10	21
<b>42</b>	23	137	20	2
<b>43</b>	181	70	25	44
<b>44</b>	178	123	30	19
<b>45</b>	45	142	15	30
<b>46</b>	151	66	11	8

<b>47</b>	68	50	10	22
<b>48</b>	159	196	6	47
<b>49</b>	87	30	10	31
<b>50</b>	187	92	20	34
<b>51</b>	10	153	20	7
<b>52</b>	54	30	10	18
<b>53</b>	54	37	21	25
<b>54</b>	180	147	10	49
<b>55</b>	154	59	17	13
<b>56</b>	93	13	30	11
<b>57</b>	38	190	10	13
<b>58</b>	104	88	25	19
<b>59</b>	78	169	10	18
<b>60</b>	127	143	7	13
<b>61</b>	24	131	31	33
<b>62</b>	82	194	10	17
<b>63</b>	165	117	10	48
<b>64</b>	23	176	20	19
<b>65</b>	55	24	10	30
<b>66</b>	148	157	21	32
<b>67</b>	186	91	10	1
<b>68</b>	26	138	10	31
<b>69</b>	78	184	20	7
<b>70</b>	165	181	22	30
<b>71</b>	11	151	20	3
<b>72</b>	103	66	20	37
<b>73</b>	188	125	10	20
<b>74</b>	197	140	10	41
<b>75</b>	100	70	30	43
<b>76</b>	83	118	20	44
<b>77</b>	66	58	20	48
<b>78</b>	83	188	30	16
<b>79</b>	12	154	10	36
<b>80</b>	158	20	19	19
<b>81</b>	161	158	24	22
<b>82</b>	41	186	27	36
<b>83</b>	162	162	20	2
<b>84</b>	89	176	13	23
<b>85</b>	86	69	14	25
<b>86</b>	20	72	21	12

<b>87</b>	41	182	30	26
<b>88</b>	178	147	10	36
<b>89</b>	140	11	10	16
<b>90</b>	60	51	20	24
<b>91</b>	100	6	30	11
<b>92</b>	18	89	10	12
<b>93</b>	85	110	10	13
<b>94</b>	37	146	10	45
<b>95</b>	181	89	10	17
<b>96</b>	26	147	20	8
<b>97</b>	52	38	30	28
<b>98</b>	131	136	18	11
<b>99</b>	46	131	21	25
<b>100</b>	35	160	16	42
<b>101</b>	187	97	10	41
<b>102</b>	172	179	1	29
<b>103</b>	156	93	25	41
<b>104</b>	85	164	23	38
<b>105</b>	17	32	16	2
<b>106</b>	175	60	18	9
<b>107</b>	82	48	8	3
<b>108</b>	62	50	20	11
<b>109</b>	137	1	10	11
<b>110</b>	140	4	10	7
<b>111</b>	143	0	10	35
<b>112</b>	198	132	10	1
<b>113</b>	25	146	30	38
<b>114</b>	137	10	21	20
<b>115</b>	15	88	10	33
<b>116</b>	38	183	10	6
<b>117</b>	121	48	17	28
<b>118</b>	62	15	29	8
<b>119</b>	19	38	16	1
<b>120</b>	144	59	18	31
<b>121</b>	125	78	11	43
<b>122</b>	12	156	30	50
<b>123</b>	53	29	30	50
<b>124</b>	22	61	17	34
<b>125</b>	159	121	20	49
<b>126</b>	59	83	24	38

<b>127</b>	59	57	10	2
<b>128</b>	168	58	29	17
<b>129</b>	12	85	10	49
<b>130</b>	85	193	25	48
<b>131</b>	155	107	10	39
<b>132</b>	195	132	6	5
<b>133</b>	55	51	17	49
<b>134</b>	189	120	10	48
<b>135</b>	59	139	26	6
<b>136</b>	121	50	24	49
<b>137</b>	10	67	30	49
<b>138</b>	52	28	10	30
<b>139</b>	17	150	20	39
<b>140</b>	101	67	20	15
<b>141</b>	69	47	10	47
<b>142</b>	164	113	20	31
<b>143</b>	79	115	30	11
<b>144</b>	80	108	20	19
<b>145</b>	184	116	20	44
<b>146</b>	27	135	10	21
<b>147</b>	29	139	10	22
<b>148</b>	84	194	10	41
<b>149</b>	186	94	20	50
<b>150</b>	114	72	22	42
<b>151</b>	90	4	22	18
<b>152</b>	41	187	20	22
<b>153</b>	92	178	26	10
<b>154</b>	46	40	15	37
<b>155</b>	174	128	20	29
<b>156</b>	30	142	10	27
<b>157</b>	28	193	30	18
<b>158</b>	185	96	30	6
<b>159</b>	74	55	3	42
<b>160</b>	23	123	15	15
<b>161</b>	103	70	10	28
<b>162</b>	42	174	13	32
<b>163</b>	8	8	11	32
<b>164</b>	83	81	21	32
<b>165</b>	57	54	20	37
<b>166</b>	26	45	8	43

<b>167</b>	83	180	15	15
<b>168</b>	70	37	10	25
<b>169</b>	97	65	20	32
<b>170</b>	53	31	10	35
<b>171</b>	179	141	20	2
<b>172</b>	100	53	22	29
<b>173</b>	55	64	30	9
<b>174</b>	8	71	10	32
<b>175</b>	174	185	15	27
<b>176</b>	176	144	20	8
<b>177</b>	123	36	10	17
<b>178</b>	142	60	13	7
<b>179</b>	53	61	20	1
<b>180</b>	62	38	19	4
<b>181</b>	123	33	20	14
<b>182</b>	136	111	19	43
<b>183</b>	28	160	27	6
<b>184</b>	85	85	14	20
<b>185</b>	12	66	20	21
<b>186</b>	107	96	23	33
<b>187</b>	168	119	14	30
<b>188</b>	171	36	6	13
<b>189</b>	188	88	20	28
<b>190</b>	135	45	7	2
<b>191</b>	26	41	14	1
<b>192</b>	195	129	40	34
<b>193</b>	108	141	13	15
<b>194</b>	75	134	16	34
<b>195</b>	141	4	20	3
<b>196</b>	60	58	20	39
<b>197</b>	196	137	20	17
<b>198</b>	7	142	20	38
<b>199</b>	190	112	20	35
<b>200</b>	63	192	12	10
<b>201</b>	97	129	11	16
<b>202</b>	71	131	11	12
<b>203</b>	159	105	20	28
<b>204</b>	109	48	3	1
<b>205</b>	189	83	43	33
<b>206</b>	142	27	11	49

<b>207</b>	68	40	23	19
<b>208</b>	56	90	15	19
<b>209</b>	98	193	27	5
<b>210</b>	121	36	20	27
<b>211</b>	101	70	20	11
<b>212</b>	134	157	20	22
<b>213</b>	103	77	21	45
<b>214</b>	87	117	10	6
<b>215</b>	183	170	17	42
<b>216</b>	193	12	3	27
<b>217</b>	127	60	23	24
<b>218</b>	178	140	10	43
<b>219</b>	95	50	25	18
<b>220</b>	25	165	24	41
<b>221</b>	124	191	24	15
<b>222</b>	138	69	9	25
<b>223</b>	177	145	10	12
<b>224</b>	50	31	20	19
<b>225</b>	37	177	21	27
<b>226</b>	48	30	50	7
<b>227</b>	197	29	16	11
<b>228</b>	40	130	1	38
<b>229</b>	23	168	13	2
<b>230</b>	24	140	20	25
<b>231</b>	20	59	20	26
<b>232</b>	81	183	11	37
<b>233</b>	190	109	10	5
<b>234</b>	28	187	16	8
<b>235</b>	145	19	20	37
<b>236</b>	164	37	19	5
<b>237</b>	186	96	20	32
<b>238</b>	105	70	10	15
<b>239</b>	140	6	30	46
<b>240</b>	146	166	24	17
<b>241</b>	56	148	20	20
<b>242</b>	66	52	10	13
<b>243</b>	53	59	20	4
<b>244</b>	179	5	9	21
<b>245</b>	122	58	15	23
<b>246</b>	37	190	10	48

<b>247</b>	155	22	24	50
<b>248</b>	175	143	20	31
<b>249</b>	199	132	10	7
<b>250</b>	131	92	22	27
<b>251</b>	83	197	20	45
<b>252</b>	180	66	13	35
<b>253</b>	87	195	10	43
<b>254</b>	129	176	18	17
<b>255</b>	185	123	30	34
<b>256</b>	77	115	20	26
<b>257</b>	7	119	30	48
<b>258</b>	195	97	20	36
<b>259</b>	162	118	20	24
<b>260</b>	196	111	30	45
<b>261</b>	32	47	27	14
<b>262</b>	100	144	17	20
<b>263</b>	13	152	20	1
<b>264</b>	192	113	20	26
<b>265</b>	25	143	20	48
<b>266</b>	51	31	20	25
<b>267</b>	14	84	20	36
<b>268</b>	12	69	10	40
<b>269</b>	165	113	10	45
<b>270</b>	138	72	8	7
<b>271</b>	135	0	10	41
<b>272</b>	8	89	13	28
<b>273</b>	110	110	14	41
<b>274</b>	177	146	10	29
<b>275</b>	87	196	30	42
<b>276</b>	89	39	19	29
<b>277</b>	65	77	19	31
<b>278</b>	63	149	17	4
<b>279</b>	143	5	10	1
<b>280</b>	32	178	10	9
<b>281</b>	193	111	20	1
<b>282</b>	162	123	40	5
<b>283</b>	67	127	21	47
<b>284</b>	73	81	19	44
<b>285</b>	50	67	10	6
<b>286</b>	38	188	20	16

<b>287</b>	183	120	10	47
<b>288</b>	129	38	10	27
<b>289</b>	35	186	10	26
<b>290</b>	180	93	10	38
<b>291</b>	146	3	24	24
<b>292</b>	8	154	20	26
<b>293</b>	200	162	24	23
<b>294</b>	8	69	20	4
<b>295</b>	131	38	20	16
<b>296</b>	78	115	10	36
<b>297</b>	123	31	20	31
<b>298</b>	39	151	26	22
<b>299</b>	53	175	28	17
<b>300</b>	1	54	14	31
<b>301</b>	40	95	25	46
<b>302</b>	5	73	10	47
<b>303</b>	7	151	20	7
<b>304</b>	166	117	20	43
<b>305</b>	39	186	5	13
<b>306</b>	165	116	10	19
<b>307</b>	99	12	30	45
<b>308</b>	48	106	23	26
<b>309</b>	154	7	32	7
<b>310</b>	18	35	24	30
<b>311</b>	135	171	27	15
<b>312</b>	11	118	33	27
<b>313</b>	78	72	9	26
<b>314</b>	19	51	19	2
<b>315</b>	69	133	8	45
<b>316</b>	56	59	30	44
<b>317</b>	99	45	21	27
<b>318</b>	103	67	20	3
<b>319</b>	53	79	14	5
<b>320</b>	194	112	10	7
<b>321</b>	42	165	22	28
<b>322</b>	68	156	28	13
<b>323</b>	172	114	20	48
<b>324</b>	98	4	20	47
<b>325</b>	104	176	19	7
<b>326</b>	200	136	20	29

<b>327</b>	21	85	20	28
<b>328</b>	67	138	11	14
<b>329</b>	111	129	27	34
<b>330</b>	97	7	20	15
<b>331</b>	114	112	17	5
<b>332</b>	125	49	17	31
<b>333</b>	75	173	30	32
<b>334</b>	42	72	18	3
<b>335</b>	53	180	4	21
<b>336</b>	174	194	8	14
<b>337</b>	39	112	26	48
<b>338</b>	6	193	13	13
<b>339</b>	6	67	10	48
<b>340</b>	14	92	30	40
<b>341</b>	3	75	10	40
<b>342</b>	23	142	10	50
<b>343</b>	57	118	21	8
<b>344</b>	65	53	10	39
<b>345</b>	140	3	10	46
<b>346</b>	187	112	30	50
<b>347</b>	112	6	39	46
<b>348</b>	63	55	10	38
<b>349</b>	104	68	10	6
<b>350</b>	30	187	10	6
<b>351</b>	194	129	20	34
<b>352</b>	53	67	20	14
<b>353</b>	186	65	39	36
<b>354</b>	12	131	28	41
<b>355</b>	122	37	30	47
<b>356</b>	36	193	20	26
<b>357</b>	113	105	23	47
<b>358</b>	84	180	12	19
<b>359</b>	136	90	18	40
<b>360</b>	61	50	10	46
<b>361</b>	97	67	10	29
<b>362</b>	190	110	20	29
<b>363</b>	87	118	10	12
<b>364</b>	137	4	10	15
<b>365</b>	179	151	20	34
<b>366</b>	189	97	20	35

<b>367</b>	15	87	20	33
<b>368</b>	157	142	13	10
<b>369</b>	138	61	19	25
<b>370</b>	65	42	12	19
<b>371</b>	60	184	7	42
<b>372</b>	85	192	10	2
<b>373</b>	93	7	20	27
<b>374</b>	95	16	20	9
<b>375</b>	125	38	30	46
<b>376</b>	149	78	6	36
<b>377</b>	185	115	20	15
<b>378</b>	158	132	12	39
<b>379</b>	29	121	46	42
<b>380</b>	122	13	27	3
<b>381</b>	57	157	26	26
<b>382</b>	8	73	20	45
<b>383</b>	115	75	26	31
<b>384</b>	41	43	30	8
<b>385</b>	104	5	26	50
<b>386</b>	172	55	11	36
<b>387</b>	184	140	15	31
<b>388</b>	12	113	2	31
<b>389</b>	40	183	20	27
<b>390</b>	193	131	20	13
<b>391</b>	129	40	20	2
<b>392</b>	84	192	10	6
<b>393</b>	95	74	20	28
<b>394</b>	8	74	20	44
<b>395</b>	181	147	20	36
<b>396</b>	17	93	20	36
<b>397</b>	66	196	23	7
<b>398</b>	123	90	21	48
<b>399</b>	110	184	10	16
<b>400</b>	121	46	3	10

**Instancia RC241**

<b>Nro Clientes</b>	<b>Capacidad vehículos</b>
-------------------------	--------------------------------

	400	1000		
	<i>Cord X</i>	<i>Cord Y</i>	<i>Entrega</i>	<i>Recogida</i>
<b>Depósito</b>	100	100	0	0
<b>1</b>	134	6	20	45
<b>2</b>	63	58	20	41
<b>3</b>	64	52	20	44
<b>4</b>	139	7	20	32
<b>5</b>	110	8	26	44
<b>6</b>	79	193	30	13
<b>7</b>	80	190	13	8
<b>8</b>	158	81	23	14
<b>9</b>	60	153	11	25
<b>10</b>	141	8	20	43
<b>11</b>	13	81	30	3
<b>12</b>	68	115	20	1
<b>13</b>	151	132	6	23
<b>14</b>	88	143	17	36
<b>15</b>	16	84	20	25
<b>16</b>	155	17	32	48
<b>17</b>	69	150	17	19
<b>18</b>	55	87	30	30
<b>19</b>	22	133	10	49
<b>20</b>	10	20	14	26
<b>21</b>	187	142	10	2
<b>22</b>	114	122	10	1
<b>23</b>	78	119	30	43
<b>24</b>	199	115	20	39
<b>25</b>	185	124	10	12
<b>26</b>	46	22	10	12
<b>27</b>	7	152	10	1
<b>28</b>	144	92	23	3
<b>29</b>	5	162	26	19
<b>30</b>	197	113	20	48
<b>31</b>	189	95	10	29
<b>32</b>	84	118	20	23
<b>33</b>	28	77	12	27
<b>34</b>	105	26	24	10
<b>35</b>	113	100	5	50
<b>36</b>	139	97	4	10
<b>37</b>	143	179	24	28

<b>38</b>	111	189	3	34
<b>39</b>	119	37	20	2
<b>40</b>	87	191	10	41
<b>41</b>	80	167	10	28
<b>42</b>	23	137	20	24
<b>43</b>	181	70	25	6
<b>44</b>	178	123	30	32
<b>45</b>	45	142	15	30
<b>46</b>	151	66	11	3
<b>47</b>	68	50	10	50
<b>48</b>	159	196	6	7
<b>49</b>	87	30	10	43
<b>50</b>	187	92	20	32
<b>51</b>	10	153	20	3
<b>52</b>	54	30	10	44
<b>53</b>	54	37	21	10
<b>54</b>	180	147	10	37
<b>55</b>	154	59	17	28
<b>56</b>	93	13	30	24
<b>57</b>	38	190	10	49
<b>58</b>	104	88	25	16
<b>59</b>	78	169	10	27
<b>60</b>	127	143	7	33
<b>61</b>	24	131	31	42
<b>62</b>	82	194	10	36
<b>63</b>	165	117	10	7
<b>64</b>	23	176	20	4
<b>65</b>	55	24	10	20
<b>66</b>	148	157	21	21
<b>67</b>	186	91	10	46
<b>68</b>	26	138	10	43
<b>69</b>	78	184	20	14
<b>70</b>	165	181	22	23
<b>71</b>	11	151	20	4
<b>72</b>	103	66	20	43
<b>73</b>	188	125	10	47
<b>74</b>	197	140	10	36
<b>75</b>	100	70	30	31
<b>76</b>	83	118	20	4
<b>77</b>	66	58	20	9

<b>78</b>	83	188	30	31
<b>79</b>	12	154	10	12
<b>80</b>	158	20	19	43
<b>81</b>	161	158	24	17
<b>82</b>	41	186	27	3
<b>83</b>	162	162	20	8
<b>84</b>	89	176	13	27
<b>85</b>	86	69	14	17
<b>86</b>	20	72	21	15
<b>87</b>	41	182	30	12
<b>88</b>	178	147	10	47
<b>89</b>	140	11	10	17
<b>90</b>	60	51	20	14
<b>91</b>	100	6	30	9
<b>92</b>	18	89	10	45
<b>93</b>	85	110	10	41
<b>94</b>	37	146	10	43
<b>95</b>	181	89	10	31
<b>96</b>	26	147	20	19
<b>97</b>	52	38	30	40
<b>98</b>	131	136	18	22
<b>99</b>	46	131	21	15
<b>100</b>	35	160	16	12
<b>101</b>	187	97	10	43
<b>102</b>	172	179	1	17
<b>103</b>	156	93	25	41
<b>104</b>	85	164	23	29
<b>105</b>	17	32	16	19
<b>106</b>	175	60	18	40
<b>107</b>	82	48	8	30
<b>108</b>	62	50	20	2
<b>109</b>	137	1	10	35
<b>110</b>	140	4	10	19
<b>111</b>	143	0	10	44
<b>112</b>	198	132	10	44
<b>113</b>	25	146	30	21
<b>114</b>	137	10	21	14
<b>115</b>	15	88	10	6
<b>116</b>	38	183	10	12
<b>117</b>	121	48	17	49

<b>118</b>	62	15	29	37
<b>119</b>	19	38	16	1
<b>120</b>	144	59	18	18
<b>121</b>	125	78	11	7
<b>122</b>	12	156	30	28
<b>123</b>	53	29	30	2
<b>124</b>	22	61	17	18
<b>125</b>	159	121	20	13
<b>126</b>	59	83	24	20
<b>127</b>	59	57	10	3
<b>128</b>	168	58	29	42
<b>129</b>	12	85	10	7
<b>130</b>	85	193	25	30
<b>131</b>	155	107	10	36
<b>132</b>	195	132	6	35
<b>133</b>	55	51	17	3
<b>134</b>	189	120	10	9
<b>135</b>	59	139	26	27
<b>136</b>	121	50	24	33
<b>137</b>	10	67	30	14
<b>138</b>	52	28	10	44
<b>139</b>	17	150	20	30
<b>140</b>	101	67	20	27
<b>141</b>	69	47	10	1
<b>142</b>	164	113	20	46
<b>143</b>	79	115	30	41
<b>144</b>	80	108	20	47
<b>145</b>	184	116	20	39
<b>146</b>	27	135	10	5
<b>147</b>	29	139	10	40
<b>148</b>	84	194	10	25
<b>149</b>	186	94	20	24
<b>150</b>	114	72	22	14
<b>151</b>	90	4	22	41
<b>152</b>	41	187	20	23
<b>153</b>	92	178	26	38
<b>154</b>	46	40	15	24
<b>155</b>	174	128	20	30
<b>156</b>	30	142	10	48
<b>157</b>	28	193	30	19

<b>158</b>	185	96	30	2
<b>159</b>	74	55	3	34
<b>160</b>	23	123	15	49
<b>161</b>	103	70	10	44
<b>162</b>	42	174	13	31
<b>163</b>	8	8	11	38
<b>164</b>	83	81	21	18
<b>165</b>	57	54	20	18
<b>166</b>	26	45	8	1
<b>167</b>	83	180	15	10
<b>168</b>	70	37	10	17
<b>169</b>	97	65	20	26
<b>170</b>	53	31	10	48
<b>171</b>	179	141	20	30
<b>172</b>	100	53	22	40
<b>173</b>	55	64	30	14
<b>174</b>	8	71	10	25
<b>175</b>	174	185	15	6
<b>176</b>	176	144	20	41
<b>177</b>	123	36	10	10
<b>178</b>	142	60	13	33
<b>179</b>	53	61	20	48
<b>180</b>	62	38	19	29
<b>181</b>	123	33	20	4
<b>182</b>	136	111	19	12
<b>183</b>	28	160	27	8
<b>184</b>	85	85	14	22
<b>185</b>	12	66	20	9
<b>186</b>	107	96	23	12
<b>187</b>	168	119	14	40
<b>188</b>	171	36	6	41
<b>189</b>	188	88	20	41
<b>190</b>	135	45	7	4
<b>191</b>	26	41	14	45
<b>192</b>	195	129	40	17
<b>193</b>	108	141	13	28
<b>194</b>	75	134	16	32
<b>195</b>	141	4	20	44
<b>196</b>	60	58	20	6
<b>197</b>	196	137	20	7

<b>198</b>	7	142	20	1
<b>199</b>	190	112	20	23
<b>200</b>	63	192	12	17
<b>201</b>	97	129	11	32
<b>202</b>	71	131	11	48
<b>203</b>	159	105	20	1
<b>204</b>	109	48	3	19
<b>205</b>	189	83	43	50
<b>206</b>	142	27	11	50
<b>207</b>	68	40	23	4
<b>208</b>	56	90	15	17
<b>209</b>	98	193	27	24
<b>210</b>	121	36	20	27
<b>211</b>	101	70	20	16
<b>212</b>	134	157	20	24
<b>213</b>	103	77	21	38
<b>214</b>	87	117	10	6
<b>215</b>	183	170	17	17
<b>216</b>	193	12	3	1
<b>217</b>	127	60	23	46
<b>218</b>	178	140	10	47
<b>219</b>	95	50	25	3
<b>220</b>	25	165	24	17
<b>221</b>	124	191	24	22
<b>222</b>	138	69	9	23
<b>223</b>	177	145	10	18
<b>224</b>	50	31	20	24
<b>225</b>	37	177	21	42
<b>226</b>	48	30	50	9
<b>227</b>	197	29	16	28
<b>228</b>	40	130	1	33
<b>229</b>	23	168	13	29
<b>230</b>	24	140	20	23
<b>231</b>	20	59	20	36
<b>232</b>	81	183	11	44
<b>233</b>	190	109	10	36
<b>234</b>	28	187	16	34
<b>235</b>	145	19	20	38
<b>236</b>	164	37	19	2
<b>237</b>	186	96	20	1

<b>238</b>	105	70	10	47
<b>239</b>	140	6	30	9
<b>240</b>	146	166	24	14
<b>241</b>	56	148	20	29
<b>242</b>	66	52	10	37
<b>243</b>	53	59	20	29
<b>244</b>	179	5	9	2
<b>245</b>	122	58	15	18
<b>246</b>	37	190	10	26
<b>247</b>	155	22	24	17
<b>248</b>	175	143	20	24
<b>249</b>	199	132	10	7
<b>250</b>	131	92	22	31
<b>251</b>	83	197	20	12
<b>252</b>	180	66	13	10
<b>253</b>	87	195	10	27
<b>254</b>	129	176	18	39
<b>255</b>	185	123	30	46
<b>256</b>	77	115	20	48
<b>257</b>	7	119	30	20
<b>258</b>	195	97	20	7
<b>259</b>	162	118	20	11
<b>260</b>	196	111	30	19
<b>261</b>	32	47	27	35
<b>262</b>	100	144	17	40
<b>263</b>	13	152	20	1
<b>264</b>	192	113	20	36
<b>265</b>	25	143	20	21
<b>266</b>	51	31	20	40
<b>267</b>	14	84	20	7
<b>268</b>	12	69	10	44
<b>269</b>	165	113	10	24
<b>270</b>	138	72	8	24
<b>271</b>	135	0	10	6
<b>272</b>	8	89	13	23
<b>273</b>	110	110	14	23
<b>274</b>	177	146	10	28
<b>275</b>	87	196	30	19
<b>276</b>	89	39	19	39
<b>277</b>	65	77	19	30

<b>278</b>	63	149	17	11
<b>279</b>	143	5	10	41
<b>280</b>	32	178	10	29
<b>281</b>	193	111	20	27
<b>282</b>	162	123	40	7
<b>283</b>	67	127	21	33
<b>284</b>	73	81	19	18
<b>285</b>	50	67	10	33
<b>286</b>	38	188	20	13
<b>287</b>	183	120	10	30
<b>288</b>	129	38	10	18
<b>289</b>	35	186	10	2
<b>290</b>	180	93	10	41
<b>291</b>	146	3	24	11
<b>292</b>	8	154	20	25
<b>293</b>	200	162	24	16
<b>294</b>	8	69	20	31
<b>295</b>	131	38	20	9
<b>296</b>	78	115	10	8
<b>297</b>	123	31	20	49
<b>298</b>	39	151	26	46
<b>299</b>	53	175	28	47
<b>300</b>	1	54	14	9
<b>301</b>	40	95	25	44
<b>302</b>	5	73	10	22
<b>303</b>	7	151	20	38
<b>304</b>	166	117	20	37
<b>305</b>	39	186	5	11
<b>306</b>	165	116	10	39
<b>307</b>	99	12	30	15
<b>308</b>	48	106	23	2
<b>309</b>	154	7	32	48
<b>310</b>	18	35	24	5
<b>311</b>	135	171	27	8
<b>312</b>	11	118	33	30
<b>313</b>	78	72	9	18
<b>314</b>	19	51	19	13
<b>315</b>	69	133	8	4
<b>316</b>	56	59	30	50
<b>317</b>	99	45	21	34

<b>318</b>	103	67	20	40
<b>319</b>	53	79	14	33
<b>320</b>	194	112	10	14
<b>321</b>	42	165	22	31
<b>322</b>	68	156	28	45
<b>323</b>	172	114	20	43
<b>324</b>	98	4	20	48
<b>325</b>	104	176	19	7
<b>326</b>	200	136	20	31
<b>327</b>	21	85	20	32
<b>328</b>	67	138	11	49
<b>329</b>	111	129	27	10
<b>330</b>	97	7	20	26
<b>331</b>	114	112	17	21
<b>332</b>	125	49	17	27
<b>333</b>	75	173	30	1
<b>334</b>	42	72	18	11
<b>335</b>	53	180	4	11
<b>336</b>	174	194	8	49
<b>337</b>	39	112	26	50
<b>338</b>	6	193	13	26
<b>339</b>	6	67	10	45
<b>340</b>	14	92	30	45
<b>341</b>	3	75	10	38
<b>342</b>	23	142	10	11
<b>343</b>	57	118	21	23
<b>344</b>	65	53	10	16
<b>345</b>	140	3	10	25
<b>346</b>	187	112	30	34
<b>347</b>	112	6	39	35
<b>348</b>	63	55	10	30
<b>349</b>	104	68	10	44
<b>350</b>	30	187	10	46
<b>351</b>	194	129	20	26
<b>352</b>	53	67	20	16
<b>353</b>	186	65	39	3
<b>354</b>	12	131	28	23
<b>355</b>	122	37	30	21
<b>356</b>	36	193	20	25
<b>357</b>	113	105	23	32

<b>358</b>	84	180	12	34
<b>359</b>	136	90	18	31
<b>360</b>	61	50	10	23
<b>361</b>	97	67	10	26
<b>362</b>	190	110	20	47
<b>363</b>	87	118	10	14
<b>364</b>	137	4	10	42
<b>365</b>	179	151	20	3
<b>366</b>	189	97	20	48
<b>367</b>	15	87	20	41
<b>368</b>	157	142	13	2
<b>369</b>	138	61	19	2
<b>370</b>	65	42	12	16
<b>371</b>	60	184	7	32
<b>372</b>	85	192	10	13
<b>373</b>	93	7	20	9
<b>374</b>	95	16	20	38
<b>375</b>	125	38	30	25
<b>376</b>	149	78	6	3
<b>377</b>	185	115	20	20
<b>378</b>	158	132	12	48
<b>379</b>	29	121	46	12
<b>380</b>	122	13	27	44
<b>381</b>	57	157	26	39
<b>382</b>	8	73	20	37
<b>383</b>	115	75	26	43
<b>384</b>	41	43	30	6
<b>385</b>	104	5	26	21
<b>386</b>	172	55	11	49
<b>387</b>	184	140	15	6
<b>388</b>	12	113	2	29
<b>389</b>	40	183	20	7
<b>390</b>	193	131	20	40
<b>391</b>	129	40	20	50
<b>392</b>	84	192	10	22
<b>393</b>	95	74	20	44
<b>394</b>	8	74	20	22
<b>395</b>	181	147	20	33
<b>396</b>	17	93	20	8
<b>397</b>	66	196	23	22

<b>398</b>	123	90	21	17
<b>399</b>	110	184	10	1
<b>400</b>	121	46	3	40

## Anexo C. Resultados computacionales

### Resultados r101

<i>Instancia</i>	r101		
<i>Constantes</i>	Babak		
<i>N° Partículas</i>	50		
<i>D</i>	Gran tour		
<i>N° Iteraciones</i>	<i>z</i>	<i>k</i>	
10	1262.6	13	
20	1260.7	13	
30	1210.9	13	
40	1224.4	13	
50	1248.7	13	
60	1254.1	13	
70	1225.2	13	
80	1220.8	13	
90	1221.6	13	
100	1247.8	13	
110	1230	13	
120	1202.8	13	
130	1196.7	13	
140	1195.5	13	
150	1215.8	13	
160	1216.6	13	
170	1171.8	13	
180	1135.1	13	
190	1191	13	
200	1203	13	

### Resultados r201\_experimento

<i>Instancia</i>	r201	<b>Mejor Z</b>
<i>Constantes</i>	Babak	
<i>N° Partículas</i>	50	666.2
<i>D</i>	Gran tour	

<b>N°</b>	<b>z</b>	<b>k</b>	<b>Z</b>	<b>Gap</b>	<b>k</b>	<b>iteraciones</b>
10	690.4457	3	694.8325	0.04297884	3	2
20	687.2385	3	701.0578	0.05232333	3	1
30	682.5957	3	690.2679	0.03612714	3	5
40	679.225	3	686.3723	0.03027965	3	4
50	677.0371	3	711.1101	0.06741234	3	1
60	682.6755	3	699.5131	0.05000465	3	4
70	680.8559	3	684.0722	0.02682708	3	2
80	680.5895	3	684.8225	0.02795332	3	6
90	677.8908	3	693.6126	0.0411477	3	2
100	682.9277	3	696.4811	0.04545347	3	4
110	679.4968	3	696.0996	0.04488082	3	2
120	679.8686	3	686.6014	0.03062354	3	3
130	677.8423	3	681.9147	0.02358856	3	6
140	680.9889	3	687.2023	0.03152552	3	1
150	679.8124	3	687.0156	0.03124527	3	2
160	675.0708	3	683.0817	0.02534029	3	6
170	671.1701	3	687.5492	0.03204623	3	4
180	674.5161	3	696.3094	0.04519574	3	6
190	674.5809	3	701.2779	0.05265371	3	1
200	-	-	683.0853	0.02534569	3	5
			698.2806	0.04815461	3	4
			681.0514	0.0222927	3	5
			687.0029	0.03122621	3	5
			687.8029	0.03242705	3	3
			692.5125	0.0394964	3	3
			688.7776	0.03389012	3	3
			685.2512	0.02859682	3	5
			681.3834	0.02279105	3	6
			693.3886	0.04081147	3	1
			688.0214	0.03275503	3	3

<b>Tratamiento (nivel)</b>	<b>Observaciones</b>					<b>Totales</b>	<b>Promedios</b>
	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>		
<b>1</b>	0.05232333	0.06741234	0.03152552	0.05265371	0.04081147	0.24472636	0.04894527
<b>2</b>	0.04297884	0.02682708	0.0411477	0.04488082	0.03124527	0.18707971	0.03741594
<b>3</b>	0.03062354	0.03242705	0.0394964	0.03389012	0.03275503	0.16919213	0.03383843
<b>4</b>	0.03027965	0.05000465	0.04545347	0.03204623	0.04815461	0.20593861	0.04118772
<b>5</b>	0.03612714	0.02534569	0.0222927	0.03122621	0.02859682	0.14358856	0.02871771

**Resultados c101\_experimento**

<b>Instancia</b>	c101						
<b>Constantes</b>	babak						
<b>N° Partículas</b>	50		<b>Mejor Z</b>				
<b>D</b>	Gran tour		1220.18				
<b>N° Iteraciones</b>	<b>z</b>	<b>k</b>	<b>Z</b>	<b>gap</b>	<b>k</b>	<b>iteraciones</b>	
10	1543	17	1530.6	0.25440509	17	2	
20	1510.5	18	1552.4	0.2722713	17	1	
30	1433.4	17	1473.5	0.20760871	18	5	
40	1585.2	17	1539.7	0.261863	18	4	
50	1539.9	17	1582.1	0.29661198	17	1	
60	1539.9	18	1539	0.26128932	18	4	
70	1546.7	17	1523.8	0.24883214	17	2	
80	1487	17	1496.4	0.22637644	17	6	
90	1529.8	17	1545.3	0.26645249	17	2	
100	1530.6	17	1502.7	0.23153961	17	4	
110	1557.4	17	1565.3	0.28284351	18	2	
120	1532.1	18	1539	0.26128932	17	3	
130	1508.5	18	1500.5	0.2297366	17	6	
140	1532.1	18	1524.8	0.24965169	17	1	
150	1466.8	17	1551.2	0.27128784	17	2	
160	1519.2	17	1523.8	0.24883214	17	6	
170	1557.2	17	1507.1	0.23514563	17	4	
180	1500.2	17	1528.1	0.25235621	18	6	
190	1508.1	17	1471.6	0.20605157	17	1	
200	1494.4	17	1468.7	0.20367487	18	5	
			1449.6	0.18802144	17	4	
			1486.3	0.21809897	17	5	
			1503.6	0.23227721	17	5	
			1520	0.24571784	17	3	
			1523.7	0.24875018	18	3	
			1555.5	0.27481191	17	3	
			1521.7	0.24711108	17	5	
			1527.7	0.25202839	18	6	
			1521.7	0.24711108	18	1	
			1527.7	0.25202839	18	3	

<i>Tratamiento (nivel)</i>	<i>Observaciones</i>					<i>Totales</i>	<i>Promedios</i>
	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>		
<b>1</b>	0.2722713	0.29661198	0.24965169	0.20605157	0.24711108	1.27169762	0.25433952
<b>2</b>	0.25440509	0.24883214	0.26645249	0.28284351	0.27128784	1.32382108	0.26476422
<b>3</b>	0.26128932	0.24571784	0.24875018	0.27481191	0.25202839	1.28259765	0.25651953
<b>4</b>	0.261863	0.26128932	0.23153961	0.23514563	0.18802144	1.177859	0.2355718
<b>5</b>	0.20760871	0.20367487	0.21809897	0.23227721	0.24711108	1.10877084	0.22175417
<b>6</b>	0.22637644	0.2297366	0.24883214	0.25235621	0.25202839	1.20932977	0.24186595

### Resultados C201

<i>Constantes</i>	Babak	
<i>N° Partículas</i>	50	
<i>D</i>	Gran tour	
<i>N° Iteraciones</i>	<i>z</i>	<i>k</i>
10	751.3142	5
20	733.5869	5
30	721.7862	5
40	701.8674	5
50	713.4305	5
60	697.7891	5
70	673.6288	5
80	686.1589	5
90	672.414	5
100	675.0649	5
110	676.0479	5
120	686.6449	5
130	675.2734	5
140	683.43366	5
150	723.893	5
160	673.2471	5
170	691.0409	5
180	674.3041	5
190	701.0162	5
200	664.2197	5

### Resultados rc101

<i>Instancia</i>	rc101
<i>Constantes</i>	babak

<b>N° Partículas</b>		50	
<b>D</b>		Gran tour	
<b>N° Iteraciones</b>		<b>z</b>	<b>k</b>
10		1365.7	12
20		1353.3	11
30		1294.2	11
40		1308	11
50		1353.1	12
60		1311.6	11
70		1309.5	11
80		1333.7	11
90		1303.8	12
100		1269	11
110		1289.2	11
120		1300.7	11
130		1322.3	11
140		1283.8	12
150		1252.2	11
160		1286.5	11
170		1218.3	11
180		1275.3	11
190		1231.4	11
200		1280.7	11

### Resultados rc201

<b>Constantes</b>		Babak	
<b>N° Partículas</b>		50	
<b>D</b>		Gran tour	
<b>N° Iteraciones</b>		<b>z</b>	<b>k</b>
10		697.9678	3
20		697.0317	3
30		688.5981	3
40		692.1897	3
50		690.5629	3
60		685.1768	3
70		698.2772	3
80		687.2880	3
90		685.6757	3

100	683.2831	3
110	695.7002	3
120	702.2355	3
130	681.9858	3
140	694.2876	3
150	675.045	3
160	689.0129	3
170	691.4177	3
180	691.7579	3
190	688.6635	3
200	698.1077	3

### Resultados R121\_Experimento

<i>Instancia</i>		R1_2_1					
<i>Constantes</i>		babak	<b>Mejor Z</b>				
<i>N° Partículas</i>		50	3375.19				
<i>D</i>		Gran tour					
<i>N° Iteraciones</i>		<i>z</i>	<i>k</i>	<b>Z</b>	<b>Gap</b>	<i>k</i>	<b>iteraciones</b>
10	4342.5	27	4512.62903	0.337	3	2	
20	4327.2	24	4293.24168	0.272	3	1	
30	4247.8	25	4229.11307	0.253	3	5	
40	4257.8	25	4276.36573	0.267	3	4	
50	4257.3	25	4313.49282	0.278	3	1	
60	4250.2	26	4266.24016	0.264	3	4	
70	4283.1	26	4303.36725	0.275	3	2	
80	4298.1	25	4434.99966	0.314	3	6	
90	4513.3	25	4296.61687	0.273	3	2	
100	4294.7	25	4364.12067	0.293	3	4	
110	4323.6	25	4266.24016	0.264	3	2	
120	4429.7	24	4289.86649	0.271	3	3	
130	4212.8	24	4195.36117	0.243	3	6	
140	4127.2	25	4313.49282	0.278	3	1	
150	4259.9	27	4242.61383	0.257	3	2	
160	4206.5	24	4252.7394	0.26	3	6	
170	4181	24	4421.4989	0.31	3	4	
180	4324.5	26	4320.2432	0.28	3	6	
190	4107.5	25	4411.37333	0.307	3	1	
200	4173.8	24	4218.9875	0.25	3	5	

4296.61687	0.273	3	4
4262.86497	0.263	3	5
4299.99206	0.274	3	5
4185.2356	0.24	3	3
4303.36725	0.275	3	3
4218.9875	0.25	3	3
4478.87713	0.327	3	5
4323.61839	0.281	3	6
4272.99054	0.266	3	1
4124.48218	0.222	3	3

Tratamiento (nivel)	Observaciones					Totales	Promedios
	1	2	3	4	5		
1	0.272	0.278	0.278	0.307	0.266	1.401	0.2802
2	0.337	0.275	0.273	0.264	0.257	1.406	0.2812
3	0.271	0.24	0.275	0.25	0.222	1.258	0.2516
4	0.267	0.264	0.293	0.31	0.273	1.407	0.2814
5	0.253	0.25	0.263	0.274	0.327	1.367	0.2734
6	0.314	0.243	0.26	0.28	0.281	1.378	0.2756

### Resultados R221

<b>Instancia</b>	R2_2_1		
<b>Constantes</b>	Babak		
<b>N° Partículas</b>	50		
<b>D</b>	Gran tour		
<b>N° Iteraciones</b>	<b>z</b>	<b>k</b>	
10	1879.1	5	
20	1786.5	5	
30	1786.5	5	
40	1828.3	5	
50	1828.3	5	
60	1766.4	5	
70	1800.4	5	
80	1806.1	5	
90	1832.2	6	
100	1824.8	5	
110	1774.1	5	

120	1768.4	5
130	1769.5	5
140	1779.2	5
150	1833.4	6
160	1847.3	5
170	1796.9	6
180	1763.2	5
190	1854.7	6
200	1789.4	6

### Resultados C121

<i>Instancia</i>	C1_2_1	
<i>Constantes</i>	Babak	
<i>N° Partículas</i>	50	
<i>D</i>	Gran tour	
<i>N° Iteraciones</i>	<i>z</i>	<i>k</i>
10	4484.6	32
20	4484.6	31
30	4600.5	31
40	4463.2	31
50	-	-
60	-	-
70	-	-
80	-	-
90	-	-
100	-	-
110	-	-
120	-	-
130	-	-
140	-	-
150	-	-
160	-	-
170	-	-
180	-	-
190	-	-
200	-	-

### Resultados C221 \_Experimento

N° Partículas			1726.59				
D			Gran tour				
N° Iteraciones			Z	Gap	k	iteraciones	
	z	k					
10	2142.7	10	2187.6	0.26700606	10		2
20	2142.7	10	2209.8	0.27986378	10		1
30	2188.7	10	2177.5	0.26115638	9		5
40	2216.4	10	2156.3	0.24887785	10		4
50	2108.8	10	2184.8	0.26538437	9		1
60	2148.8	9	2175.5	0.25999803	9		4
70	2139.4	9	2209.4	0.27963211	9		2
80	2124.6	9	2143.3	0.24134855	9		6
90	2136	10	2175	0.25970844	11		2
100	2142.7	10	2150.8	0.24569238	10		4
110	2142.7	10	2161.2	0.25171581	10		2
120	2135.8	10	2213.7	0.28212257	11		3
130	2133.9	9	2159.3	0.25061537	10		6
140	2129.5	9	2241.2	0.29804991	10		1
150	2138.4	9	2301.8	0.333148	9		2
160	2085.7	10	2130.2	0.23376134	10		6
170	1884.7	9	2152.7	0.24679281	11		4
180	2172.2	10	2125.3	0.23092338	10		6
190	1966.8	10	2202.8	0.27580954	10		1
200	2070	10	2076	0.20236999	10		5
			2249.2	0.30268332	9		4
			2130	0.23364551	10		5
			2131.5	0.23451427	10		5
			2216.1	0.28351259	10		3
			2192	0.26955444	10		3
			2129.1	0.23312425	10		3
			2159.1	0.25049954	10		5
			2206.1	0.27772083	10		6
			2331.8	0.35052329	9		1
			2137.6	0.23804725	10		3

Tratamiento (nivel)	Observaciones					Totales	Promedios
	1	2	3	4	5		
1	0.27986378	0.26538437	0.29804991	0.27580954	0.35052329	1.46963089	0.29392618
2	0.26700606	0.27963211	0.25970844	0.25171581	0.333148	1.39121042	0.27824208

3	0.28212257	0.28351259	0.26955444	0.23312425	0.23804725	1.30636109	0.26127222
4	0.24887785	0.25999803	0.24569238	0.24679281	0.30268332	1.30404439	0.26080888
5	0.26115638	0.20236999	0.23364551	0.23451427	0.25049954	1.1821857	0.23643714
6	0.24134855	0.25061537	0.23376134	0.23092338	0.27772083	1.23436948	0.2468739

### Resultados RC121

<b>Instancia</b>	RC1_2_1		
<b>Constantes</b>	Babak		
<b>N° Partículas</b>	50		
<b>D</b>	Gran tour		
<b>N° Iteraciones</b>	<b>z</b>	<b>k</b>	
10	4186.9	26	
20	4212.9	26	
30	4100.7	26	
40	4225.3	26	
50	4248.8	25	
60	4222.7	25	
70	4212.1	26	
80	4171.9	26	
90	4180.1	25	
100	4191.6	25	
110	4265.4	26	
120	4153.6	24	
130	4210.9	26	
140	4421.6	26	
150	4220.2	26	
160	4255.4	27	
170	4195.4	26	
180	4013.8	25	
190	4013.5	25	
200	4270.3	24	

### Resultados RC221

<b>Constantes</b>	Babak		
<b>N° Partículas</b>	50		
<b>D</b>	Gran tour		
<b>N°</b>	<b>z</b>	<b>k</b>	

<b>Iteraciones</b>			
10	2051.8	5	
20	2041.3	5	
30	-	-	
40	-	-	
50	-	-	
60	-	-	
70	-	-	
80	-	-	
90	-	-	
100	-	-	
110	-	-	
120	-	-	
130	-	-	
140	-	-	
150	-	-	
160	-	-	
170	-	-	
180	-	-	
190	-	-	
200	-	-	

### Resultados R141

<b>Instancia</b>	R1_4_1		
<b>Constantes</b>	Babak		
<b>N° Partículas</b>	50		
<b>D</b>	Gran tour		
<b>N° Iteraciones</b>	<b>z</b>	<b>k</b>	
10	12294	58	
20	11936	57	
30	12004	59	
40	12454	60	
50	12310	57	
60	12283	56	
70	12290	59	
80	12321	56	
90	12251	57	
100	12309	57	

110	13085	68
120	12194	57
130	12263	58
140	12210	56
150	12076	57
160	11891	59
170	12207	57
180	11937	57
190	12154	58
200	12250	57

### Resultados R241\_ experimento

<i>Instancia</i>		R2_4_1					
<i>Constantes</i>		Babak		<i>Mejor Z</i>			
<i>N° Partículas</i>		50		3560.73			
<i>D</i>		Gran tour					
<i>N° Iteraciones</i>	<i>z</i>	<i>k</i>	<i>Z</i>	<i>Gap</i>	<i>k</i>	<i>iteraciones</i>	
10	4635.8	12	4782.1	0.34301112	14	2	
20	4657.3	11	4593.5	0.29004446	12	1	
30	4592.4	12	4762.5	0.33750663	14	5	
40	4646.4	11	4524.5	0.27066641	12	4	
50	4630.5	12	4665.1	0.31015269	13	1	
60	4729.9	14	4729.8	0.32832312	11	4	
70	4610.2	12	4604.7	0.29318988	11	2	
80	4580.5	13	4573.4	0.28439955	13	6	
90	4564.4	11	4632	0.30085685	13	2	
100	4579.6	12	4608.7	0.29431324	11	4	
110	4561.4	13	4621.1	0.29779568	13	2	
120	4656.8	13	4584	0.28737646	11	3	
130	4399.5	12	4634.8	0.3016432	14	6	
140	4586.9	14	4700.5	0.32009448	11	1	
150	4402.3	11	4762.5	0.33750663	13	2	
160	4478.4	11	4487.2	0.26019103	11	6	
170	4577.1	12	4574.9	0.28482081	12	4	
180	4620.3	12	4634.8	0.3016432	12	6	
190	3997.7	11	4681.6	0.31478657	13	1	
200	4459.2	11	4590.1	0.2890896	11	5	
			4743.2	0.3320864	14	4	

4529.4	0.27204253	12	5
4462.5	0.25325425	12	5
4914.5	0.38019451	16	3
4685.5	0.31588186	11	3
4507.7	0.26594827	12	3
4707.7	0.32211653	15	5
4681.8	0.31484274	12	6
4689.7	0.31706139	11	1
4622.2	0.2981046	12	3

Tratamiento (nivel)	Observaciones					Totales	Promedio
	1	2	3	4	5		
1	0.29004446	0.31015269	0.32009448	0.31478657	0.31706139	1.55213959	0.310427
2	0.34301112	0.29318988	0.30085685	0.29779568	0.33750663	1.57236016	0.314472
3	0.28737646	0.38019451	0.31588186	0.26594827	0.2981046	1.54750571	0.309501
4	0.27066641	0.32832312	0.29431324	0.28482081	0.3320864	1.51020999	0.3020
5	0.33750663	0.2890896	0.27204253	0.25325425	0.32211653	1.47400954	0.294801
6	0.28439955	0.3016432	0.26019103	0.3016432	0.31484274	1.46271972	0.292543

### Resultados C141

<b>Instancia</b>	C2_4_1	
<b>Constantes</b>	Babak	
<b>N° Partículas</b>	50	
<b>D</b>	Gran tour	
<b>N° Iteraciones</b>	<b>z</b>	<b>k</b>
10	13938	68
20	13844	70
30	14436	68
40	14037	69
50	13800	68
60	14067	67
70	14010	70
80	13872	69
90	-	-
100	-	-
110	-	-
120	-	-

130	-	-
140	-	-
150	-	-
160	-	-
170	-	-
180	-	-
190	-	-
200	-	-

### Resultados C241\_experimento

<i>Instancia</i>		C2_4_1				
<i>Constantes</i>		Babak		<i>Mejor Z</i>		
<i>N° Partículas</i>		50		3549.2		
<i>D</i>		Gran tour				
<i>N° Iteraciones</i>		<i>z</i>	<i>k</i>	<i>Z</i>	<i>Gap</i>	<i>k</i>
10		4932.8	16	4883.8	0.3760284	16
20		4945.1	16	4662.1	0.31356362	17
30		4891.9	16	4943.9	0.39296179	16
40		4714	16	4942.1	0.39245464	16
50		4705.7	16	4861.9	0.369858	19
60		4751	16	4942.1	0.39245464	16
70		4811.5	18	4861.9	0.369858	19
80		4954.9	16	4825.8	0.35968669	16
90		4831.1	15	4762.8	0.34193621	17
100		4595.6	16	4975.1	0.40175251	16
110		4711.4	16	4864.3	0.3705342	17
120		4811	16	4799.9	0.35238927	16
130		4788.1	17	4987.1	0.40513355	16
140		4782.1	18	4905.7	0.38219881	16
150		4675.1	16	4933.1	0.38991885	18
160		4738.6	16	4821.2	0.35839062	17
170		4858.6	16	4748.1	0.33779443	16
180		4762	16	4922.3	0.38687592	17
190		4424.3	16	5126.2	0.44432548	16
200		4807.4	16	4828.1	0.36033472	16
				4840.9	0.36394117	16
				4740.7	0.33570946	17
				4778.3	0.34630339	16

4994.7	0.40727488	17
4723.3	0.33080694	16
4972	0.40087907	17
4914.9	0.38479094	16
4956.8	0.39659642	16
5080.9	0.43156204	16
4990.6	0.40611969	16

Tratamiento (nivel)	Observaciones					Totales	Promedios
	1	2	3	4	5		
1	0.31356362	0.369858	0.38219881	0.44432548	0.43156204	1.94150795	0.38830159
2	0.3760284	0.369858	0.34193621	0.3705342	0.38991885	1.84827567	0.36965513
3	0.35238927	0.40727488	0.33080694	0.40087907	0.40611969	1.89746985	0.37949397
4	0.39245464	0.39245464	0.40175251	0.33779443	0.36394117	1.88839739	0.37767948
5	0.39296179	0.36033472	0.33570946	0.34630339	0.38479094	1.8201003	0.36402006
6	0.35968669	0.40513355	0.35839062	0.38687592	0.39659642	1.9066832	0.38133664

### Resultados RC141

<b>Instancia</b>	RC1_4_1	
<b>Constantes</b>	Babak	
<b>N° Partículas</b>	50	
<b>D</b>	Gran tour	
<b>N° Iteraciones</b>	<b>z</b>	<b>k</b>
10	12406	54
20	12113	55
30	12122	57
40	-	-
50	-	-
60	-	-
70	-	-
80	-	-
90	-	-
100	-	-
110	-	-
120	-	-
130	-	-
140	-	-
150	-	-

160	-	-
170	-	-
180	-	-
190	-	-
200	-	-

### Resultados RC241

<b>Instancia</b>	RC2_4_1	
<b>Constantes</b>	Babak	
<b>N° Partículas</b>	50	
<b>D</b>	Gran tour	
<b>N° Iteraciones</b>	<b>z</b>	<b>k</b>
10	-	-
20	-	-
30	-	-
40	-	-
50	-	-
60	-	-
70	-	-
80	-	-
90	-	-
100	-	-
110	-	-
120	-	-
130	-	-
140	-	-
150	-	-
160	-	-
170	-	-
180	-	-
190	-	-
200	-	-

Anexo D. Artículo de investigación

## **ALGORITMO HÍBRIDO PARA LA RESOLUCIÓN DEL PROBLEMA DE RUTEO DE VEHÍCULOS CON ENTREGA Y RECOLECCIÓN SIMULTÁNEA.**

**LUDY JULIANA GONZÁLEZ VILLAMIZAR**  
ESTUDIANTE INGENIERÍA INDUSTRIAL  
UNIVERSIDAD INDUSTRIAL DE SANTANDER  
[ludy.gonzalez@correo.uis.edu.co](mailto:ludy.gonzalez@correo.uis.edu.co)

**CAMILO CRUZ JIMÉNEZ**  
ESTUDIANTE INGENIERÍA INDUSTRIAL  
UNIVERSIDAD INDUSTRIAL DE SANTANDER  
[camilo.cruz@correo.uis.edu.co](mailto:camilo.cruz@correo.uis.edu.co)

### **RESUMEN**

En este artículo se propone la formulación del problema de ruteo de vehículos con entrega y recolección simultánea (VRPSPD) y del método de optimización de enjambre de partículas para solucionarlo. El algoritmo desarrollado se basa en el GLNPSO, un PSO con múltiples estructuras sociales. Se propone una representación de la solución con componentes aleatorios y un método de decodificación para implementar el PSO al VRPSPD. La representación de la solución para el VRPSPD con  $n$  clientes y  $m$  vehículos, es una partícula con  $(n+2m)$  dimensiones. El método de decodificación inicia transformando la partícula en una lista de prioridad de clientes para entrar a las rutas y una matriz de prioridad de vehículos para servir cada cliente. Las rutas de los vehículos son construidas con base en la lista de prioridad de clientes y en la matriz de prioridad de vehículos. El algoritmo es evaluado usando 18 instancias disponibles en la literatura. Los resultados computacionales muestran que el método propuesto arroja buenas soluciones para problemas de 100, 200 y 400 clientes, sin embargo no mejora los resultados obtenidos por otros métodos para resolver el VRPSPD.

**PALABRAS CLAVE:** Ruteo de vehículos, Entrega y Recolección, Enjambre de Partículas, Optimización, Metaheurísticas, Algoritmo Híbrido.

### **ABSTRACT**

In this paper, a formulation for the vehicle routing problem with simultaneous pick-up and delivery (VRPSPD) and a particle swarm optimization algorithm (PSO) for solving it, is proposed. The PSO algorithm is developed based on GLNPSO, a PSO with multiple social structures. It is proposed a solution representation with random elements and a decoding method for implementing PSO for VRPSPD. The solution representation for VRPSPD with  $n$  customers and  $m$  vehicles is a particle with  $(n+2m)$  dimensions. The decoding method begins by transforming particle into a priority list of customers to enter the routes and a priority matrix of vehicles to serve each customer. Vehicle routes are constructed based on priority list of customers and vehicles priority matrix. The algorithm is tested using 18 instances available from the literature. Computational results show that the proposed method gives good solutions for 100, 200 and 400 customer problems, although it did not give any better result than other published methods for solving VRPSPD.

**KEY WORDS:** Vehicle Routing, Pick-up and delivery, Particle Swarm, Optimization, Metaheuristics, Hybrid Algorithm.

## 1. INTRODUCCIÓN

El Problema Ruteo de Vehículos (VRP por sus siglas en inglés) es un término genérico dado a una clase de problemas que consiste en la construcción de rutas desde un depósito hasta un número determinado de clientes ubicados en un espacio geográfico, con una flota de vehículos de capacidad limitada, con el fin de ofrecerles algún tipo de servicio, siendo la entrega de bienes el más común. Dada su vigencia en la gestión empresarial, han surgido variaciones del problema relacionadas con el tipo de actividad logística llevada a cabo en la asignación de rutas de transporte, y, de igual manera, los métodos de resolución del problema han evolucionado para encontrar soluciones adecuadas en tiempo razonable.

Una de las variaciones del VRP que encuentra gran aplicación en las operaciones logísticas de las empresas en la actualidad es el Problema de ruteo de vehículos con entrega y recolección simultáneas, propuesta por Min.[1] (VRPSPD por sus siglas en inglés). En este problema, 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 tiempo con el objetivo de minimizar costos, distancias, o tiempos de viaje.

Se trata de un problema de optimización combinatoria de tipo NP-hard para el cual se han aplicado técnicas heurísticas y metaheurísticas que consiguen soluciones factibles bastante adecuadas para cada situación, dado que los métodos exactos requieren un tiempo de cómputo significativamente grande debido a la complejidad algorítmica de tipo exponencial del problema[2].

En este trabajo de investigación se propone un algoritmo híbrido para la resolución del problema de ruteo de vehículos con entrega y recolección simultáneas (VRPSPD) utilizando un enfoque de optimización de enjambre de partículas (PSO) combinado con la técnica heurística de construcción de rutas *cheapest insertion heuristic* aplicada para decodificar la solución y una heurística 2-opt de búsqueda local que permite reoptimizar soluciones cuya pertinencia radica en la aplicación del algoritmo propuesto al contexto empresarial.

El algoritmo propuesto se ha desarrollado en MatLab® y para evaluar su desempeño se utilizaron 18 instancias disponibles en la literatura las cuales fueron implementadas por Tang y Galvão.[3].

## 2. PROBLEMA DE RUTEO DE VEHÍCULOS CON ENTREGA Y RECOLECCIÓN SIMULTÁNEA.

El VRP es un tema de investigación de gran interés en el campo de la optimización de sistemas productivos y logísticos dadas las aplicaciones prácticas encontradas. En él, se quiere determinar un conjunto de rutas de vehículos; la ruta vehicular parte de un depósito, sirve a un conjunto de clientes ubicados en una zona geográfica y retorna al mismo depósito, de tal forma que se satisfagan los requerimientos de la demanda, las restricciones de la operación y se minimice el costo total de transporte.[4]

La red de vías, usadas para el transporte de bienes, generalmente se representa a través de un grafo, cuyos arcos describen los tramos de las vías y cuyos vértices corresponden a intersecciones de las vías y a la localización de los clientes. Los arcos pueden ser dirigidos o no dirigidos, dependiendo de si pueden ser recorridos en un solo sentido, o bien, en ambos sentidos. Cada arco tiene un costo asociado que generalmente representa su longitud o su tiempo de viaje[4].

Se trata de un problema de optimización combinatoria, ya que las variables que definen las rutas, los vehículos y los clientes son de naturaleza discreta y se relacionan de diferentes maneras unas a otras[5]. El VRP fue introducido por primera vez por Dantzig y Ramser (1959), y a partir de ese momento, diferentes y más complejas variaciones al problema fueron tratadas a lo largo de los años[6]. Algunos tienen en cuenta el horario de entrega o recogida en cada lugar de interés (ventanas de tiempo), la naturaleza de la demanda de los clientes (determinística, estocástica) o el tipo de la flota de los vehículos (homogénea, heterogénea); otros consideran la disponibilidad de múltiples depósitos o la aleatoriedad de algunos parámetros del problema, entre otros.

Recientemente el VRP con entrega y recolección (VRPPD) ha recibido mayor atención por profesionales e investigadores debido a su alta importancia en aplicaciones prácticas de logística inversa. Dentro de las variantes del VRPPD se encuentra el VRP con retornos (VRPB, VRP with backhauls), el VRP con entrega y recolección mixta, el VRP con entrega y recolección simultánea (VRPSPD) y el problema Dial-a-Ride (DARP).

En 1989, H. Min[1] se interesó por un problema de la vida real de transporte de libros en una biblioteca, y desde ahí, planteó por primera vez la variación del VRP con entrega y recolección simultánea (VRPSPD). En su trabajo, planteó una aproximación a la solución que consistía en lo siguiente: (i) En primer lugar los clientes son clusterizados de forma que la capacidad del vehículo no sea excedida en cada grupo; (ii) un vehículo es asignado a cada cliente; y (iii) para cada grupo de clientes se resuelve el Problema del Agente Viajero (TSP)[6].

Las aplicaciones del VRPSPD son encontradas frecuentemente en los sistemas de distribución de las cadenas de abarrotes y alimentos, donde cada tienda de abarrotes usualmente presenta demandas tanto de entrega (comida o bebidas frescas) como de recolección (alimentos vencidos o botellas vacías) y es atendida en una sola visita del proveedor. Con esto, los minoristas pueden negociar la devolución del exceso de productos no vendidos a los productores con efectos beneficiosos para ambas partes. Esto también ocurre en industrias que aplican el uso de pallets o contenedores para el transporte de mercancías, en las cuales se reutilizan estos elementos. Además, dado que las leyes fuerzan a las empresas a tomar responsabilidad por la vida útil de sus productos, estas se ven obligadas a coleccionar productos usados con el fin de darles un apropiado procesamiento o disposición final. La planeación de rutas de vehículos para tales funciones, también es un VRPSPD[7].

Después de Min (1989), algunos investigadores también contribuyeron a la formulación matemática del VRPSPD y a sus técnicas de solución. Los métodos aproximados han sido ampliamente utilizados, debido a la complejidad del problema en instancias de gran escala, para el cual los métodos exactos son ineficientes. Si bien el problema puede ser abordado mediante métodos exactos para instancias pequeñas, se encuentra que los casos reales de aplicación del VRPSPD son siempre problemas de gran tamaño[8]. Toth y Vigo (2000) han mostrado sus enfoques por ser computacionalmente viables para problemas de más de 300 clientes.

Dethloff (2001)[7] propuso una heurística de inserción en la que los clientes son insertados en las rutas creadas de acuerdo a tres criterios: distancia de viaje, capacidad residual y distancia desde el depósito. Nagy y Salhi (1999)[9] proponen heurísticas basadas en inserción en las cuales se construyen rutas parciales para un conjunto de clientes, y luego se insertan los clientes faltantes a las rutas existentes. Nagy y Salhi (2005)[10] también propusieron una heurística de búsqueda local con cuatro fases para resolver el VRPSPD. Luego de encontrar una solución inicial en la primera fase, esta es continuamente mejorada en cada una de las siguientes fases manteniendo las condiciones de factibilidad. En ambos trabajos, se refirieron no solo al VRPSPD si no también al caso mixto de VRP en el que algunos clientes requieren entrega y otros requieren recolección. Encontraron que el VRPSPD es una generalización del problema mixto.

Tang y Galvao (2006)[3] desarrollaron un algoritmo de búsqueda tabú, el cual combina diferentes esfuerzos por obtener soluciones internas de cada ruta y entre las rutas alternativas. Este incluye reubicación de un cliente de una ruta a otra, intercambio de pares de clientes entre dos rutas, cruce entre dos rutas y procedimientos 2-opt.

Dell'Amico et al. (2006)[11] publicaron por primera vez un trabajo de investigación en métodos exactos para

resolver el VRPSPD. Presentaron un algoritmo de optimización basado en Generación de Columnas, programación dinámica y el método de branch-and-price (híbrido de branch-and-bound y generación de columnas). En su trabajo quedó evidenciada la complejidad computacional del VRPSPD en el que, tras analizar los resultados, se observó que 1 hora de tiempo computacional era insuficiente para resolver un problema pequeño de 40 clientes.

Wassan et al. (2007)[12] diseñaron un método de Búsqueda Tabú Reactivo que consiste en una fase de construcción de una solución inicial, y una segunda fase de mejoramiento. La solución inicial se construye mediante una modificación del método de barrido, que consiste en excluir clientes cercanos al depósito para ejecutar el barrido y cuya finalidad es la de dar flexibilidad para insertar aquellos clientes excluidos en mejores posiciones durante la fase de mejoramiento.

Chen et al (2006)[13] fueron unos de los primeros en aplicar la metaheurística PSO en su versión discreta, combinada con el procedimiento de búsqueda llamado Recocido simulado (SA, Simulated Annealing) para resolver el problema VRP básico. Otros métodos de búsqueda basados en comportamiento poblacional, como algoritmos genéticos y optimización de colonia de hormigas, han sido ampliamente utilizados.

Biancessi y Righini (2007)[14] evaluaron y compararon el desempeño de diferentes heurísticas constructivas, métodos de búsqueda local e implementaciones de búsqueda Tabú para VRPSPD. En particular, se enfocaron en la dificultad de aplicar el paradigma de Búsqueda Tabú a algoritmos basados en vecindades variables y complejas. Para este propósito, se combinaron vecindades basadas en intercambio de arcos y en intercambio de nodos, empleando listas tabú diferentes pero relacionadas.

Gajpal y Abad (2009)[15] presentaron una metaheurística de optimización de Colonia de Hormigas. En su trabajo, utilizaron una regla de construcción de rutas y dos rutinas de búsqueda local: 2-opt, inserción e intercambio de clientes multi-ruta, e intercambio de sub secuencias multi-ruta.

Ai y Kachitvichyanukul(2009)[4] propusieron una estrategia de PSO con múltiples estructuras sociales aplicada al VRPSPD. Las rutinas de decodificación están basadas en la construcción de listas de prioridad de clientes para entrar a las rutas y una matriz de prioridad de vehículos para servir a los mismos, de acuerdo a la posición de las partículas del enjambre generadas en cada iteración.

Zachariadis y Kiranoudis (2011)[16], propusieron una metodología metaheurística basada en un algoritmo de Búsqueda Local que usa dos conceptos algorítmicos llamados la estrategia Static Move Descriptor (SMD) para explorar eficientemente las soluciones de vecindades, y el promise concept para evitar búsquedas cíclicas e inducir diversificación. Estos algoritmos fueron aplicados previamente para modelos de VRP Abierto y VRPB con resultados satisfactorios. La experimentación del algoritmo luego de una comparación con 18 instancias de gran tamaño obtenidas de la literatura arrojó buenos resultados, obteniendo mejoramiento de soluciones obtenidas previamente.

### 1.1. FORMULACIÓN MATEMÁTICA DEL VRPSPD

El problema VRPSPD se puede formular de la siguiente manera[4]:

Sea  $G = (V, A)$  un grafo en donde  $V = \{v_0, v_1, \dots, v_n\}$  es un conjunto de nodos, y  $A = \{(v_i, v_j) | v_i, v_j \in V, i \neq j\}$  es un conjunto de arcos. Asociada al conjunto  $A$  se encuentra una matriz de distancias ( $d_{ij}$ ) Y una matriz de tiempo de viaje ( $t_{ij}$ ). El nodo  $V_0$  representa un depósito en el cual se encuentra un número de  $m$  vehículos homogéneos, mientras que los demás nodos corresponden a  $n$  clientes por atender. Cada cliente tiene cantidades no negativas de recogida  $p_i$ , de entrega  $q_i$  y un tiempo de servicio  $s_i$ . Cada vehículo tiene un costo fijo  $f$ , un costo variable por unidad de distancia  $g$ , capacidad  $Q$ , y duración límite de servicio  $D$ . El problema VRPSPD consiste en el diseño de un grupo de máximo  $m$  rutas de tal manera que:

- (1) Cada ruta empieza y termina en el depósito;
- (2) cada cliente es visitado una vez por un solo vehículo;
- (3) la carga total del vehículo en el trayecto entre dos clientes (considerando la demanda de entrega  $q_i$  y de recogida  $p_i$ ) no excede la capacidad de carga total del vehículo ( $Q$ );
- (4) la duración total de cada ruta (incluyendo el tiempo de viaje y de servicio) no excede el límite previamente establecido  $D$ ; y
- (5) el costo total de ruteo es mínimo.

El VRPSPD también se puede formular como problema de programación lineal entera mixta.

*Variables de decisión*

$x_{ijk}$  es una variable binaria que indica si el arco ( $i, j$ ) es recorrido o no por el vehículo  $k$

$x_{ijk} = 1$  si el vehículo  $k$  recorre el arco ( $i, j$ )

$x_{ijk} = 0$  si el vehículo  $k$  no recorre el arco ( $i, j$ )

$y_{ijk}$  carga del vehículo  $k$  que recorre el arco ( $i, j$ )

$\delta_{ijk}$  tiempo de inicio de servicio del cliente  $i$  efectuado el vehículo  $k$

#### Función Objetivo

$$\text{Minimizar } Z = f \sum_{k=1}^m \sum_{j=1}^n x_{0,jk} + g \sum_{i=0}^n \sum_{j=1}^{n+1} \sum_{k=1}^m d_{ij} * x_{ijk} \quad (1)$$

Sujeto a:

$$\sum_{i=0}^n \sum_{k=1}^m x_{ijk} = 1 \quad \text{para } 1 \leq j \leq n \quad (2)$$

$$\sum_{j=0}^n x_{jik} = \sum_{j=1}^{n+1} x_{ijk} \quad \text{para } 1 \leq i \leq n, 1 \leq k \leq m \quad (3)$$

$$\sum_{j=1}^n x_{0,jk} \leq 1 \quad \text{para } 1 \leq k \leq m \quad (4)$$

$$\delta_{ik} + s_i + t_{ij} + \delta_{jk} \leq (1 - x_{ijk})M \quad \text{para } 0 \leq i \leq n, 1 \leq j \leq n+1, 1 \leq k \leq m \quad (5)$$

$$\delta_{n+1,k} - \delta_{0k} \leq D, \quad \text{para } 1 \leq k \leq m \quad (6)$$

$$y_{ijk} \leq x_{ijk} * Q, \quad \text{para } 0 \leq i \leq n, 1 \leq j \leq n+1, 1 \leq k \leq m \quad (7)$$

$$\sum_{j=1}^n y_{ojk} = \sum_{j=1}^n q_j * \sum_{i=0}^n x_{ijk}, \quad \text{para } 1 \leq k \leq m \quad (8)$$

$$\sum_{i=0}^n y_{ijk} + (p_j - q_j) * \sum_{i=0}^n x_{ijk} = \sum_{i=1}^{n+1} y_{ijk}, \quad \text{para } 1 \leq j \leq n, 1 \leq k \leq m \quad (9)$$

$$x_{ijk} \in \{0, 1\}, \quad \text{para } 0 \leq i \leq n, 1 \leq j \leq n+1, 1 \leq k \leq m \quad (10)$$

$$y_{ijk} \geq 0, \quad \text{para } 0 \leq i \leq n, 1 \leq j \leq n+1, 1 \leq k \leq m \quad (11)$$

$$\delta_{ijk} \geq 0, \quad \text{para } 0 \leq i \leq n, 1 \leq j \leq n+1, 1 \leq k \leq m \quad (12)$$

En la formulación anterior, la función objetivo (1) muestra la minimización de costos de transporte teniendo en cuenta los costos fijos y variables asociados a los vehículos.

La restricción (2) indica que cada cliente es visitado únicamente por un vehículo; la restricción (3) indica que cada vehículo que llega a un nodo (cliente) debe salir de él y ese vehículo puede servir como máximo a una ruta. Esta restricción debe además garantizar que cada vehículo retorne al depósito luego de haber visitado el último cliente de la ruta, por lo cual debe considerarse un arco desde el nodo  $n$  hasta el nodo  $n+1$ , siendo este último el depósito al final del recorrido. Las restricciones (5) y (6) explican la relación entre las variables de tiempo y los parámetros del modelo. Así, la restricción (5) relaciona el tiempo de inicio de servicio entre un cliente y otro, y la restricción (6) indica la relación entre el primer y último cliente en cada ruta, es decir, establece el límite de duración del servicio.

La restricción (7) indica la máxima capacidad ( $Q$ ) del vehículo  $k$  mientras sirve al cliente  $j$  tras haber servido al cliente  $i$ . Por otro lado, la restricción (8) asegura que todas las entregas a clientes partan desde el depósito, y la restricción (9) se encarga de equilibrar la carga del vehículo  $k$  después del servicio a un cliente.

Finalmente, las restricciones (10)-(12) indican el dominio de las variables de decisión.

## 1. OPTIMIZACIÓN DE ENJAMBRE DE PARTÍCULAS PARA VRPSPD

En esta sección, se propone un algoritmo de optimización de enjambre de partículas para resolver el problema VRPSPD descrito previamente.

### 1.1. ALGORITMO PSO

El algoritmo de optimización de enjambre de partículas es un algoritmo poblacional de búsqueda basado en la simulación de comportamiento social de aves de una bandada. Formalmente un enjambre puede ser definido como un grupo de individuos que se comunican directa o indirectamente entre ellos, actuando en su respectivo ambiente. La interacción entre agentes resulta en estrategias colectivas de resolución de problemas. El término Inteligencia de Enjambres, hace referencia al comportamiento de resolución de problemas que resulta de la interacción de tales individuos e Inteligencia Computacional de enjambres a los modelos algorítmicos de tal comportamiento. Más formalmente, Inteligencia de Enjambres es la propiedad de un sistema por el cual el comportamiento colectivo de individuos poco sofisticados interactuando localmente con su ambiente generan patrones globales coherentes y funcionales para emerger[17].

El objetivo de los modelos de inteligencia computacional de partículas es modelar los simples

comportamientos individuales, y las interacciones ambientales y con individuos vecinos, con el fin de obtener comportamientos más complejos que puedan ser usados para resolver problemas complejos, principalmente de optimización. En el PSO, por su parte, los individuos, referidos como partículas, son “lanzados” en un espacio de búsqueda hiperdimensional. Los cambios de posición de las partículas dentro del espacio de búsqueda están basados en las tendencias sicosociales de individuos de emular el éxito de otros individuos., es decir, influenciados por la experiencia o conocimiento de sus vecinos. En otras palabras, se modelan dos simples comportamientos: (1) cada individuo se mueve en dirección a su vecino más cercano, y (2) regresa al estado que el individuo ha experimentado por sí solo como el mejor. Como resultado, el comportamiento colectivo que surge es que todos los individuos convergerán al estado que es mejor para todos ellos[18].

El algoritmo original está constituido por un enjambre de  $L$  partículas, con masa y volumen despreciables, que se mueve sobre un espacio de  $H$  dimensiones. La posición de la partícula  $\Theta_{i(t)} = [\theta_{i1}, \theta_{i2}, \dots, \theta_{iH}]$  representa una solución de distintos problemas de optimización. La habilidad de una partícula para encontrar soluciones se representa por su vector velocidad que dirige el movimiento de la partícula de una posición a otra.

Para realizar este movimiento, el enjambre cuenta con términos asociados a la ‘memoria’ que tiene cada partícula acerca de su experiencia y las de sus vecinos, de manera que la partícula actualiza su velocidad con respecto al comportamiento del enjambre. El modelo PSO combina dos tipos de aprendizaje para cada partícula del enjambre y un objetivo fijo que es común a cada una de ellas. En el PSO básico el primer tipo de aprendizaje de la partícula está asociado a la experiencia personal que ésta desarrolla en la medida en que se desplaza por el espacio de búsqueda conocido como *pbest*. Este comportamiento recibe el nombre de “*comportamiento cognitivo*”, y se define como la posición que arroja el mejor valor de la función objetivo entre todas las posiciones que han sido visitadas por la partícula. El segundo tipo de aprendizaje corresponde al término que se relaciona con el aprendizaje que obtiene la partícula de su interacción con el enjambre, conocido como *Gbest*. Éste es denominado “*comportamiento social*” que se relaciona con la experiencia del vecindario de cada partícula, y se define como la posición que arroja el mejor valor de la función objetivo entre las posiciones que han sido visitadas por todas las partículas.

La posición de cada partícula se puede denotar como:  $\Theta_{l(\tau)} = [\theta_{l1}, \theta_{l2}, \dots, \theta_{lH}]$  y puede ser codificada como la ‘solución’ de distintos problemas de optimización.

$\Theta_{l(\tau)}$  = Vector de posición de la partícula l en la iteración  $\tau$ .

$\theta_{lH}$  = Posición de la partícula l en la dimensión H.

La actualización del vector de posición está asociada a la velocidad, la cual se compone de los términos de aprendizaje de la partícula y el peso inercial de ésta. La velocidad para cada agente l del enjambre en la iteración  $\tau$  es un vector denotado como:

$$\Omega_l(\tau) = [\omega_{l1}, \omega_{l2}, \dots, \omega_{lH}]$$

$\omega_{lH}$  = Componente de la velocidad de la partícula l en la dimensión H

De esta manera, la posición de la partícula l, cambiará en cada iteración  $\tau$  de acuerdo a la siguiente formulación matemática:

$$\Theta_l(\tau + 1) = \Theta_l(\tau) + \Omega_l(\tau + 1)$$

$\Omega_l(\tau)$  = Vector de velocidad de la partícula en la iteración  $\tau$

Para utilizar el algoritmo aplicado a problemas de optimización, se requieren métodos de codificación que permitan reemplazar el ‘objetivo’ del enjambre por la función objetivo del problema. A partir del cambio en el desempeño de esta función, el método permite que la velocidad direcciona el movimiento de todo el enjambre. En este direccionamiento se refleja el conocimiento de todas las partículas del enjambre y el efecto de la inercia de cada una de ellas.

### 1.1. VARIACIÓN DEL ALGORITMO – GLNPSO

El PSO cuenta con numerosas variaciones surgidas, por ejemplo, a partir de modificaciones en el espacio de búsqueda (PSO discreto) o de la introducción de nuevos comportamientos sociales dentro de la estructura de aprendizaje de las partículas. Tal es el caso del GLNPSO, propuesto por Pongchairerks y Kachitvichyanukul[18]. En este algoritmo se combinan tres estructuras sociales estudiadas anteriormente de manera independiente: Global Best, Local Best y Near Neighbor Best.

La primera estructura, *Global Best*, corresponde a la estructura básica del PSO propuesto inicialmente. Por otro lado, la estructura denominada ‘Local Best’ y denotada como  $\Psi_l^L$ , incluye el concepto de vecindad de

partículas. En este caso, el enjambre no estará influenciado únicamente por su experiencia y la experiencia de la mejor partícula del enjambre, sino también por la experiencia de  $K$  vecinos más próximos.

Por otro lado, la estructura ‘Near neighbor Best’[19] utiliza el concepto de interacción entre partículas y define la actualización de la mejor posición  $Nbest$  ( $\Psi_l^N$ ), como aquella que maximiza la función fitness-distance-ratio:

$$FDR = \frac{Z(\Theta_l) - Z(\Psi_l)}{|\Psi_{jl} - \theta_{lh}|}$$

La inclusión de estos dos comportamientos modifica el cálculo de la velocidad para cada iteración del algoritmo, pues ésta tiene un término inercial y tantos términos como comportamientos de aprendizaje tengan las partículas.

El algoritmo propuesto para resolver el VRSPD se presenta a continuación. En primer lugar se inicializa el enjambre y se evalúa el desempeño de la función fitness mediante el método de codificación. De esta manera se actualiza la información de los comportamientos del enjambre. Posteriormente se actualiza la velocidad y la posición, se evalúa el criterio de parada y continúa con la siguiente iteración.

#### Algoritmo 1. PSO para VRSPD

Comenzar Programa

$\tau = 1$

para todas las partículas inicializar  $\Theta_l, \Omega_l = 0$  y  $\Psi_l = \Theta_l$

decodificar  $\Theta_l(\tau)$  en un conjunto de rutas  $R_l$

guardar las mejores posiciones  $pbest, gbest, lbest$  y generar  $nbest$

mientras ( $\tau < T$ )

actualizar inercia, velocidad y posición de las partículas

$$w(\tau) = w(T) + \frac{\tau - T}{1 - T} [w(1) - w(T)]$$

$$\omega_{lH}(\tau + 1) = w(\tau)\omega_{lH}(\tau) + c_p u(\Psi_{lH} - \theta_{lH}(\tau)) + c_g u(\Psi_{gH} - \theta_{lH}(\tau)) + c_l u(\Psi_{lH}^L - \theta_{lH}(\tau)) + c_n u(\Psi_{lH}^N - \theta_{lH}(\tau))$$

$$\theta_{lH}(\tau + 1) = \theta_{lH}(\tau) + \omega_{lH}(\tau + 1)$$

Evaluar y decodificar el nuevo  $\Theta_l(\tau)$  en un conjunto de rutas  $R_l$

Actualizar y guardar las mejores soluciones del enjambre  $pbest, gbest, lbest$  y generar  $nbest$

$\tau = \tau + 1$

fin\_mientras

fin\_programa

### 1.1.1. REPRESENTACIÓN DE LA SOLUCIÓN

La representación de la solución del VRSPD a través del paradigma PSO consiste de dos partes en las cuales se usa la posición de las partículas con múltiples dimensiones para representar una solución en un vector codificado, cuyos valores son números reales.

La primera parte de la representación consta de  $n$  dimensiones de la partícula (dimensiones en el espacio de búsqueda), en la que cada dimensión es asignada a un cliente. El menor valor de la dimensión corresponde a la mayor prioridad de asignación.

La segunda parte de la representación está basada en la orientación de rutas de los vehículos. Esta orientación de rutas se define como un punto en el mapa de servicio que representa cierta área en la cual el vehículo tiende a servir. Así, un vehículo tendrá una mayor tendencia a visitar a los clientes que estén en los alrededores de su correspondiente punto de orientación. En la Figura 1 se muestra la relación del área que recorre un vehículo representada por su punto de orientación.

Un punto de orientación de la ruta se define por un par de coordenadas  $x$ - $y$  en el mapa. Cada punto de orientación se representa por dos dimensiones de una partícula, una dimensión para el valor de la coordenada en  $x$  y otra dimensión para el valor de la coordenada en  $y$ . De esta manera, la representación deberá consistir en  $2m$  dimensiones de las partículas que corresponden a la flota de  $m$  vehículos disponibles.

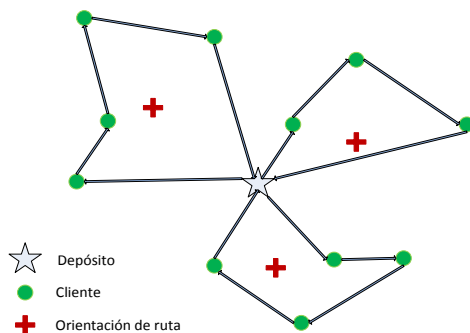


Figura 1. Rutas y puntos de orientación de rutas

Una vez todos los puntos de orientación de las rutas se identifican, se determina la preferencia de los vehículos para servir a cada uno de los clientes, basado en la distancia del cliente al punto de orientación. Estas preferencias se definen para asegurar la cercanía espacial

entre los clientes de una ruta, teniendo en cuenta que la cercanía entre los clientes y el punto de referencia de la ruta se mantiene. Así, la distancia total de la ruta será más corta y su correspondiente costo será minimizado.

Por lo tanto, la representación de la solución para el VRSPD para  $n$  clientes y  $m$  vehículos requiere partículas con  $n + 2m$  dimensiones. Las primeras  $n$  dimensiones corresponden al número de clientes y las siguientes  $2m$  relacionadas con la flota de vehículos. Por ejemplo, si el problema consta de 8 clientes y una flota de 2 vehículos, las dimensiones del enjambre serán las mostradas en la Figura 2.



Figura 2. Vector posición de una partícula del enjambre

### 1.1.2. MÉTODO DE DECODIFICACIÓN

Ahora, para convertir la codificación de los componentes de la heurística en una solución al problema original, se utiliza un algoritmo de decodificación. Este consta de tres pasos:

1. En el primer paso, se construye la lista de prioridad de clientes siguiendo la regla de mayor prioridad al cliente con menor valor en la dimensión correspondiente, a partir de las primeras  $n$  dimensiones de la partícula. La manera más simple de implementar esta regla es ordenando en orden ascendente los valores de posición de las  $n$  dimensiones y tomando el índice de dimensión como la lista de prioridades.
2. El siguiente paso es extraer los puntos de orientación de las rutas y construir la matriz de prioridad de los vehículos. La matriz se construye a partir de la distancia relativa entre los puntos de orientación y los clientes, los cuales pueden ser todos graficados en un mismo plano cartesiano. Un cliente es servido, en primer lugar, por el vehículo cuyo punto de referencia esté más cercano. Cada columna de la matriz contiene la prioridad de los vehículos para clientes con la misma prioridad.

1. El último paso de decodificación es construir rutas con base en la lista de prioridad de clientes y la matriz de prioridad de vehículos. Uno a uno, los clientes en la lista de prioridad de vehículos son asignados a los vehículos según su prioridad y sus restricciones de capacidad del vehículo y límite de duración de servicio. Cada nueva asignación de un cliente es insertada en la mejor posición de la ruta existente basado en el menor costo adicional, con la ayuda de la heurística de inserción más barata. Una vez el cliente sea asignado, se aplica una heurística de re-optimización para mejorar la ruta emergente. Esta heurística es el método conocido como 2-opt.

Este método es presentado de forma detallada en el algoritmo 2.

#### Algoritmo 1. Método de decodificación

```

Comenzar programa
  Construir lista de prioridad de Clientes ( $U$ )
    Construir el conjunto  $S = (1, 2, \dots, n)$  y  $U = \emptyset$ 
    mientras  $S \neq \emptyset$ 
      Seleccionar cliente  $c$  del conjunto  $S$  tal que corresponda a la menor dimensión de  $\theta_n$ 
      Agregar  $c$  a la última posición de  $U$ 
      Eliminar  $c$  del conjunto  $S$ .
    fin_mientras
  Construir matriz de prioridad de vehículos. ( $W$ )
    Para  $j = 1 \dots m$ , definir la posición de referencia de los  $m$  vehículos.
       $x^r e^f_j = \theta_{n+2j-1}$  y  $y^r e^f_j = \theta_{i,n+2j}$ 
    Para cada cliente  $i$ ,  $i = 1 \dots n$ 
      Calcular la distancia euclidiana entre el cliente  $i$  y los todos puntos de orientación
      mientras  $S \neq \emptyset$ 
        Construir conjunto  $S = (1, 2, \dots, m)$  y  $W_i = \emptyset$ 
        Seleccionar vehículo  $c$  del conjunto  $S$  tal que tenga la menor distancia euclidiana al cliente  $i$ .
        Agregar  $c$  a la última posición del conjunto  $W_i$ 
      fin_mientras
  Construir rutas de vehículos
     $k = 1$ 
    mientras  $k < n$ 
      agregar clientes uno a uno a la ruta
         $c = U_k$ ,  $b = 1$ 
        mientras  $b < m$ 
           $j = W_{c,b}$ 
          Hacer a  $c$  como candidato a insertar en la posición que genere el menor costo adicional en la ruta  $R_j$ , según heurística de inserción más barata.
          Evaluar la carga y el tiempo de servicio de la nueva ruta.
          si (cumple con condiciones de factibilidad)
            Actualizar la ruta  $R_j$  y reoptimizar con 2-opt
          fin_si
           $b = b + 1$ 
        fin_mientras
       $k = k + 1$ 
    fin_mientras
fin_programa

```

## 1. RESULTADOS COMPUTACIONALES

Los experimentos numéricos efectuados para evaluar el desempeño de la metaheurística propuesta se llevaron a cabo con la aplicación del método en un conjunto de datos de instancias desarrolladas para el VRSPD. Se trata de 18 instancias de gran tamaño introducidas por Tang y Galvão[3], quienes adaptaron instancias desarrolladas inicialmente para el CVRP por Solomon[20] y Gehring & Homberger[21], generando aleatoriamente demandas de recolección, en el mismo intervalo usado por Solomon para generar las demandas de entrega. La distancia de los arcos se obtiene calculando la distancia euclidiana entre los vértices, luego la matriz de distancias resultante es simétrica y triangular. El costo fijo por vehículo  $f$  y el costo variable por unidad de distancia  $g$  están definidos en 0 y 1 respectivamente. El límite de duración de servicio  $D$  se calcula generando una solución del Problema del Agente Viajero (TSP) con los clientes de cada una de las instancias, y el número de vehículos  $m$  se calcula luego de totalizar demandas de entrega y recogida de los clientes y dividir sobre la capacidad de los vehículos disponibles. La tabla 1 resume las características de las 18 instancias utilizadas, y pueden ser consultadas detalladamente en el enlace: <http://users.ntua.gr/ezach/>

El algoritmo es implementado en MatLab, versión R2012a, en un equipo con procesador Intel Core 2 Duo con 4 GB de memoria RAM instalada.

Los parámetros utilizados para la aplicación del algoritmo GLNPSO fueron establecidos a partir de resultados de experimentos preliminares realizados para observar el comportamiento del algoritmo en el trabajo de Moghaddam et al.[22], los cuales se resumen en la tabla 2.

Se obtuvieron soluciones para cada instancia utilizando diferente número de iteraciones. En las tablas 3, 4 y 5 se muestra la comparación de los resultados obtenidos con el algoritmo GLNPSO propuesto y la mejor solución conocida para el grupo de instancias mencionado para 100, 200 y 400 clientes respectivamente.

En la tabla 6 se muestra la diferencia de la mejor solución obtenida con el algoritmo GLNPSO propuesto, con los resultados de la búsqueda tabú propuesta por Tang-Montanè y Galvão[3], la metodología híbrida GLS – Tabu Search de Zachariadis et al. [23], el método evolucionario VLBR de Zachariadis et al [24], y el método APA de Zachariadis et al. [16].

**Tabla 1. Instancias para VRPSPD**

<i>Instancia</i>	<i>n</i>	<i>Q</i>	<i>D</i>	<i>P</i>
r101	100	200	1458	2339
r201	100	1000	1458	2262
c101	100	200	1810	3070
c201	100	700	1810	2910
rc101	100	200	1724	1912
rc201	100	1000	1724	2076
R1_2_1	200	200	3513	4406
R2_2_1	200	1000	3513	4358
C1_2_1	200	200	3530	5370
C2_2_1	200	700	3770	6010
RC1_2_1	200	200	3558	4473
RC2_2_1	200	1000	3558	4299
R1_4_1	400	200	7109	10433
R2_4_1	400	1000	7109	9571
C1_4_1	400	200	7190	12470
C2_4_1	400	700	7560	10050
RC1_4_1	400	200	7127	10065
RC2_4_1	400	1000	7127	10100

**n:** número de clientes, **Q:** Capacidad del vehículo, **D:** Demanda total de entrega, **P:** Demanda total de recogida.

**Tabla 2. Parámetros para el GLNPSO**

<i>Parámetros</i>	<i>Valor</i>
Número de partículas	$N = 50$
Número de vecinos	$K = 5$
Peso inercial Inicial	$w(1) = 0.9$
Peso inercial Final	$w(T) = 0.4$
Constante de aceleración Pbest	$c_p = 0,7$
Constante de aceleración Gbest	$c_g = 0,3$
Constante de aceleración Lbest	$c_l = 1,5$
Constante de aceleración Nbest	$c_n = 1,5$

**Tabla 3. Resultados para instancias de 100 clientes**

<i>Problemas de 100 clientes</i>							
<i>instancia</i>	<i>Q</i>	<i>Mejor solución Literatura</i>		<i>Mejor solución GLNPSO</i>		<i>Gap (%)</i>	
		<i>Z</i>	<i>k</i>	<i>Z</i>	<i>k</i>	<i>Z</i>	<i>K</i>
C1_2_1	200	3641,89	28	4463.2	31	0.226	0.11
C2_2_1	700	1726,59	9	1884.7	9	0.092	0.00
R1_2_1	200	3316,94	23	4107.5	25	0.217	0.09
R2_2_1	1000	1665,58	5	1763.20	5	0.059	0
RC1_2_1	200	3316,94	23	4013.5	25	0.210	0.087
RC2_2_1	1000	1560	5	2041.3	5	0.309	0.00

En los resultados obtenidos para las instancias de 100 clientes, se pudo observar que la desviación entre los resultados obtenidos y los mejores resultados obtenidos en la literatura no superan el 18% para la función de costos. También se puede observar que los menores valores de desviación corresponden a los problemas donde la capacidad de la flota es grande, en las que a su vez se encuentra el mejor número de vehículos en la solución.

**Tabla 4. Resultados para instancias de 200 clientes**

<i>Problemas de 200 clientes</i>							
<i>instancia</i>	<i>Q</i>	<i>Mejor solución Literatura</i>		<i>Mejor solución GLNPSO</i>		<i>Gap (%)</i>	
		<i>Z</i>	<i>k</i>	<i>Z</i>	<i>k</i>	<i>Z</i>	<i>K</i>
r101	200	1009,95	12	1135.1	13	0.124	0.08
r201	1000	666,2	3	671.17	3	0.007	0
c101	200	1220,18	16	1433.4	17	0.175	0.06
c201	700	662,07	5	664.22	5	0.003	0
rc101	200	1059,32	10	1218.3	11	0.150	0.10
rc201	1000	672,92	3	675.05	3	0.003	0

En los resultados obtenidos para las instancias de 200 clientes, se pudo observar que la desviación entre los resultados obtenidos y los mejores resultados obtenidos en la literatura no superan el 31% para la función de costos. El resultado con mayor desviación corresponde a la instancia RC2\_2\_1, para la cual, sin embargo, se encontró el mismo número de vehículos de la mejor solución conocida. También se puede observar que los menores valores de desviación corresponden a los problemas C2\_2\_1 y R2\_2\_1, donde la capacidad de la flota es grande, en las que a su vez se encuentra el mejor número de vehículos en la solución.

**Tabla 5. Resultados para instancias de 400 clientes**

<i>Problemas de 400 clientes</i>							
<i>instancia</i>	<i>Q</i>	<i>Mejor solución Literatura</i>		<i>Mejor solución GLNPSO</i>		<i>Gap (%)</i>	
		<i>Z</i>	<i>k</i>	<i>Z</i>	<i>k</i>	<i>Z</i>	<i>K</i>
C1_4_1	200	11125,14	63	13800	68	0.240	0.079
C2_4_1	700	3549,2	15	4424.3	16	0.247	0.067
R1_4_1	200	9668,18	53	11891	59	0.230	0.113
R2_4_1	1000	3560,73	10	3997.7	11	0.123	0.10
RC1_4_1	200	9520,06	51	12113	55	0.272	0.078
RC2_4_1	1000	3414,9	11	-	-	-	-

En los resultados obtenidos para las instancias de 400 clientes, se pudo observar en general una mayor desviación entre los resultados obtenidos y los mejores resultados obtenidos en la literatura que en las instancias con menos clientes. Sin embargo, ninguno de las desviaciones supera el 28% para la función de costos.

En cuanto al número de vehículos, se observa que en ningún caso se encontró el mejor número conocido, por lo que se sospecha que la obtención de un buen número de vehículos esté afectada por el número de clientes que tenga la instancia. La desviación para este caso nunca fue mayor del 10%.

En cuanto a la instancia RC2\_4\_1, no se logró obtener resultados luego de correr el programa por alrededor de 15 minutos.

**Tabla 1. Comparación resultados con otros métodos para resolver el VRPSPD**

i.	TG		GTS		VLBR		APA		GLNPSO	
	Z	k	Z	k	Z	k	Z	k	Z	k
01	1042,62	12	1019,48	12	1009,95	12	1009,95	12	1135,1	13
01	671,03	3	666,2	3	666,2	3	666,2	3	671,17	3
01	1259,79	17	1220,99	16	1220,99	16	1220,18	16	1433,4	17
01	666,01	5	662,07	5	662,07	5	662,07	5	664,22	5
101	1094,15	11	1059,32	10	1059,32	10	1059,32	10	1218,3	11
201	674,46	3	672,92	3	672,92	3	672,92	3	675,05	3
21	3447,2	23	3393,31	23	3376,3	23	3375,19	23	4107,5	25
21	1690,67	5	1673,65	5	1665,58	5	1665,58	5	1763,2	5
21	3792,62	29	3652,76	28	3643,82	28	3641,89	28	4463,2	31
21	1767,58	9	1735,68	9	1726,59	9	1726,73	9	1884,7	9
2121	3427,19	24	3341,25	23	3323,56	23	3316,94	23	4013,5	25
221	1645,94	5	1562,34	5	1560	5	1560	5	2041,3	5
41	10027,81	54	9758,77	54	9691,6	54	9668,18	53	11891	59
41	3695,26	10	3606,72	10	3572,38	10	3560,73	10	3997,7	11
41	11676,27	65	11207,37	63	11179,36	63	11125,14	63	13800	68
41	3732	15	3630,72	15	3549,27	15	3549,2	15	4424,3	16
141	9883,31	52	9697,65	52	9645,27	52	9520,06	51	12113	55
241	3603,53	11	3498,3	11	3423,62	11	3414,9	11	-	-

A partir de los resultados, se realizaron procedimientos estadísticos para verificar el comportamiento del algoritmo cuando se varían algunos parámetros tanto de la metaheurística PSO como del problema VRPSPD. En primer lugar se evaluó el efecto de variar el número de iteraciones que usa el algoritmo propuesto como criterio de parada. Después se verificó si existe algún efecto importante de la capacidad de vehículos y del número de clientes, como parámetros del problema VRPSPD, así como de su interacción en la obtención de soluciones cercanas a la mejor solución conocida para el problema.

En todos casos, el desempeño del algoritmo se evaluó mediante una métrica definida para el análisis del algoritmo que consiste en el error porcentual entre las soluciones del algoritmo GLNPSO propuesto y las mejores soluciones conocidas de las instancias.

$$Desv_{opt} = \frac{Z_{GLNPSO} - Z_{best}}{Z_{best}}$$

Adicionalmente, en el análisis del efecto de la capacidad, y el efecto del número de clientes, se evaluó también el impacto sobre la desviación en el número de vehículos asignados como parte de la solución que arroja el algoritmo.

$$Desv_K = \frac{K_{GLNPSO} - K_{best}}{K_{best}}$$

Se realizó el análisis de esta manera, consistentemente con los trabajos previos para resolver el VRPSPD en los cual se ignoró el costo fijo de los vehículos. Sin embargo, se ha probado la importancia del costo fijo de los vehículos y se ha considerado como un componente importante de la función objetivo a lo largo de varios estudios en torno a otras áreas del VRP, y es por esta razón que se analizó la variable de salida relacionada con el número de vehículos para evaluar el comportamiento del algoritmo.

*Análisis de varianza del número de iteraciones*

Debido a que los parámetros seleccionados de la literatura para el GLNPSO no incluyeron el número de iteraciones, se realizó un análisis estadístico para determinar el efecto que tienen los niveles de éste factor sobre el desempeño que arroja el algoritmo para solucionar el VRPSPD. Esta es una variable que depende del problema que se quiere resolver, y es importante determinar cuál es el valor más apropiado para aplicarlo al algoritmo respectivo[17]. Se definieron los niveles del número de iteraciones (10, 20, 30, 40, 50 y 60) y se seleccionaron 6 instancias en las cuales se realizó el análisis: dos para 100 clientes, dos para 200 clientes y dos para 400 clientes.

*Análisis factorial de efectos fijos*

Tras obtener los resultados de las iteraciones, se tuvo la sospecha de la existencia de un efecto en las soluciones arrojadas por el algoritmo causado por la variación en la capacidad de los vehículos y el número de clientes del problema en sus diferentes instancias. Para verificar esta hipótesis se efectuó un análisis estadístico diseñado para establecer si existe un efecto significativo de los factores Capacidad de vehículos y número de clientes en las soluciones obtenidas con el algoritmo, medidas por las desviaciones de las dos variables de salida con las mejores soluciones conocidas. Para este análisis se seleccionaron las instancias representativas de 100,200 y 400 clientes, como se muestra en la tabla 7

**Tabla 1. Instancias probadas en análisis estadístico**

<i>Instancias</i>	<i>Capacidad</i>	<i>Clientes</i>
r201	1000	100
c101	200	100
C221	1000	200
R121	200	200
R241	1000	400
R141	200	400

El análisis permitió establecer que la variación en el número de clientes tiene un impacto significativo sobre la variable de respuesta. Del análisis de varianza y de las pruebas de rangos múltiples se puede concluir que el algoritmo arroja soluciones más cercanas al mejor valor conocido cuando las instancias tienen vehículos con capacidad grande (1000 u.) que cuando tienen capacidad pequeña (200 u.). Esto podría deberse a que con una flota de gran capacidad se requerirán menos vehículos para atender a los clientes, y esto representa un menor costo computacional para el algoritmo.

Además, se observa también que, en general, la desviación de la solución para problemas con vehículos de capacidad de 200 clientes no varía con los diferentes niveles de número de clientes. Sin embargo, para problemas con vehículos de capacidad 1000 se logran encontrar mejores soluciones cuando hay 100 clientes que cuando hay 200 o 400 clientes.

## **1. CONCLUSIONES Y TRABAJOS FUTUROS**

En la mayoría de las instancias en las cuales se hizo el análisis estadístico, se encontró que la utilización de diferentes números de iteraciones definidos como criterio de parada influye en la solución que arroja el algoritmo. En promedio, se obtuvo que con 50 iteraciones, el programa arroja soluciones con una menor desviación respecto a la mejor solución conocida del problema, comparado con los resultados obtenidos con otros número de iteraciones.

Del análisis factorial de los efectos fijos, se pudo comprobar que el tamaño del problema influye significativamente en la eficiencia del algoritmo para encontrar una buena solución. Para problemas de gran número de clientes, el algoritmo tuvo una mayor dificultad en encontrar soluciones que tuvieran menor desviación en su costo, respecto a las mejores soluciones conocidas.

También, se pudo concluir que para todos los tamaños del problema, la capacidad de la flota de vehículos disponible, es una variable que tiene un impacto significativo en la obtención de buenas soluciones. Dado

que la capacidad de los vehículos es una restricción del problema, la complejidad aumenta cuando ésta se hace menor.

La interacción entre tamaño del problema y capacidad de vehículos implica una variación significativa en la obtención de resultados con el algoritmo propuesto. Se puede concluir que el algoritmo obtiene mejores soluciones con problemas pequeños de hasta 200 clientes con una flota de vehículos de mayor capacidad. A medida que el problema aumenta de tamaño y se reduce el umbral de la restricción de capacidad, el algoritmo se hace menos eficiente.

Para futuras investigaciones enfocadas a encontrar métodos de solución para el VRPSPD, se recomienda tener en cuenta el valor de costo fijo relacionado con la utilización de los vehículos ya que, aunque para las instancias estudiadas en este proyecto se fijó este valor en cero, previos estudios en torno a otras variables del VRP han mostrado que éste tiene un alto impacto en la función de costos para aplicaciones prácticas del problema. El método propuesto puede manejar fuertemente esta situación puesto que el modelo matemático y la programación del algoritmo consideran el costo fijo en la función objetivo.

Es importante considerar otros procedimientos para establecer el número inicial de vehículos necesarios para que la rutina usada para calcular el número apropiado de vehículos sea más eficiente y se encuentren mejores soluciones al problema.

De acuerdo a la formulación matemática del VRPSPD, es conveniente considerar los tiempos de servicio y las restricciones que estos representan, de manera que las soluciones que se obtengan con métodos Metaheurísticos sean más próximas a problemas reales.

Algoritmos adicionales dentro del método de decodificación que mejoren las rutinas de construcción de rutas pueden tener un impacto significativo en la búsqueda de nuevos algoritmos para resolver el VRPSPD. Dado que las rutas son fluctuantes, no se construyen progresivamente una a una y podrían tener un número diferente de clientes en cada una de ellas, es posible aplicar heurísticas de intercambio entre-rutas y no únicamente intra-rutas para mejorar las secuencias emergentes.

## REFERENCIAS

- [1] H. Min, "The multiple vehicle routing problem with simultaneous delivery and pickup points," *Transportation Research*, vol. 23, pp. 337–386, 1989.
- [2] J.-F. Cordeau, M. Gendreau, G. Laporte, J.-Y. Potvin, and F. Semet, "A guide to vehicle routing heuristics," *Journal of the Operational Research Society*, vol. 53, no. 5, pp. 512–522, May 2002.
- [3] F. Alfredo Tang Montané and R. D. Galvão, "A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service," *Computers & Operations Research*, vol. 33, no. 3, pp. 595–619, Mar. 2006.
- [4] T. J. Ai and V. Kachitvichyanukul, "A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery," *Computers & Operations Research*, vol. 36, no. 5, pp. 1693–1702, May 2009.
- [5] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, pp. 1942–1948, 1995.
- [6] F. P. Goksal, I. Karaoglan, and F. Altiparmak, "A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery," *Computers & Industrial Engineering*, Jan. 2012.
- [7] J. Dethloff, "Vehicle routing and reverse logistics: The vehicle routing problem with simultaneous delivery and pick-up," *OR Spektrum*, vol. 23, no. 1, pp. 79–96, Feb. 2001.
- [8] P. Toth and D. Vigo, "Models, relaxations and exact approaches for the capacitated vehicle routing problem," *Discrete Applied Mathematics*, vol. 123, no. 1–3, pp. 487–512, Nov. 2002.
- [9] S. Salhi and G. Nagy, "A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling," *Journal of the Operational Research Society*, vol. 50, no. 10, pp. 1034–1042, Oct. 1999.
- [10] G. Nagy and S. Salhi, "Heuristic Algorithms For Single And Multiple Depot Vehicle Routing Problems With Pickups And Deliveries," *European Journal of Operational Research*. ELSEVIER SCIENCE, 01-Apr-2005.
- [11] M. D. Amico, G. Righini, and M. Salani, "A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection," 2005.
- [12] N. a. Wassan, a. H. Wassan, and G. Nagy, "A reactive tabu search algorithm for the vehicle routing problem with simultaneous pickups and deliveries," *Journal of Combinatorial Optimization*, vol. 15, no. 4, pp. 368–386, Jun. 2007.
- [13] J.-F. Chen and T.-H. Wu, "Vehicle routing problem with simultaneous deliveries and pickups," *Journal of the Operational Research Society*, vol. 57, no. 5, pp. 579–587, Jul. 2005.
- [14] N. Bianchessi and G. Righini, "Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery," *Computers & Operations Research*, vol. 34, no. 2, pp. 578–594, Feb. 2007.
- [15] Y. Gajpal and P. Abad, "An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup," *Computers & Operations Research*, vol. 36, no. 12, pp. 3215–3223, Dec. 2009.
- [16] E. E. Zachariadis and C. T. Kiranoudis, "A local search metaheuristic algorithm for the vehicle routing problem with simultaneous pick-ups and deliveries," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2717–2726, Mar. 2011.
- [17] A. P. Engelbrecht, *Computational Intelligence*, Second Edi. Southern Gate, Chichester West Sussex PO19 \*SQ, England: John Wiley & Sons, Ltd ISBN 978-0-470-03561-0, 2007.
- [18] P. Pongchairerks and V. Kachitvichyanukul, "A NON-HOMOGENOUS PARTICLE SWARM OPTIMIZATION WITH MULTIPLE SOCIAL STRUCTURES," in *Proceedings of the 2005 International Conference on Simulation and Modeling V.*, 2005.
- [19] K. Veeramachaneni and L. A. Osadciw, "Optimization Using Particle Swarms with Near Neighbor Interactions," 2003.
- [20] M. M. Solomon, "Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints," *Operations Research*, vol. 35, no. 2, pp. 254–265, 1987.
- [21] H. Gehring, J. Homberger, and D.- Hagen, "A Parallel Two-phase Metaheuristic for Routing Problems with Time Windows Abstract :," Hagen, Germany, 1999.
- [22] B. F. Moghaddam, R. Ruiz, and S. J. Sadjadi, "Vehicle routing problem with uncertain demands: An advanced particle swarm algorithm," *Computers & Industrial Engineering*, vol. 62, no. 1, pp. 306–317, Feb. 2012.
- [23] E. E. Zachariadis, C. D. Tarantilis, and C. T. Kiranoudis, "A hybrid metaheuristic algorithm for the vehicle routing problem with simultaneous delivery and pick-up service," *Expert Systems with Applications*, vol. 36, no. 2, pp. 1070–1081, Mar. 2009.
- [24] E. E. Zachariadis, C. D. Tarantilis, and C. T. Kiranoudis, "An adaptive memory methodology for the vehicle routing problem with simultaneous pick-ups and deliveries," *European Journal of Operational Research*, vol. 202, no. 2, pp. 401–411, Apr. 2010.