

OPTIMIZACIÓN POR SIMULACIÓN BASADO EN UN SISTEMA EVOLUTIVO DE  
OPTIMIZACIÓN DE ENJAMBRE DE PARTÍCULAS PARA EL PROBLEMA DE  
RUTEO DE VEHÍCULOS CON DEMANDAS ESTOCÁSTICAS

SILVIA ADRIANA GALVÁN NÚÑEZ

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

2012

OPTIMIZACIÓN POR SIMULACIÓN BASADO EN UN SISTEMA EVOLUTIVO DE  
OPTIMIZACIÓN DE ENJAMBRE DE PARTÍCULAS PARA EL PROBLEMA DE  
RUTEO DE VEHÍCULOS CON DEMANDAS ESTOCÁSTICAS

SILVIA ADRIANA GALVÁN NÚÑEZ

Trabajo de investigación para optar el título de Magíster en Ingeniería Industrial

Director

HENRY LAMOS DÍAZ, Ph.D.

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

2012

## TABLA DE CONTENIDO

INTRODUCCIÓN .....	10
1. REVISIÓN DE LA LITERATURA .....	13
1.1 ESTADO DEL ARTE .....	13
1.2 ANÁLISIS DE LA REVISIÓN DE LA LITERATURA .....	23
2. DEFINICIÓN DEL PROBLEMA.....	25
3. ENFOQUE PARA LA SOLUCIÓN DEL VRPSD .....	29
3.1 OPTIMIZACIÓN DE ENJAMBRE DE PARTÍCULAS (PSO) .....	31
3.2 OPTIMIZACIÓN DE ENJAMBRE DE PARTÍCULAS CON OPERADORES EVOLUTIVOS PARA EL VRPSD.....	36
4. RESULTADOS.....	45
4.1 CONFIGURACIÓN DE LOS PARÁMETROS DEL EPSO.....	45
4.2 TEST BED .....	45
4.3 DISEÑO EXPERIMENTAL.....	46
5. DISCUSIÓN DE LOS RESULTADOS.....	49
5.1 ANÁLISIS DE LOS RESULTADOS DEL DISEÑO EXPERIMENTAL .....	49
5.2 COMPARACIÓN DEL RENDIMIENTO DEL ALGORITMO.....	51
6. CONCLUSIONES .....	53
7. RECOMENDACIONES .....	54
REFERENCIAS BIBLIOGRÁFICAS.....	55

## LISTA DE FIGURAS

Figura 1. Ilustración de la política a priori .....	15
Figura 2. Ilustración de la política de descarga/abastecimiento preventivo .....	21
Figura 3. Esquema general del enfoque de optimización por simulación .....	30
Figura 4. Framework SIM-EPSo para el VRPSD .....	31
Figura 5. Esquema general del PSO .....	35
Figura 6. Diagrama Pareto para los efectos sobre la función objetivo .....	50

## LISTA DE TABLAS

Tabla 1. Ejemplo para la conversión de las soluciones en valores posición continuos .....	37
Tabla 2. Ejemplo para la aplicación del ROV .....	38
Tabla 3. Definición de los factores y niveles del diseño experimental .....	46
Tabla 4. Diseño experimental y resultados de SIM-EPSO.....	48
Tabla 5. Efectos estimados para la función objetivo .....	49
Tabla 6. Análisis de Varianza para la función objetivo.....	51
Tabla 7. Resultados obtenidos con EPSO y PSO.....	52

## RESUMEN

TÍTULO: OPTIMIZACIÓN POR SIMULACIÓN BASADO EN UN SISTEMA EVOLUTIVO DE OPTIMIZACIÓN DE ENJAMBRE DE PARTÍCULAS PARA EL PROBLEMA DE RUTEO DE VEHÍCULOS CON DEMANDAS ESTOCÁSTICAS<sup>1</sup>

AUTOR: SILVIA ADRIANA GALVÁN NÚÑEZ<sup>2</sup>

PALABRAS CLAVE: VRPSD, PSO, OPERADORES EVOLUTIVOS, SIMULACIÓN MONTE CARLO.

En el presente trabajo se resuelve el Problema de Ruteo de Vehículos con Demandas Estocásticas (VRPSD), el cual se analizó utilizando la política de descarga preventiva para el caso de un solo vehículo. Adicionalmente, se propuso el framework SIM-EPHO para la solución del VRPSD basado en el enfoque de optimización por simulación utilizando el híbrido de Optimización de Enjambre de Partículas con los operadores evolutivos de cruce y mutación (EPHO) y simulación de Monte Carlo para la evaluación de la función objetivo. Por otro lado, se llevó a cabo un diseño experimental (DOE) tipo Taguchi sobre un conjunto de instancias cuidadosamente generadas con el fin de analizar el impacto de los parámetros de entrada del VRPSD sobre la función objetivo. Adicionalmente, se compararon los resultados obtenidos en el framework propuesto con las soluciones dadas en la versión básica de la metaheurística Optimización de Enjambre de Partículas (PSO). Los resultados muestran que existen algunos parámetros del VRPSD que tienen una incidencia positiva sobre el valor de la función objetivo al cambiar de un nivel bajo a un nivel alto, como el número de clientes, capacidad del vehículo y la desviación de la demanda de los clientes. Finalmente, los resultados evidencian la capacidad del SIM-EPHO en encontrar mejores soluciones en términos de valor de función objetivo respecto a la versión básica del PSO, justificando la incorporación de los operadores evolutivos al método de solución propuesto.

---

<sup>1</sup> Trabajo de Grado

<sup>2</sup> Facultad de Ingenierías Fisicomecánicas. Escuela de Estudios Industriales y Empresariales.  
Director: Henry Lamos Díaz, Ph.D.

## ABSTRACT

TITLE: SIMULATION-OPTIMIZATION BASED ON AN EVOLUTIONARY SYSTEM OF PARTICLE SWARM OPTIMIZATION FOR THE VEHICLE ROUTING PROBLEM WITH STOCHASTIC DEMANDS<sup>1</sup>

AUTHOR: SILVIA ADRIANA GALVÁN NÚÑEZ<sup>2</sup>

KEY WORDS: VRPSD, PSO, EVOLUTIONARY OPERATORS, MONTE CARLO SIMULATION

In this paper the Vehicle Routing Problem with Stochastic Demands (VRPSD) is solved. The VRPSD was analyzed using the preventive restocking policy for the case of a single vehicle. In addition, the framework SIM-EPHO was proposed for the solution of the VRPSD using an approach based on simulation-optimization in which a hybrid Particle Swarm Optimization with evolutionary operators' crossover and mutation (EPHO) and Monte Carlo simulation to evaluate the objective function were considered. Furthermore, it was carried out a design of experiments (DOE) Taguchi type on a set of instances carefully generated in order to analyze the impact of the input parameters of the VRPSD on the objective function. Finally, the results obtained with the proposed framework were compared with the given solutions in the basic version of the metaheuristic Particle Swarm Optimization (PSO). The results show that there are some parameters VRPSD that have a positive impact on the value of the objective function to change from a low to a high level, like number of customers, vehicle capacity and demand's spread. Finally, results show the capacity of the framework SIM-EPHO of find better solutions in terms of objective function values than the basic PSO, justifying the incorporation of evolutionary operators to the solution method proposed.

---

<sup>1</sup> Research work

<sup>2</sup> Faculty of Physical-Mechanical Engineering, School of Business and Industrial Studies. Advisor: Henry Lamos Díaz, Ph.D.

## INTRODUCCIÓN

El diseño de rutas vehiculares es una de las funciones más críticas dentro del sistema de transporte terrestre enmarcado en la optimización de la cadena de abastecimiento. Es razonable pensar que a medida que van surgiendo nuevos productos, nuevas marcas, el nivel de exigencia de los clientes va incrementando, aumentándose de esta forma la necesidad de contar con un servicio al cliente más efectivo. En este orden de ideas, la selección, adquisición, implementación y evaluación de herramientas computacionales para el diseño de rutas, son consideradas como actividades relevantes dentro del proceso de búsqueda de ventajas competitivas en las organizaciones. Es por ello que el mejoramiento del sistema de decisiones para la distribución de bienes y servicios a través del diseño de rutas vehiculares, representa una oportunidad para reducir los costos fijos y operativos, aumentando de esta manera los niveles de servicio del sistema de transporte y por ende, la competitividad empresarial.

Tradicionalmente, las decisiones de distribución en el transporte terrestre se han abordado considerando los parámetros del modelo de manera determinista; como el costo, la demanda de los clientes, los tiempos de entrega, los tiempos de viaje de los vehículos, entre otros. Sin embargo, cuando se desean abordar situaciones prácticas, el problema de ruteo de vehículos se aparta del modelo idealizado, para incorporar modelos más flexibles que respondan al ambiente dinámico y complejo de los sistemas de transporte actuales, considerando entonces el diseño de rutas bajo incertidumbre de algunos o de todos los parámetros. Estos modelos son

llamados estocásticos. El problema de ruteo de vehículos con uno o más parámetros estocásticos se denota como SVRP tal y como lo presenta Bianchi<sup>1</sup>.

En la literatura se conocen tres variaciones importantes del SVRP presentadas por Gendrau, Laporte y Séguin<sup>2</sup>. Bertsimas<sup>3</sup> resolvió el SVRP considerando como elemento aleatorio el número de clientes; en este caso se tiene un Problema de Ruteo de Vehículos con Clientes Estocásticos (VRPSC). Por su parte, Gendrau, Laporte y Séguin<sup>4</sup>; y Kenyon y Morton<sup>5</sup> establecieron la situación en la cual el tiempo de viaje no es determinístico, el problema se denomina Problema de Ruteo de Vehículos con Tiempos de Viaje Estocástico (VRPSTT). Por último, Tillman<sup>6</sup> presentó el caso en el que la demanda de todos o de algunos clientes no es conocida al momento de diseñar la ruta; esta variante se conoce como Problema de Ruteo de Vehículos con Demandas Estocásticas (VRPSD).

Un aspecto importante en el desarrollo de nuevos algoritmos en el Problema de Ruteo de Vehículos es la tendencia a crear algoritmos a través del modelado basado en la inteligencia biológica y natural. Estos algoritmos inteligentes incluyen

---

<sup>1</sup> BIANCHI, Leonora., et al. Metaheuristics for the Vehicle Routing Problem with Stochastic Demands. En: Journal of Mathematical Modelling and Algorithms [en línea]. Diciembre 2005. Vol. 5, no. 1, pp. 91–110. DOI 10.1007/s10852-005-9033-y. Disponible en: <http://www.springerlink.com/index/10.1007/s10852-005-9033-y>.

<sup>2</sup> GENDREAU, Michel, LAPORTE, Gilbert y SÉGUIN, René, 1996. Stochastic vehicle routing. En: European Journal of Operational Research [en línea]. Enero 6 de 1996. Vol. 88, no. 1, pp. 3–12. DOI 10.1016/0377-2217(95)00050-X. Disponible en: <http://linkinghub.elsevier.com/retrieve/pii/037722179500050X>.

<sup>3</sup> BERTSIMAS, D. J. A Vehicle Routing Problem with Stochastic Demand. En: Operations Research [en línea]. Mayo 1992. Vol. 40, no. 3, pp. 574–585. DOI 10.1287/opre.40.3.574. Disponible en: <http://or.journal.informs.org/cgi/doi/10.1287/opre.40.3.574>.

<sup>4</sup> Ibid

<sup>5</sup> KENYON, Astrid S. y MORTON, David P. Stochastic Vehicle Routing with Random Travel Times. En: Transportation Science [en línea]. Febrero 2003. Vol. 37, no. 1, pp. 69–82. DOI 10.1287/trsc.37.1.69.12820. Disponible en: <http://transci.journal.informs.org/cgi/doi/10.1287/trsc.37.1.69.12820>.

<sup>6</sup> TILLMAN, Frank A. The Multiple Terminal Delivery Problem with Probabilistic Demands The Multiple Terminal Delivery Problem with Probabilistic Demands. En: Transportation Science. 1969. Vol. 3, no. 3, pp. 192–204.

entre otros, redes neuronales, la computación evolutiva, inteligencia de partículas, sistemas inmunes artificiales y sistemas difusos. Bajo el paradigma de Inteligencia de Partículas (SI), se encuentran los procedimientos de comportamiento social de organismos tales como colonias de hormigas (ACO) y enjambres de partículas (PSO) como métodos para abordar la solución de problemas optimización combinatoria con altos requerimientos computacionales.

La investigación se enfoca en crear un framework que combine la metaheurística híbrida Optimización de Enjambre de Partículas Evolutivo (EPSO) y la Simulación de Monte Carlo para la solución del VRPSD. Con el fin de determinar el impacto de los parámetros de entrada del VRPSD sobre la función objetivo, se realizará un diseño experimental tipo Taguchi sobre un *test bed* cuidadosamente generado. La validación de la eficacia y eficiencia del modelo propuesto, se hará a través de comparaciones del framework propuesto con respecto a las soluciones obtenidas utilizando la versión básica del PSO.

El documento está organizado de la siguiente manera. En la Sección 1 se encuentra el compendio de la revisión de la literatura realizada en la investigación. En la Sección 2 se define formalmente el VRPSD. En la Sección 3 se plantea y se explica detalladamente el framework SIM-EPSO para la solución del VRPSD. En la Sección 4 se describe el diseño experimental desarrollado y los resultados computacionales del algoritmo. En la Sección 5 se realiza la discusión formal de los resultados obtenidos. En la Sección 6 se presentan las conclusiones de la investigación realizada. Finalmente, en la Sección 7 se encuentran las referencias bibliográficas consultadas para el desarrollo de la investigación.

# 1. REVISIÓN DE LA LITERATURA

## 1.1 ESTADO DEL ARTE

Gendrau, Laporte y Séguin<sup>7</sup>; Bertsimas<sup>8</sup>; y Cordeau<sup>9</sup> et al, entre otros, consideran que el Problema de Ruteo de Vehículos, VRP, es un problema clásico en la literatura científica. Fisher<sup>10</sup>, en su trabajo, asume que en el VRP los costos, las demandas de los clientes y los tiempos de viaje de los vehículos son determinísticos. Sin embargo, el presente trabajo de investigación se ocupará de una variante del VRP en cual se considera que la demanda de los clientes sigue una distribución de probabilidad conocida. Bajo esta consideración, la demanda sólo será revelada en el momento en que el vehículo visita al cliente. En la literatura, este problema se conoce como el Problema de Ruteo de Vehículos con Demandas Estocásticas (VRPSD).

El VRPSD surge debido a situaciones reales de entrega o de recolección de mercancías en las cuales la empresa encargada de la distribución cuenta con clientes en donde sus demandas son inciertas. Tal y como lo presenta Manfrin<sup>11</sup> en su trabajo, en el caso determinista las rutas se planean de tal forma que los vehículos tengan suficiente capacidad para satisfacer las demandas de los

---

<sup>7</sup> Ibid., p 9.

<sup>8</sup> Ibid., p 9.

<sup>9</sup> CORDEAU, Jean-françois., et al. Vehicle Routing. En: Handbook in OR & MS. 14. S.I.: s.n. pp. 367–428. 1995.

<sup>10</sup> FISHER, Marshall. Vehicle Routing. En: Handbook in OR & MS, vol 8. S.I.: s.n. 1995.

<sup>11</sup> MANFRIN, Max. Ant Colony Optimization for the Vehicle Routing Problem. S.I.: Université Libre de Bruxelles. 2004.

clientes dadas unas rutas preestablecidas. Sin embargo, cuando las demandas son estocásticas, el concepto de “rutas preestablecidas” tiene una interpretación diferente y se requieren de reglas de decisión o *políticas de ruteo* que reinterpretan el concepto anterior.

Por su parte, Yang, Mathur y Ballou<sup>12</sup> expresan que para el VRPSD las rutas pueden no ejecutarse como se planearon debido a la incertidumbre de las demandas cuando el vehículo visita a un determinado cliente. Por consiguiente, al momento de la visita es posible que ocurra una de las siguientes situaciones: (i) el vehículo satisface completamente la demanda del cliente o (ii) la demanda del cliente excede la capacidad que lleva el vehículo en ese momento. En esta última situación, se dice que la ruta falla y se debe tomar una decisión o acción recursiva, la cual consiste en que el vehículo regrese al depósito, se reaprovisione y regrese al último cliente que no pudo atender; o bien, al visitar al cliente actual  $i$ , el vehículo bien puede continuar con la ruta asignada o regresar al depósito para un reabastecimiento preventivo. Teniendo en cuenta lo anterior, se observa que la distancia recorrida es un valor aleatorio debido a que no se conoce cuáles son los clientes en los cuales se deben tomar tales acciones.

Secomandi<sup>13</sup> clasifica las políticas de ruteo en tres enfoques:

- a) A priori (política estática, *offline*).
- b) Dinámica (política de reoptimización, *online*).
- c) Mixta (política de descargue/abastecimiento preventivo).

En primer lugar se encuentra la política estática o *a priori* la cual se enmarca dentro de los procesos estocásticos de dos estados, en donde en el primer estado

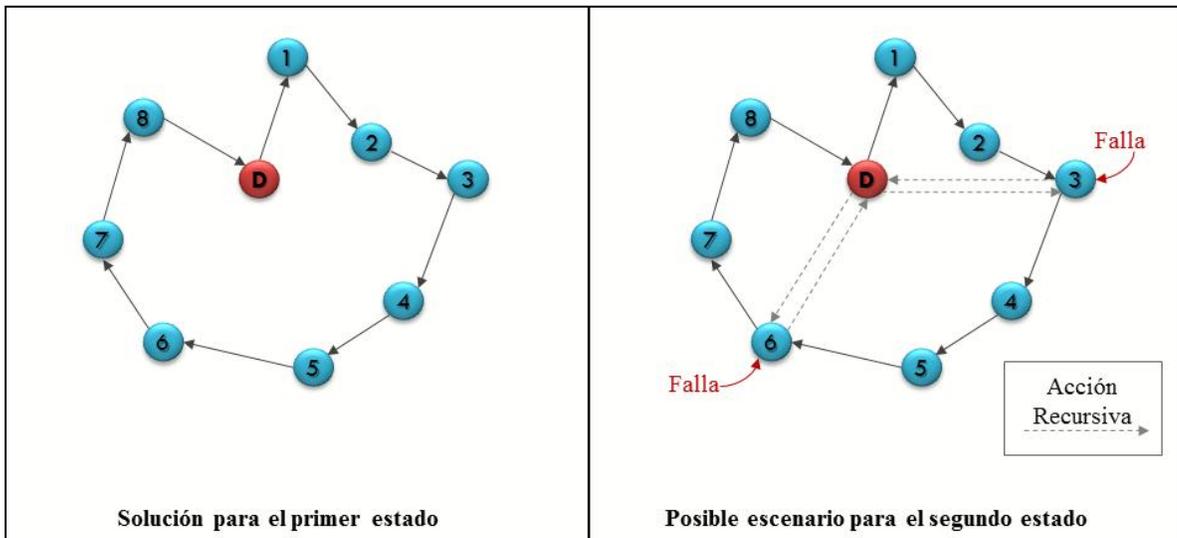
---

<sup>12</sup> YANG, Wen-Huei, MATHUR, K y BALLOU, R. Stochastic Vehicle Routing Problem with Restocking. En: Transportation Science [en línea]. Abril 2000. Vol. 34, no. 1, pp. 99–112. Disponible en: <http://www.ncbi.nlm.nih.gov/pubmed/20422950>.

<sup>13</sup> SECOMANDI, Nicola. A rollout policy for the vehicle routing problem with stochastic demands. En: Operations Research. 2001. Vol. 49, no. 5, pp. 796–802.

se determina una secuencia de clientes denotada por  $\tau$  (ruta a priori) los cuales deben ser visitados en ese orden por un vehículo y en el segundo estado se ejecuta la ruta tal y como se definió en el primer estado. En caso que la ruta falle, se toma una acción recursiva (Figura 1). Bertsimas<sup>14</sup> fue el primer autor en presentar una formulación para el VRPSD considerando la política de ruteo a priori. Este trabajo se centra en proponer inicialmente dos estrategias de actualización de la ruta con el fin de evitar el rediseño de las rutas cuando se necesite, esto es, después de haber diseñado la secuencia a priori, se aplica alguna de las dos estrategias dependiendo de la información que se tenga respecto a la demanda de los clientes.

Figura 1. Ilustración de la política a priori



Fuente: Autor

<sup>14</sup>Ibid.. p 9.

En la estrategia A, el vehículo visita todos los clientes en el orden definido por la secuencia a priori. En este caso, el costo total corresponde a la distancia total asociada a recorrer dicha secuencia más la distancia incurrida en caso que la ruta falle. Bajo la estrategia B, al igual que en la estrategia A, la ruta sigue la secuencia a priori, sólo que en caso que los clientes no tengan demanda en alguna instancia del problema, son excluidos de la ruta.

Adicionalmente en su trabajo, Bertsimas<sup>15</sup> propone una heurística cíclica obteniendo buenos resultados teniendo en cuenta la perspectiva del peor caso cuando la distribución de probabilidad de los clientes es idéntica. Además, demuestra que las soluciones obtenidas mediante las aproximaciones realizadas con la secuencia a priori son muy cercanas a las soluciones obtenidas con la estrategia de reoptimización.

Por otro lado, Bertsimas, Jaillet y Odoni<sup>16</sup> desarrollan y prueban diferentes heurísticas para el VRPSD a través de experimentos computacionales que son contrastados con las soluciones obtenidas bajo la estrategia de reoptimización. Las heurísticas son clasificadas en dos categorías: las primeras son técnicas de construcción de rutas entre las que se encuentran las heurísticas *The Space-filling Curve Heuristic*, *The Radial Sorting Heuristic*, *The 2-interchange: Homogeneous PTSP*, *The 2-interchange: Heterogeneous PTSP* y las segundas son procedimientos de mejoramiento local.

Otros autores han propuesto diferentes aproximaciones para resolver el VRPSD bajo esta política de ruteo. Gendreau, Laporte y Séguin<sup>17</sup> proponen la primera solución exacta del problema en la literatura a través del método entero *L-Shaped*;

---

<sup>15</sup> Ibid.. p 9.

<sup>16</sup> BERTSIMAS, D. J., JAILLET, P. y ODONI, a. R. A Priori Optimization. En: Operations Research [en línea]. Noviembre1990. Vol. 38, no. 6, pp. 1019–1033. DOI 10.1287/opre.38.6.1019. Disponible en: <http://or.journal.informs.org/cgi/doi/10.1287/opre.38.6.1019>.

<sup>17</sup> Ibid.. p 9.

Tan, Cheong y Goh<sup>18</sup> resuelven el problema utilizando un algoritmo evolutivo multiobjetivo el cual incorpora dos heurísticas de explotación local y un método de simulación para el cálculo del costo de la ruta. Por su parte, Rei, Gendreau y Soriano<sup>19</sup> utilizan un método exacto para la solución del problema en donde la evaluación de la función objetivo se realiza utilizando Simulación de Monte Carlo. Laporte, Louveaux y Van Hamme<sup>20</sup>; y Hjorring y Holt<sup>21</sup>, implementan el método *L-Shaped*. Mendoza et al<sup>22</sup> y Mendoza, Castanier y Guéret<sup>23</sup> resuelven el VRPSD con múltiples compartimentos utilizando Algoritmos meméticos y evolutivos y heurísticas constructivas respectivamente. Por su parte, Chepuri y Homem-de-Mello<sup>24</sup> consideran una versión alternativa al VRPSD en donde proponen una heurística basada en el método de entropía cruzada (*cross-entropy method*), incorporando además la simulación Monte Carlo con el propósito de mejorar el valor de la función objetivo en cada punto del dominio.

---

<sup>18</sup> TAN, K, CHEONG, C y GOH, C. Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation. En: European Journal of Operational Research [online]. 1 Marzo 2007. Vol. 177, no. 2, pp. 813–839. [Accessed 28 January 2011]. DOI 10.1016/j.ejor.2005.12.029. Disponible en: <http://linkinghub.elsevier.com/retrieve/pii/S0377221706000208>.

<sup>19</sup> REI, W., GENDREAU, M. y SORIANO, P. A Hybrid Monte Carlo Local Branching Algorithm for the Single Vehicle Routing Problem with Stochastic Demands. En: Transportation Science [en línea]. 22 Enero 2010. Vol. 44, no. 1, pp. 136–146. [Consultado Julio 2012]. DOI 10.1287/trsc.1090.0295. Disponible en: <http://transci.journal.informs.org/cgi/doi/10.1287/trsc.1090.0295>.

<sup>20</sup> LAPORTE, G, LOUVEAUX, Francois y VAN HAMME, Luc. An integer L-Shaped algorithm for the capacitated vehicle routing problem with stochastic demands. En: Operations Research. 2001. Vol. 50, no. 3, pp. 415–423.

<sup>21</sup> HJORRING, Curt y HOLT, John. New optimality cuts for a single-vehicle stochastic routing problem. En: Annals of Operations Research. 1999. Vol. 86, pp. 569–584.

<sup>22</sup> MENDOZA, Jorge E., et al. A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands. En: Computers & Operations Research [online]. Noviembre 2010. Vol. 37, no. 11, pp. 1886–1898. [Consultado Marzo 2011]. DOI 10.1016/j.cor.2009.06.015. Disponible en: <http://linkinghub.elsevier.com/retrieve/pii/S0305054809001713>.

<sup>23</sup> MENDOZA, Jorge E, CASTANIER, Bruno y GUÉRET, Christelle. Constructive Heuristics for the Multicompartment Vehicle Routing Problem with Stochastic Demands. En: Transportation Science. 2011. pp. 1–18. DOI 10.1287/trsc.1100.0353.

<sup>24</sup> CHEPURI, Krishna y HOMEM-DE-MELLO, Tito. Solving the Vehicle Routing Problem with Stochastic Demands using the Cross-Entropy Method. En: Annals of Operations Research [en línea]. Febrero 2005. Vol. 134, no. 1, pp. 153–181. DOI 10.1007/s10479-005-5729-7. Disponible en: <http://www.springerlink.com/index/10.1007/s10479-005-5729-7>.

En segundo lugar se encuentran las políticas dinámicas que consisten básicamente en que dado el estado actual del sistema, se determina cuál cliente se debería visitar después. Este tipo de políticas se formulan matemáticamente como un problema estocástico de múltiples estados.

Secomandi<sup>25</sup> considera el VRPSD para el caso de un solo vehículo teniendo en cuenta una política de reoptimización en donde en cada estado del problema (cada nodo), se define cuál es el siguiente cliente a visitar. Para este trabajo se lleva a cabo la metodología de Programación Neuro-Dinámica en donde se compararan dos algoritmos: la política de iteración optimista aproximada y la política *rollout*. Más adelante, Christiansen y Lysgaard<sup>26</sup> resuelven el Problema de Ruteo de Vehículos con Capacidad y Demandas Estocásticas (CVRPSD) por medio del método exacto branch and price el cual permite replantear la ruta en caso que ésta falle, es decir, cuando el vehículo no pueda satisfacer la demanda del cliente  $j$ . Para ello, consideran las dos estrategias planteadas por Bertsimas<sup>27</sup>, observando que la estrategia A (estrategia de visitar a todos los clientes sin importar si éste tiene o no demanda) es mejor que la estrategia B en donde por el contrario, no se visitan los clientes que no tienen demanda en determinado día. La instancia más grande que se ha corrido con el algoritmo propuesto consta de 60 clientes y 16 rutas. Asimismo, Secomandi y Margot<sup>28</sup> utilizan diferentes enfoques de reoptimización para resolver el VRPSD en donde se destacan las heurísticas de reoptimización parcial.

---

<sup>25</sup> Ibid.. p 12.

<sup>26</sup> CHRISTIANSEN, C y LYSGAARD, J. A branch-and-price algorithm for the capacitated vehicle routing problem with stochastic demands. En: Operations Research Letters [en línea]. Noviembre 2007. Vol. 35, no. 6, pp. 773–781. [Consultado 11 Abril 2011]. DOI 10.1016/j.orl.2006.12.009. Disponible en: <http://linkinghub.elsevier.com/retrieve/pii/S0167637707000168>.

<sup>27</sup> Ibid.. p 9.

<sup>28</sup> SECOMANDI, Nicola. y MARGOT, F. Reoptimization Approaches for the Vehicle-Routing Problem with Stochastic Demands. En: Operations Research [online]. 1 Enero 2009. Vol. 57, no. 1, pp. 214–230. [Consultado Julio 2012]. DOI 10.1287/opre.1080.0520. Disponible en: <http://or.journal.informs.org/cgi/doi/10.1287/opre.1080.0520>.

En su trabajo de investigación, Andreatta y Lulli<sup>29</sup> resuelven el TSP con demandas estocásticas urgentes y regulares. Las demandas urgentes son aquellas que tienen que ser satisfechas inmediatamente mientras que las regulares no tienen que ser atendidas inmediatamente sino que pueden esperar ser atendidas hasta el día siguiente. El problema se modela como un proceso de Markov con costos; sin embargo, como el número de estados del problema crece exponencialmente con el tamaño de la instancia del problema, los autores proponen un modelo de Markov agregado donde el número de estados del problema sólo crece en forma polinomial al incrementar el tamaño de la instancia y de esta forma encontrar una solución aproximada a la óptima.

Por otra parte, Novoa y Storer<sup>30</sup>; y Secomandi<sup>31</sup> desarrollan los algoritmos *rollout* para resolver el VRPSD con un solo vehículo bajo una aproximación dinámica. El algoritmo *rollout* es un tipo de política iterativa en la cual se mejoran secuencialmente una o múltiples políticas bases iniciales subóptimas; este algoritmo se puede extender a las secuencias bases (por ejemplo soluciones a priori), políticas *look-ahead* y esquemas reducidos. En el primer trabajo mencionado, los autores hacen tres aportes principales en su trabajo: en primer lugar, desarrollan un algoritmo *rollout* de dos pasos que proporciona soluciones 1,6% más económica que las dadas por el algoritmo *rollout* de un paso. En segundo lugar, usan simulación Monte Carlo para calcular el costo esperado de las secuencias bases actualizadas como una alternativa a los cálculos exactos. Y finalmente desarrollan secuencias bases mejoradas y esquemas reducidos

---

<sup>29</sup> ANDREATTA, G. y LULLI, G. A multi-period TSP with stochastic regular and urgent demands. En: European Journal of Operational Research [en línea]. Febrero 2008. Vol. 185, no. 1, pp. 122–132. [consultado 11 Abril 2011]. DOI 10.1016/j.ejor.2006.12.040. Disponible en: <http://linkinghub.elsevier.com/retrieve/pii/S0377221707000707>.

<sup>30</sup> NOVOA, C y STORER, R. An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. En: European Journal of Operational Research [en línea]. Julio 2009. Vol. 196, no. 2, pp. 509–515. [Consultado Junio 2010]. DOI 10.1016/j.ejor.2008.03.023. Disponible en: <http://linkinghub.elsevier.com/retrieve/pii/S0377221708003172>.

<sup>31</sup> Ibid.. p 12.

superando al costo de reducción en un 4% aproximadamente sobre métodos previos.

Ak y Erera<sup>32</sup> presentan una estrategia llamada *paired-vehicle strategy* para el VRPSD. Bajo esta estrategia, las rutas a priori de los vehículos se planean en par. Este esquema es también conocido como una estrategia E/2, en donde "E" indica que la acción de recursividad se aplica como consecuencia de un evento (que el vehículo alcance su capacidad) y "2" indica que la decisión de recursividad puede resultar en hacer de nuevo la planeación para dos vehículos simultáneamente. Para encontrar la solución inicial y hacer el replanteamiento de cada par de rutas, los autores utilizan la metaheurística Búsqueda Tabú.

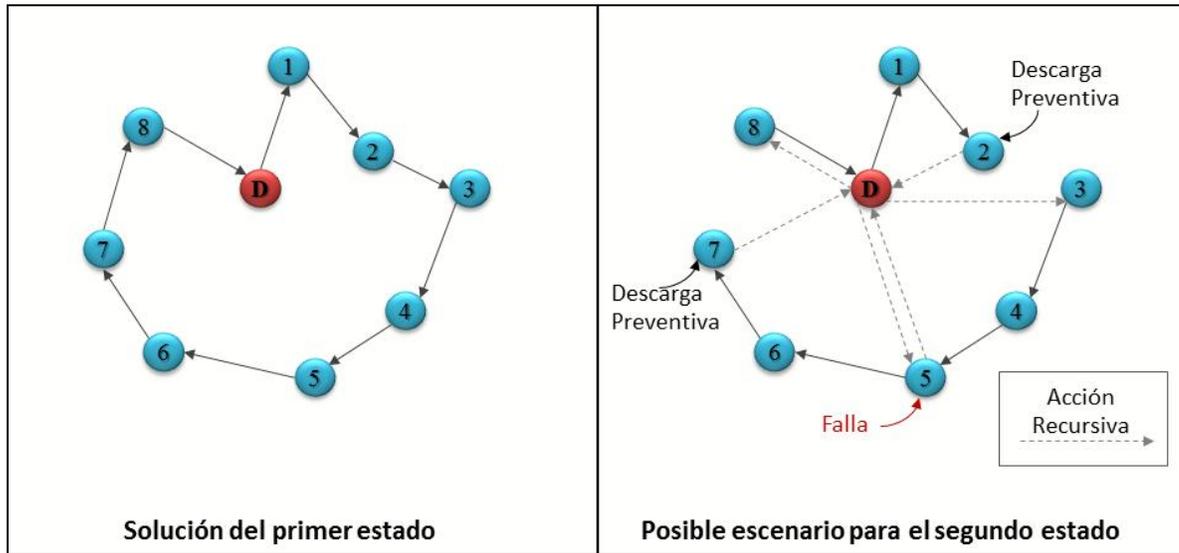
En tercer lugar se encuentran las políticas de descargue/abastecimiento preventivo o políticas mixtas. Las políticas mixtas combinan elementos de las políticas a priori y dinámica enmarcándose en un proceso estocástico de dos estados, en donde el vehículo sigue una ruta preestablecida (primer estado) y a la vez está habilitado con reglas dependientes de estado que permiten reaprovisionamientos anticipados (segundo estado). La Figura 2 representa la política. Yang, Mathur y Ballou<sup>33</sup> resuelven el VRPSD con descargas preventivas para el caso de contar con uno y varios vehículos. Para ello, desarrollan dos heurísticas: la primera *route first-cluster next* y la segunda llamada *cluster first-route next* con el fin de construir una y múltiples rutas que minimicen la distancia total recorrida.

---

<sup>32</sup> AK, A. y ERERA, L., A Paired-Vehicle Recourse Strategy for the Vehicle-Routing Problem with Stochastic Demands. En: *Transportation Science* [en línea]. Mayo 2007. Vol. 41, no. 2, pp. 222–237. DOI 10.1287/trsc.1060.0180. Disponible en: <http://transci.journal.informs.org/cgi/doi/10.1287/trsc.1060.0180>.

<sup>33</sup> Ibid.. p 12.

Figura 2. Ilustración de la política de descarga/abastecimiento preventivo



Fuente: Autor

Por otra parte, Bianchi et al<sup>34</sup> prueban el impacto en el desarrollo de las metaheurísticas Recocido Simulado, Búsqueda Tabú, Colonia de Hormigas y algoritmos evolutivos, el uso de la distancia de la ruta a priori como una aproximación rápida de la función objetivo. Fan y Wang<sup>35</sup> proponen una heurística para resolver el VRPSD con múltiples vehículos a través de la heurística *cluster first-route next* en donde la solución obtenida bajo dicha heurística es refinada por medio de una política de reoptimización basado en el algoritmo *rollout*.

Otro aporte realizado al problema estuvo a cargo de Ismail e Irhamah<sup>36</sup> los cuales resuelven el VRPSD basado en el híbrido Algoritmos Genéticos y Búsqueda Tabú

<sup>34</sup> Ibid.. p 9.

<sup>35</sup> FAN, Jianhua y WANG, Xiufeng. A Multiple Vehicles Routing Problem Algorithm with Stochastic Demand. En: 6th World Congress on Intelligent Control and Automation [en línea]. S.l.: Ieee. 2006. pp. 1688–1692. Disponible en: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1712640>.

<sup>36</sup> ISMAIL, Z . y IRHAMAH. Solving the Vehicle Routing Problem with Stochastic Demands via Hybrid Genetic Algorithm-Tabu Search. En: Journal of Mathematics and Statistics. 2008. Vol. 4, no. 3, pp. 161–167.

para una aplicación real relacionada con la recolección de basuras en un área residencial de Malasia. La solución inicial se determina usando dos heurísticas: la primera se llama el vecino aleatorio más cercano y la segunda es la inserción aleatoria más lejana.

Por otro lado, Tripathi, Kuriger y Wan<sup>37</sup> incorporan un elemento adicional para el cálculo de la función objetivo basado en la técnica de optimización por simulación. Para ello, presentan una aproximación para resolver el VRPSD teniendo en cuenta la búsqueda en el vecindario incrustado en un sistema de hormigas adaptativo y una aproximación de optimización de colonia de hormigas. Además, los autores usan diseños experimentales con el fin de comparar esta aproximación con los algoritmos de colonia de hormigas y los algoritmos genéticos, los cuales arrojaron una mayor robustez de la técnica propuesta frente a las dos metaheurísticas ya mencionadas. Shanmugam, Ganesan y Vanathi<sup>38</sup> resuelven el VRPSD empleando las metaheurísticas PSO, GA y PSO con operadores genéticos. Farhang, Ruiz y Jafar<sup>39</sup> proponen un método de decodificación de las soluciones obtenidas al usar PSO.

Por su parte, Bianchi et al<sup>40</sup> resuelven el VRPSD en donde la exploración del espacio de búsqueda se realiza mediante una hibridación entre metaheurísticas intercalando con el cálculo de la función objetivo del TSP.

---

<sup>37</sup> TRIPATHI, Mukul, KURIGER, Glenn y WAN, Hung-da. An Ant Based Simulation Optimization for the Vehicle Routing Problem with Stochastic Demands. En: Proceedings of the 2009 Winter Simulation Conference, IEEE. S.I.: s.n. 2009. pp. 2476–2487.

<sup>38</sup> SHANMUGAM, Geetha, GANESAN, Poonthalir y VANATHI, P T. Meta Heuristic Algorithms for Vehicle Routing Problem with Stochastic Demands. En: Journal of Computer Science. 2011. Vol. 7, no. 4, pp. 533–542.

<sup>39</sup> FARHANG, Babak, RUIZ, Rubén y JAFAR, Seyed. Vehicle routing problem with uncertain demands : An advanced particle swarm algorithm. En: Computers & Industrial Engineering [en línea]. 2011. pp. 1–12. DOI 10.1016/j.cie.2011.10.001. Disponible en: <http://dx.doi.org/10.1016/j.cie.2011.10.001>.

<sup>40</sup> BIANCHI, Leonora., et al. Hybrid Metaheuristics for the Vehicle Routing Problem with Stochastic Demands. Reporte Técnico. S.I. 2005.

## 1.2 ANÁLISIS DE LA REVISIÓN DE LA LITERATURA

Teniendo en cuenta la revisión en el estado del arte de los diferentes aportes realizados al VRPSD, se puede afirmar que para una instancia dada del VRPSD y para un conjunto de acciones recursivas, el conjunto de las políticas a priori es un subconjunto de las políticas de descargue/abastecimiento preventivo, que a su vez, están incluidas dentro de las políticas dinámicas. Dado que todas las políticas apuntan al mismo objetivo que es la minimización de los costos como por ejemplo la distancia total esperada, se podría pensar en principio que las políticas dinámicas ofrecen mejores soluciones sobre las demás políticas debido a su versatilidad a la hora de buscar una buena solución del problema.

Sin embargo, el costo computacional que se incurre por obtener esas buenas soluciones es considerablemente superior al de las demás políticas lo que hace que algunas compañías que presten el servicio de distribución o recolección de mercancías no dispongan de los recursos suficientes para reoptimizar las rutas en todo momento, o sencillamente la empresa tenga otras prioridades como la personalización del servicio visto desde una óptica diferente a una estrategia basada en la demanda.

Por otro lado, las investigaciones realizadas bajo la política de descargue/aprovisionamiento preventivo han venido evolucionando con mayor rapidez respecto a las políticas a priori y dinámica dado que se producen soluciones más estables y más predecibles debido a que se predetermina la secuencia de los clientes. Las soluciones que se obtienen al implementar esta política, son bastante competitivas en términos del valor esperado de la función objetivo y mucho más atractivas si se mira desde el punto de vista del costo computacional comparada con las políticas dinámicas.

Es por esto que la presente investigación se centra en la política de descargue/aprovisionamiento preventivo para el VRPSD y adicionalmente se pretende abordar, como método de solución, un enfoque basado en la optimización por simulación ya que a través de este enfoque, se han obtenido buenas estimaciones de la función objetivo, como se reflejan en los trabajos de Tripathi, Kuriger y Wan<sup>41</sup>; y Juan et al<sup>42</sup>. Para efectos del presente proyecto de investigación, el módulo de optimización está dado por un método de Optimización de Enjambre de Partículas Evolutivo y en el módulo de simulación se va a utilizar la simulación de Monte Carlo.

---

<sup>41</sup> Ibid.. p 20.

<sup>42</sup> ISMAIL, Z . y IRHAMAH. Solving the Vehicle Routing Problem with Stochastic Demands via Hybrid Genetic Algorithm-Tabu Search. En: Journal of Mathematics and Statistics. 2008. Vol. 4, no. 3, pp. 161–167.

## 2. DEFINICIÓN DEL PROBLEMA

En esta sección se define formalmente el VRPSD.

Sea  $G = \{V, A, C\}$  un grafo completo, donde:

- $V = \{0, 1, \dots, n\}$ , es el conjunto de nodos, el nodo 0 corresponde al depósito.
- $A = \{(i, j) : i, j \in V, i \neq j\}$ , es el conjunto de arcos que conecta a los nodos.
- $C = (c_{ij})$ , denota la matriz de costo de viaje entre los nodos  $i$  y  $j$ .

Se considera que la matriz  $C$  es simétrica y satisface la desigualdad triangular:  $c(i, j) \leq c(i, k) + c(k, j)$ . Además, todos los clientes tienen demandas estocásticas  $\xi_i, i = 1, \dots, n$ , las cuales tienen distribuciones de probabilidad independientes y conocidas y su valor sólo será revelado en el momento en que se visita al cliente. De igual manera se asume que el máximo valor de demanda del cliente  $i$  no excede la capacidad del vehículo  $Q$  y sigue una distribución de probabilidad discreta  $p_{ik} = \text{prob}(\xi_i = k), k = 0, 1, 2, \dots, K \leq Q, i = 1, \dots, n$ .

El objetivo del VRPSD considerando una política de descarga/abastecimiento preventivo consiste entonces en encontrar una ruta a priori y una política de descargue/abastecimiento preventivo en cada nodo que minimice el costo total esperado. Para ello, es importante hacer una aclaración de los costos que se consideran para este problema:

- Costo de viajar desde un cliente hacia otro como se planeó.
- Costo de descargue: costo de regresar al depósito para realizar el descargue antes de visitar el siguiente cliente planeado.

- Costo de falla de la ruta: costo de regresar al depósito para realizar el descargue causado por la insuficiencia en la capacidad del vehículo para atender la demanda del cliente a visitar.

Una ruta a priori se denota como  $1 \rightarrow 2 \rightarrow \dots \rightarrow n$  para un vehículo en particular. Después de haber atendido al cliente  $j$ , se asume que el vehículo tiene una carga residual  $q$ , esto es, la capacidad que tiene el vehículo después de haber visitado al cliente  $j$  y se define  $f_j(q)$  como el costo total esperado desde el nodo  $j$  hasta el final de la ruta. El costo esperado de la ruta planeada, esto es, el valor de la función objetivo, es  $f_0(Q)$ . Sea  $S_j$  el conjunto de todas las posibles cargas que un vehículo puede tener después de completar el servicio en el cliente  $j$ , el problema consiste entonces en encontrar una ruta a priori con el mínimo costo total esperado. El conjunto de ecuaciones (2.1)-(2.4) describen lo anteriormente mencionado.

$$f_j(q) = \min\{f_j^p(q), f_j^r(q)\}, \quad (2.1)$$

Donde

$$f_j^p(q) = c_{j,j+1} + \sum_{k:k \leq q} f_{j+1}(q-k)p_{j+1,k} + \sum_{k:k > q} [b + 2c_{j+1,0} + f_{j+1}(q+Q-k)]p_{j+1,k} \quad (2.2)$$

Y

$$f_j^r(q) = c_{j,0} + c_{0,j+1} + \sum_{k=1}^K f_{j+1}(Q-k)p_{j+1,k} \quad (2.3)$$

Con la condición de frontera

$$f_n(q) = c_{n,0} = c_{n,0} \quad q \in S_n \quad (2.4)$$

En las ecuaciones (2.1)-(2.3),  $f_j^p(q)$  representa el costo esperado de proceder directamente al siguiente nodo y  $f_j^r(q)$  representa el costo de la descarga/abastecimiento preventivo. Las anteriores ecuaciones se usan recursivamente con el propósito de determinar el valor de la función objetivo de la ruta planeada y la secuencia óptima de decisiones después de haber atendido a los clientes. La ecuación (2.4) representa la condición de frontera la cual quiere decir que el costo esperado después de haber visitado al último cliente de la ruta hacia adelante, independiente de la carga residual que tenga el vehículo en ese momento, es igual al costo fijo de ir del cliente  $n$  al depósito.

Analíticamente, el valor de la función objetivo  $f_0(Q)$  se determina utilizando un algoritmo de programación dinámica con recursión en reversa, tal y como se ilustra en el Algoritmo 2.1.

#### Algoritmo 2.1 Cálculo de la función de costo para el VRPSD

Para  $(q = Q, Q-1, \dots, 0)$  hacer

$$f_n(q) = c_{n,0}$$

Para  $(j = n-1, n-2, \dots, 1)$  hacer

Calcular  $f_j^r(q)$  usando  $f_{j+1}(\cdot)$

Para  $(q = Q, Q-1, \dots, 0)$  hacer

Calcular  $f_j^p(q)$

Comparar  $f_j^r(q)$  y  $f_j^p(q)$  para encontrar el umbral  $h_j$

Calcular  $f_j(q)$  usando  $f_{j+1}(\cdot)$

Fin del for

Fin del for

Calcular  $f_0(Q)$

Fin del for

El VRPSD es aplicado en casos prácticos en los cuales los requerimientos de los clientes sufren cambios en el momento en que son visitados o donde es prácticamente imposible predecir la demanda de los clientes; tal es el caso de la recolección de residuos donde no se puede determinar con exactitud la cantidad de desechos que se deben recoger en cada lugar. La anterior situación puede ser vista también desde la óptica de la logística inversa: las empresas en su interés de personalizar el servicio que ofrecen a sus clientes, necesitan planear, ejecutar y controlar de una manera eficiente y rentable el flujo de los subproductos y residuos generados por la mercancía que éstas venden con el propósito de reutilizarlos, reciclarlos o eliminarlos por completo; en este caso, tampoco se conoce con certeza la cantidad de residuos o de subproductos que deben ser recogidos a los puntos donde se encuentran los consumidores.

### 3. ENFOQUE PARA LA SOLUCIÓN DEL VRPSD

El VRPSD es un problema de optimización combinatoria perteneciente a la clase *NP-hard*. Encontrar soluciones óptimas a este problema implica contemplar, para cada una de las posibles permutaciones de clientes, todas las realizaciones de las demandas que éstos presenten. Es por ello que tal y como lo afirma Bianchi et al<sup>43</sup>, la búsqueda de soluciones óptimas incluso para instancias pequeñas del VRPSD, se convierte en una opción prácticamente inviable si ésta se realiza de manera exacta.

El presente trabajo utiliza el enfoque de Optimización por Simulación (*Simulation-Optimization*) el cual según se presenta en Ólafsson y Kim<sup>44</sup> y April et al<sup>45</sup>, busca determinar la mejor configuración de parámetros de entrada que alimenten al modelo de simulación produciendo resultados cercanos al óptimo. El enfoque está conformado por dos módulos: un Módulo de Optimización en donde se lleva a cabo la exploración en el espacio de búsqueda a través de un método de optimización y un Módulo de Simulación el cual se encarga de medir la optimalidad utilizando un modelo de simulación (Figura 3).

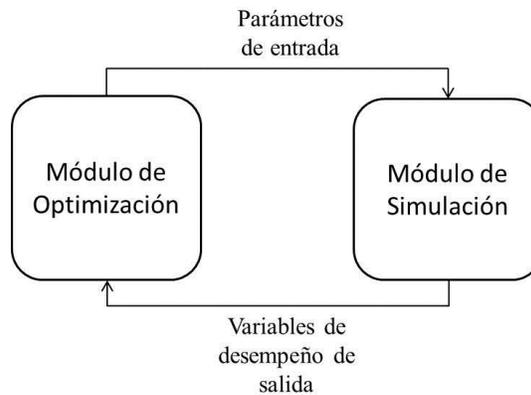
---

<sup>43</sup> Ibid.. p 9.

<sup>44</sup> ÓLAFSSON, Sigurdur y KIM, Jumi. Simulation Optimization. En: Proceedings of the 2002 Winter Simulation Conference E. Yiicesan, C.-H. Chen, J. L. Snowdon, and J. M. Chames, eds. S.I.: IEEE. 2002. pp. 79–84.

<sup>45</sup> APRIL, Jay., et al. Practical introduction to simulation optimization. En: Proceedings of the 2003 Winter Simulation Conference. S.I.: s.n. 2003.

Figura 3. Esquema general del enfoque de optimización por simulación



Fuente: Autor

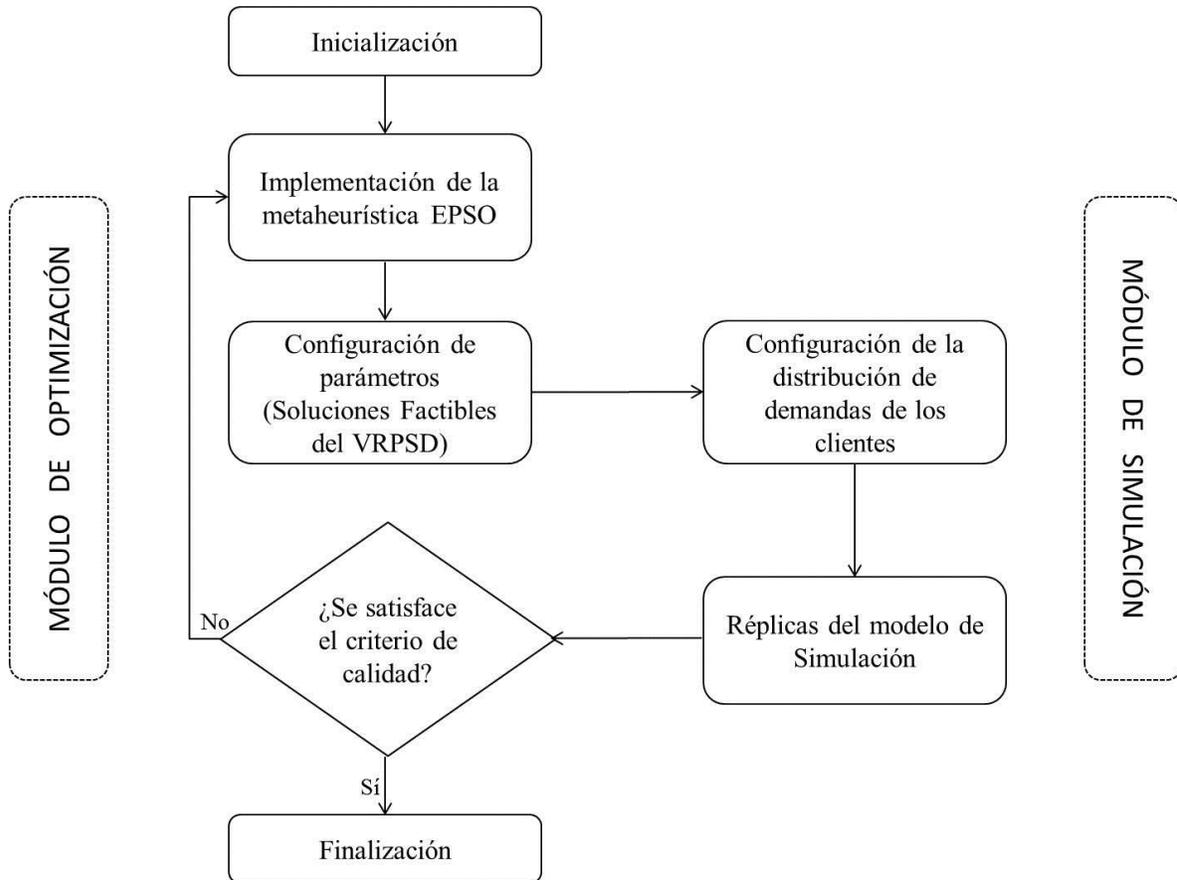
En la literatura se destacan los trabajos de Rei, Gendrau y Soriano<sup>46</sup>; y Tripathi, Kuriger y Wan<sup>47</sup>, los cuales utilizan Optimización por Simulación para el VRPSD con uno y múltiples vehículos. La Figura 4 presenta el framework SIM-EPSo propuesto en este trabajo para resolver el VRPSD, en donde los módulos de optimización y simulación están dados por la metaheurística híbrida Optimización de Enjambre de Partículas Evolutivo (EPSo) y Simulación de Monte Carlo, respectivamente. Cada una de las etapas correspondientes al módulo de optimización se discuten detalladamente en los siguientes numerales.

---

<sup>46</sup> Ibid.. p 15.

<sup>47</sup> Ibid.. p 20.

Figura 4. Framework SIM-EPHO para el VRPSD



Fuente: Autor

### 3.1 OPTIMIZACIÓN DE ENJAMBRE DE PARTÍCULAS (PSO)

El presente trabajo utiliza el PSO como método de solución para el VRPSD incorporando los operadores evolutivos de cruce y mutación con el fin de proporcionar mejores resultados. Inicialmente se presentará de manera formal el PSO para el VRPSD y después se incorporarán los operadores genéticos al algoritmo inicial.

El PSO es un método de optimización estocástico propuesto por Kennedy y Eberhart<sup>48</sup> el cual se basa en imitar a nivel computacional el comportamiento social de las aves dentro de una bandada a partir de la interacción entre sus integrantes y de éstos con el entorno en el que se desenvuelven. El proceder colectivo que emerge de su comportamiento es determinante para descubrir regiones óptimas en el espacio de búsqueda multidimensional.

La motivación para la selección del PSO en la presente investigación se fundamenta en los resultados obtenidos en diferentes trabajos desarrollados para resolver problemas de optimización combinatoria, entre los que se destacan el de Liu et al<sup>49</sup> para el Bin Packing Problem, Marinakis y Marinaki<sup>50</sup>, Ai y Kachitvichyanukul a.<sup>51</sup>, Geetha, Ganesan y Vanathi<sup>52</sup> y Ai y Kachitvichyanukul b.<sup>53</sup> para el VRP, entre otros; los cuales destacan la facilidad en la implementación y rápida convergencia. Por otro lado, la aplicación de esta metaheurística al VRPSD no ha sido suficientemente explorada comparada con otros métodos de solución,

---

<sup>48</sup> KENNEDY, James y EBERHART, Russell. Particle Swarm Optimization. En: Natural Networks, 1995. Proceedings., IEEE International Conference. S.l.: s.n. 1995. pp. 1942–1948.

<sup>49</sup> LIU, D S., et al. On solving multiobjective bin packing problems using evolutionary particle swarm optimization. En: European Journal of Operational Research. 2008. Vol. 190, pp. 357–382. DOI 10.1016/j.ejor.2007.06.032.

<sup>50</sup> MARINAKIS, Yannis y MARINAKI, Magdalene. A hybrid genetic – Particle Swarm Optimization Algorithm for the vehicle routing problem. En: Expert Systems With Applications [en línea]. 2010. Vol. 37, no. 2, pp. 1446–1455. DOI 10.1016/j.eswa.2009.06.085. Disponible en: <http://dx.doi.org/10.1016/j.eswa.2009.06.085>.

<sup>51</sup> AI, The Jin y KACHITVICHYANUKUL, Voratas. Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem. En: Computers & Industrial Engineering [en línea]. 2009. Vol. 56, no. 1, pp. 380–387. DOI 10.1016/j.cie.2008.06.012. Disponible en: <http://dx.doi.org/10.1016/j.cie.2008.06.012>.

<sup>52</sup> GEETHA, S, GANESAN, Poonthalir y VANATHI, P T. A Hybrid Particle Swarm Optimization with Genetic Operators for Vehicle Routing Problem. En: Journal of Advances in Information Technology. 2010. Vol. 1, no. 4, pp. 181–188. DOI 10.4304/jait.1.4.181-188.

<sup>53</sup> AI, The Jin y KACHITVICHYANUKUL, Voratas. Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem. En: Computers & Industrial Engineering [en línea]. 2009. Vol. 56, no. 1, pp. 380–387. DOI 10.1016/j.cie.2008.06.012. Disponible en: <http://dx.doi.org/10.1016/j.cie.2008.06.012>.

por lo que se convierte en un campo interesante a la hora de obtener nuevos resultados. Entre los aportes al VRPSD usando PSO, se destacan los trabajos de Shanmugam, Ganesan y Vanathi<sup>54</sup> y Farhang, Ruiz y Jafar<sup>55</sup>.

En el PSO cada partícula representa una solución potencial en el espacio de búsqueda, en donde cada una de ellas tiene asociado un valor fitness y una velocidad, las cuales direccionan a las partículas hacia la solución. Un enjambre de partículas se representa de la siguiente manera:

Sea:

$N$  : Dimensión de la partícula.

$m$  : Número de partículas.

$X_i = (x_{i1}, x_{i2}, \dots, x_{i1}, \dots, x_{iN})$  : Posición de la partícula  $i$ -ésima.

$V_i = (v_{i1}, v_{i2}, \dots, v_{i1}, \dots, v_{iN})$  : Velocidad de la partícula  $i$ -ésima.

$P_i = (p_{i1}, p_{i2}, \dots, p_{i1}, \dots, p_{iN})$  : Mejor solución de la partícula  $i$ -ésima ( $p_{best}$ ).

$P_{gi} = (p_{gi1}, p_{gi2}, \dots, p_{gi1}, \dots, p_{giN})$  : Mejor solución global del enjambre ( $g_{best}$ ).

Durante el proceso iterativo, las partículas actualizan sus posiciones y velocidades de la siguiente forma:

Sea  $x_{il}(t)$  la posición de la partícula  $i$  en la dimensión  $l$ -ésima en el momento  $t$ .

La posición de la partícula se actualiza mediante la fórmula:

$$x_{il}(t+1) = x_{il}(t) + v_{il}(t+1), \quad (3.1)$$

---

<sup>54</sup> SHANMUGAM, Geetha, GANESAN, Poonthalir y VANATHI, P T. Meta Heuristic Algorithms for Vehicle Routing Problem with Stochastic Demands. En: Journal of Computer Science. 2011. Vol. 7, no. 4, pp. 533–542.

<sup>55</sup> FARHANG, Babak, RUIZ, Rubén y JAFAR, Seyed. Vehicle routing problem with uncertain demands : An advanced particle swarm algorithm. En: Computers & Industrial Engineering [en línea]. 2011. pp. 1–12. DOI 10.1016/j.cie.2011.10.001. Disponible en: <http://dx.doi.org/10.1016/j.cie.2011.10.001>.

donde  $x_{il}(t)$  es la posición de la partícula  $i$  en la dimensión  $l$ -ésima en el momento  $t$  y  $v_{il}(t)$  es la velocidad de la partícula  $i$  en la dimensión  $l$ -ésima en el momento  $t$ .

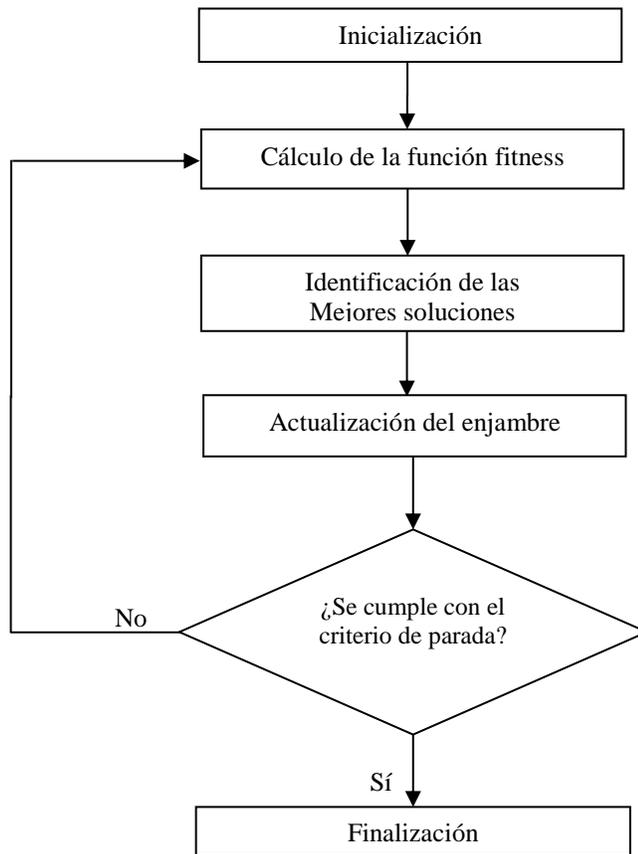
A su vez la, velocidad de cada partícula se actualiza mediante:

$$v_{il}(t+1) = w(t)v_{il}(t) + (c_p \text{rand}()) [p_{il}(t) - x_{il}(t)] + (c_g \text{rand}()) [p_{gl}(t) - x_{il}(t)], \quad (3.2)$$

aquí,  $c_p$  y  $c_g$  son los factores de aprendizaje cognitivo y social,  $\text{rand}()$  es un número aleatorio con distribución uniforme en el intervalo  $[0,1]$ ,  $w(t)$  es el factor de inercia en el momento  $t$ ,  $p_{il}(t)$  es la mejor posición personal de la partícula  $i$  y finalmente  $p_{gl}(t)$ , que representa la mejor solución global del enjambre.

La velocidad de cada partícula a su vez conduce el proceso de optimización y tiene bajo consideración el conocimiento propio de ésta y el conocimiento social, resultado del intercambio de información de la vecindad de la partícula. El conocimiento propio de la partícula también es conocido como el *componente cognitivo*, el cual es proporcional a la distancia de la partícula desde su mejor posición encontrada dado el primer movimiento realizado y se denota como  $p_{best}$ . El intercambio de información social se conoce como el *componente social* de la ecuación de velocidad, denotado como  $g_{best}$ . En la Figura 5 se presenta el esquema general del PSO en donde se definen sus principales etapas.

Figura 5. Esquema general del PSO



Fuente: Autor

En el esquema anterior se reflejan claramente las seis etapas del PSO que a continuación se describen de manera detallada:

En la primera etapa se inicializa el enjambre de manera aleatoria. En la segunda etapa se calcula el valor de la función de aptitud (fitness) para cada partícula. En la tercera etapa se identifican, la mejor solución personal (pbest) y la mejor solución global (gbest). En la cuarta etapa se actualiza la velocidad y la posición de cada partícula, utilizando las ecuaciones (3.1) y (3.2).

Una vez se haya actualizado el enjambre, se verifica el criterio de calidad, en donde se evalúa la finalización del algoritmo. El criterio puede estar dado en un porcentaje de convergencia o en un determinado número de iteraciones.

### **3.2 OPTIMIZACIÓN DE ENJAMBRE DE PARTÍCULAS CON OPERADORES EVOLUTIVOS PARA EL VRPSD**

El PSO fue concebido para resolver problemas de optimización con variables continuas y el VRPSD es un problema de optimización combinatoria, por consiguiente, es necesario explicar cómo se representa la solución en este caso y de qué manera se convierten las permutaciones de clientes en valores posición, para de esta forma, crear el enjambre de partículas.

#### **Representación de la Solución**

La solución del VRPSD teniendo en cuenta el PSO, se codifica como la permutación entera  $Y_i = (y_{i1}, y_{i2}, \dots, y_{il}, \dots, y_{iN})$ , donde  $i$  representa a la partícula  $i$ -ésima y  $l$  el cliente  $l$ -ésimo.

#### **Conversión de la Permutación de Clientes a Valores Posición**

Cada solución inicial generada es una permutación de todos los clientes. La conversión se efectúa usando la siguiente expresión:

$$x_{il} = x_{\min} + \frac{(x_{\max} - x_{\min})}{n * [y_{il} - 1 + rand()]} \quad (3.3)$$

Donde:

$y_{il}$ :  $l$ -ésima dimensión de la solución  $i$ -ésima (cliente).

$x_{il}$ :  $l$ -ésima dimensión de la partícula  $i$ -ésima en el espacio del enjambre (valor posición de cada cliente en la partícula).

rand(): número aleatorio distribuido uniformemente en  $[0,1]$ .

$n$ : número de clientes.

$x_{\min}$ : valor de frontera mínimo del valor posición de la partícula en el espacio del enjambre.

$x_{\max}$ : valor de frontera máximo del valor posición de la partícula en el espacio del enjambre.

El siguiente ejemplo muestra cómo se realiza la transformación de la permutación de clientes (solución codificada) a sus valores posición equivalente en el espacio solución del PSO utilizando la ecuación (3.3), en donde los valores frontera mínimo y máximo ( $x_{\min}$  y  $x_{\max}$ ) son seleccionados arbitrariamente.

$Y_i = (3, 4, 2, 5, 1, 6)$  Solución codificada

$n=6$  Número de clientes

$x_{\min}=1$

$x_{\max}=10$

El vector posición equivalente a la solución codificada se presenta en la Tabla 1:

Tabla 1. Ejemplo para la conversión de las soluciones en valores posición continuos

Dimensión ( $l$ )	1	2	3	4	5	6
Solución codificada ( $Y_i$ )	3	4	2	5	1	6
Vector posición ( $X_i$ )	2.9177	4.8203	6.9363	8.4473	1.2364	9.9559

De igual forma se debe realizar el proceso inverso denominado decodificación de la solución PSO. La solución encontrada en el espacio solución PSO deberá decodificarse como una solución del problema original, la cual esta dada por medio de una permutación de clientes. Es por ello que los valores continuos de posición del vector  $X_i$  se convierten nuevamente a la solución codificada  $Y_i$  utilizando el Rank Order Value (ROV). El ROV es una técnica propuesta por Bo, Wang y Jing<sup>56</sup> la cual convierte los valores continuos de las partículas en permutación de clientes.

Con el ROV, la conversión se realiza seleccionando el menor valor posición de la partícula y asignándole a éste el valor de 1. Luego, se selecciona el segundo menor valor posición y se le asigna el valor de 2 y así sucesivamente con todos los elementos del vector posición. El vector resultante de este proceso, esto es, el vector que contiene los valores enteros que se usaron para la asignación, corresponde a la permutación de clientes resultante del proceso iterativo del PSO.

Siguiendo con la instancia de seis clientes, y suponiendo que el vector posición que se tiene es resultado de haber implementado el PSO, la conversión a permutación de clientes sería la presentada en la Tabla 2.

Tabla 2. Ejemplo para la aplicación del ROV

Dimensión ( $l$ )	1	2	3	4	5	6
Vector posición ( $X_i$ )	4,728	7,213	9,874	3,936	1,633	6,700
Solución codificada ( $Y_i$ )	3	5	6	2	1	4

---

<sup>56</sup> LIU, Bo, WANG, Ling y JIN, Yi-hui. An effective hybrid PSO-based algorithm for flow shop scheduling with limited buffers. En: Computers & Operations Research. 2008. Vol. 35, pp. 2791–2806. DOI 10.1016/j.cor.2006.12.013.

Teniendo en cuenta lo anterior, el Algoritmo 3.1 presenta de manera formal el algoritmo para resolver el VRPSD utilizando PSO con operadores evolutivos (EPSO).

#### Algoritmo 3.1. EPSO aplicado al VRPSD

1. Inicializar las partículas usando la heurística del vecino más cercano
2. Repetir mientras se cumpla el criterio de parada
3. Para cada partícula
  - 3.1 Calcular el valor del fitness usando simulación de Monte Carlo
  - 3.2 Si el valor del fitness es mejor que el valor fitness anterior  
Establecer el actual valor como el nuevo pbest
  - 3.3 Fin
4. Escoger la partícula con el mejor valor de fitness de todo el enjambre como gbest
5. Para cada partícula
  - 5.1 Convertir la solución en su respectivo vector posición
  - 5.2 Calcular la velocidad de las partículas
  - 5.3 Actualizar la posición de las partículas
  - 5.4 Convertir las partículas a su respectiva permutación codificada
  - 5.5 Aplicar el operador de cruce
  - 5.6 Aplicar el operador de mutación
  - 5.7 Fin
6. Regresar al paso 3
7. Fin
8. Establecer el gbest como la mejor solución global

En el paso 1 se inicializan las soluciones mediante la generación de un conjunto de soluciones (soluciones codificadas), en donde cada una de ellas va a estar representada como una permutación de todos los clientes de la instancia. Una manera intuitiva de obtener dicho conjunto es generando la permutación aleatoriamente, de manera análoga a la generación de las partículas en el PSO

básico Kennedy y Eberhart<sup>57</sup>. Sin embargo, con el fin de garantizar una población inicial de mejor calidad, ésta es generada utilizando el algoritmo del vecino más cercano.

En el paso 3 se calcula el valor de la función de aptitud (fitness) para cada solución. En este caso, se obtiene el valor esperado de la función objetivo para cada solución del VRPSD.

En los pasos 3.2 - 4, se llevan a cabo las siguientes actividades:

1. Actualización del pbest: el pbest es un vector de tamaño  $m$ , donde cada elemento contiene el mejor valor de fitness de la partícula  $m$ -ésima en todo el proceso iterativo. En la primera iteración el pbest es igual a los valores fitness hallados en ese momento y a partir de la segunda iteración, este vector se actualiza si el fitness de al menos una partícula en ese instante, es mejor al valor anterior.
2. Actualización del gbest: teniendo en cuenta el pbest, se escoge el mejor valor de fitness de todas las partículas con el fin de identificar la mejor partícula.

En el paso 5 se realiza la actualización del enjambre de la siguiente manera: como ya se mencionó, en el PSO las partículas se mueven en un espacio continuo, por lo que es necesario convertir las soluciones codificadas en sus valores posición equivalente usando la ecuación (3.3). Con el fin de realizar la actualización de la velocidad y la posición de cada partícula, se utilizan las ecuaciones (3.1) y (3.2). Una vez se hayan actualizado estos valores, las partículas se convierten de nuevo a la permutación de clientes usando el ROV y de esta forma, llevar a cabo el proceso de aplicación de los operadores evolutivos de cruce y mutación.

El proceso se repite hasta que se haya satisfecho el criterio de calidad, estableciendo al *gbest* como la mejor solución global del problema.

---

<sup>57</sup> Ibid.. p 30.

## Parámetros básicos del PSO

El PSO está influenciado por los parámetros de control: dimensión del problema, número de partículas, coeficientes de aceleración, inercia, número de iteraciones y números aleatorios que escalan la contribución de los componentes cognitivo y social. A continuación se discuten cada uno de los parámetros tal y como los define Engelbrecht<sup>58</sup>.

- **Dimensión del problema ( $N$ ):** la dimensión del problema representa el máximo orden de la dimensión de las partículas en el espacio multidimensional. Para el VRPSD,  $N$  está dada por el total de clientes, en donde a cada cliente se le asocia con una dimensión.
- **Número de partículas ( $m$ ):** el número de partículas o tamaño del enjambre es un parámetro que es definido directamente por el diseñador y su valor varía según el tipo de problema. Sin embargo, en términos generales, entre más partículas se tengan, mayor es la diversidad en las soluciones ya que permite un mayor cubrimiento del espacio de búsqueda. No obstante, un número elevado de partículas incrementa la complejidad computacional del problema. Por otra parte, si se toma un número reducido de partículas, la diversidad en las soluciones no se garantiza. De esta manera, la selección del número de partículas, es un proceso empírico en el cual, a través de la experimentación, se define el tamaño “adecuado” del enjambre.
- **Coefficientes de aceleración ( $c_p, c_g$ ):** los coeficientes de aceleración controlan la influencia estocástica de los componentes cognitivo y social dentro de la velocidad de la partícula. El parámetro  $c_p$ , expresa qué tanta confianza tiene

---

<sup>58</sup> ENGELBRECHT, Andries. Computational Intelligence.pdf. En: Computational Intelligence, An Introduction. Second Edi. S.l.: John Wiley & Sons, Ltd. pp. 285–357. 2007.

una partícula en sí misma, mientras que el parámetro  $c_g$  expresa la confianza que tiene una partícula en sus vecinos.

Si bien la elección de los valores de estos parámetros obedece a un proceso experimental, existen en la literatura análisis de ciertas situaciones que se pueden presentar para las configuraciones de  $c_p$  y  $c_g$ .

- a)  $c_p = c_g = 0$ . Las partículas se mantienen volando a su velocidad actual hasta que alcancen un límite en el espacio de búsqueda.
  - b)  $c_p > 0$ ,  $c_g = 0$ . Todas las partículas construyen su camino de manera independiente. De esta manera, cada partícula encuentra su mejor posición en su vecindad reemplazando la mejor posición actual, siempre y cuando la nueva posición sea mejor a la anterior. Dicho en otras palabras, las partículas realizan búsqueda local.
  - c)  $c_g > 0$ ,  $c_p = 0$ . Todo el enjambre es atraído por la mejor posición global.
  - d)  $c_p \approx c_g$ . El enjambre logra ser más eficiente cuando existe cooperación entre sus integrantes, por lo que debe existir un balance entre los coeficientes, haciendo que las partículas sean atraídas hacia el promedio.
  - e)  $c_p \gg c_g$ . Cada partícula es más atraída a su mejor posición personal, haciendo que las soluciones se queden atrapadas en óptimos locales.
  - f)  $c_g \gg c_p$ . Las partículas se ven más atraídas a la mejor posición global, causando que el enjambre converja prematuramente.
- **Inercia ( $w$ ):** la inercia es un mecanismo que controla la exploración y la explotación de las soluciones en el espacio de búsqueda por parte del enjambre. El valor de este parámetro cobra vital importancia debido a que asegura la convergencia y un balance en la exploración y la explotación. Cuando  $w \geq 1$ , las velocidades de las partículas aumentan y el enjambre diverge; adicionalmente, las partículas pueden fracasar en hacer cambios de dirección acertados con el fin de regresar hacia regiones promisorias. Cuando

$w < 1$ , las partículas reducen sus velocidades incrementando la exploración local.

## Operadores evolutivos

El PSO tiene varias limitaciones las cuales están asociadas principalmente a la configuración de los parámetros propios de la metaheurística como lo establecieron en su trabajo Shanmugam, Ganesan y Vanathi<sup>59</sup>. Tales parámetros varían de un problema a otro, por lo que encontrar los valores óptimos de los parámetros del PSO no es una tarea sencilla. Como se mencionó anteriormente, el enjambre puede converger prematuramente debido, entre otras razones, a un gran flujo de información entre las partículas asociados a altos valores coeficientes de aceleración; o bien, la convergencia a la mejor solución se podría ver demorada innecesariamente por bajas asignaciones en los coeficientes tanto de aceleración y de inercia y además, por el tamaño del espacio multidimensional en el cual se va a mover el enjambre, restringiendo de esta manera la diversidad en las soluciones.

Teniendo en cuenta lo anterior, en el presente trabajo se incorporan algoritmos evolutivos al PSO con el propósito de superar sus limitaciones. Mientras con el PSO se puede controlar en cierta medida la convergencia, los operadores evolutivos pueden garantizar una mayor diversidad de soluciones en el espacio de búsqueda, lo cual hace que el enjambre se desplace hacia otra parte del espacio de búsqueda, evitando de esta manera una convergencia prematura. Para este trabajo se utilizó secuencialmente un operador de cruce simple el cual es aplicado a todo el enjambre con el *gbest*, seguido por un operador de mutación basado en el intercambio de genes. En el Algoritmo 3.2 y en el Algoritmo 3.3 se ilustran la implementación los operadores evolutivos de cruce y mutación respectivamente.

---

<sup>59</sup> Ibid.. p 20.

### Algoritmo 3.2. Implementación del operador evolutivo de cruce

1. Para cada solución codificada  $Y_i$ 
  - 1.1 Seleccionar aleatoriamente un punto  $p$  entre 1 y N
  - 1.2 Intercambiar los valores de la solución codificada con el  $g_{best}$  codificado
  - 1.3 Fin

### Algoritmo 3.3. Implementación del operador evolutivo de mutación

1. Para cada solución cruzada
  - 1.1 Seleccionar dos puntos  $p_1$  y  $p_2$  e intercambiarlos
  - 1.2 Fin

## 4. RESULTADOS

En esta sección se describe el diseño experimental utilizado y los resultados obtenidos al implementar el framework SIM-EPHO. En primer lugar se presenta la configuración de los parámetros del EPHO, seguido por la descripción el establecimiento del *test bed* utilizado en este trabajo, pasando por la definición de los factores y sus respectivos niveles del diseño experimental; por último, se presentan los resultados obtenidos de los experimentos realizados.

### 4.1 CONFIGURACIÓN DE LOS PARÁMETROS DEL EPHO

El framework SIM-EPHO fue programado en Matlab 7.1 y ejecutado en un equipo con procesador Intel Core i3 y 4GB de RAM. Dentro de los parámetros del EPHO, se establecieron los siguientes valores: número de partículas:  $m=20$ ; coeficientes de aceleración:  $c_p=1$  y  $c_g=1,5$ ; inercia:  $w=0,7$ ; valores frontera para convertir las soluciones en partículas:  $x_{\min}=0,1$  y  $x_{\max}=0,7$  y el número de iteraciones del algoritmo es igual a 100.

### 4.2 TEST BED

En la literatura no es común realizar comparaciones con instancias benchmark para el VRPSD, Bianchi et al<sup>60</sup>. Por tal motivo, todas las instancias utilizadas en este trabajo fueron generadas de la siguiente manera: el depósito se encuentra ubicado en (0,0); para la generación de las demandas de los clientes, se consideraron 100 muestras, teniendo en cuenta que las demandas de los clientes

---

<sup>60</sup> Ibid.. p 9.

siguen una distribución de probabilidad uniforme discreta. Los demás elementos que constituyen el test bed se discuten en la siguiente sección.

### 4.3 DISEÑO EXPERIMENTAL

Con el fin de determinar el impacto de los parámetros de entrada del VRPSD al framework propuesto y analizar el impacto que éstos tienen en la función objetivo, se llevó a cabo un diseño experimental tipo Taguchi en donde se definieron seis factores, cada uno con dos niveles. Para ello, se realizó un diseño fraccionado en el cual se seleccionó un arreglo ortogonal  $L_8$ , es decir, un arreglo con  $2^{6-3} = 8$  experimentos, cada uno de ellos con cinco réplicas asociadas.

Tabla 3. Definición de los factores y niveles del diseño experimental

FACTOR	NIVEL 1	NIVEL 2
Número de clientes ( $n$ )	100	200
Ubicación de los clientes ( $P$ )	Uniforme discreta	Distribución normal con dos clusters
Demanda promedio ( $D_i$ )	[50,70]	[80,100]
Desviación ( $S_i$ )	[1,5]	[10,15]
Número promedio de clientes atendidos antes de la descarga preventiva ( $r$ )	4	8
Umbral ( $h_i$ )	Demanda promedio	Primer cuartil de la demanda

En la Tabla 3 se definen los seis factores seleccionados para el diseño experimental. En primer lugar se encuentra el número de clientes el cual toma el valor de 100 en el Nivel 1 y de 200 en el Nivel 2. Por otro lado, se considera que la ubicación de los clientes está dada por una distribución uniforme en el intervalo  $[0,100]$  para el Nivel 1 y para el Nivel 2, se asume que los clientes están ubicados aleatoriamente con distribución normal alrededor de dos centros (por lo que se tienen dos clusters); tanto la media como la desviación estándar para cada uno de los centros se seleccionaron aleatoriamente con una distribución uniforme en el intervalo  $[0,100]$ .

La distribución de las demandas de los clientes se genera teniendo en cuenta dos parámetros: la demanda promedio ( $D_i$ ) y la desviación estándar ( $S_i$ ). Para los dos niveles, se asume que las demandas de los clientes siguen una distribución de probabilidad uniforme discreta en el intervalo  $[D_i - S_i, D_i + S_i]$ .

La capacidad del vehículo se calcula teniendo en cuenta la formulación propuesta por Gendreau, Laporte y Séguin<sup>61</sup> (Michel Gendreau, Laporte, Séguin, Seguin 1995) en el cual consideran que un factor que influye en la complejidad del problema es la cantidad de clientes atendidos por el vehículo en un solo recorrido. La capacidad del vehículo está dada por la ecuación (4.1).

$$Q = \frac{\sum (D_i) * r}{n} \quad (4.1)$$

En donde  $r$  es el número promedio de clientes atendidos por el vehículo antes que éste realice la descarga preventiva.  $r$  toma el valor de 4 para el Nivel 1 y de 8 para el Nivel 2. El último factor definido es el umbral o política de descargue, el cual representa el nivel de carga residual que tiene el vehículo para realizar el descargue preventivo; en el Nivel 1 tal valor está dado por la media de la demanda

---

<sup>61</sup> GENDREAU, Michel., et al. An Exact Algorithm for the Vehicle Routing Problem with Stochastic Demands and Customers. En: Transportation Science. 1995. Vol. 29, no. 2, pp. 143–155.

del cliente  $j$  y en el Nivel 2 es igual a el primer cuartil dentro de la distribución de demanda del cliente  $j$ .

En la Tabla 4 se encuentran los resultados del diseño experimental realizado. Cada experimento fue replicado cinco veces y las conclusiones se hacen con respecto al valor promedio de la función objetivo.

Tabla 4. Diseño experimental y resultados de SIM-EPSSO

<i>Exp</i>	<i>n</i>	<i>P</i>	$D_i$	$S_i$	<i>r</i>	$h_i$	VALORES DE LA FUNCIÓN OBJETIVO						
							Réplica 1	Réplica 2	Réplica 3	Réplica 4	Réplica 5	Media	Desviación estándar
1	1	1	1	1	1	1	5103,92	5098,42	5121,99	5094,48	5098,49	5103,46	9,74
2	1	1	2	2	2	2	2882,82	2889,65	2854,73	2880,20	2880,24	2877,53	11,91
3	1	2	1	2	2	1	2257,31	2245,06	2244,49	2246,38	2245,87	2247,82	4,79
4	1	2	2	1	1	2	4058,76	4096,88	4067,02	4049,21	4102,71	4074,92	21,16
5	2	1	1	2	1	2	9276,72	9285,62	9268,53	9299,94	9243,31	9274,82	18,89
6	2	1	2	1	2	1	5085,00	5087,41	5074,53	5107,58	5110,89	5093,08	13,92
7	2	2	1	1	2	2	4121,47	4143,09	4122,03	4126,20	4130,46	4128,65	7,91
8	2	2	2	2	1	1	8016,96	7990,46	8013,87	8007,66	7997,86	8005,36	9,91

## 5. DISCUSIÓN DE LOS RESULTADOS

### 5.1 ANÁLISIS DE LOS RESULTADOS DEL DISEÑO EXPERIMENTAL

A continuación se presenta un análisis de la varianza para la métrica (valor de la función objetivo) en que se evaluó el framework SIM-EPSO. Para estudiar el efecto de cierto conjunto de parámetros sobre el valor de la función objetivo, se propuso una serie de experimentos cuidadosamente diseñados para un test bed. Se considera que los factores que se seleccionaron podría tener un gran impacto en el valor de la función objetivo. Teniendo en cuenta lo anterior, se llevó a cabo un diseño de Taguchi ortogonal L8 que tiene ocho corridas experimentales, y con él se pueden estudiar desde dos hasta siete en dos niveles cada uno. Con el presente diseño se estudiaron 6 factores, cada uno en dos niveles con el propósito de estudiar los efectos principales.

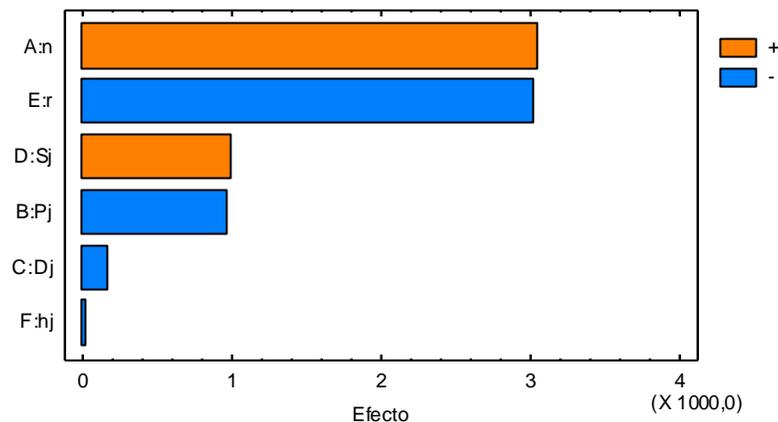
En la Tabla 5 se muestran las estimaciones para cada uno de los efectos principales para la variable respuesta valor de la función objetivo.

Tabla 5. Efectos estimados para la función objetivo

<b>Efecto</b>	<b>Estimado</b>	<b>Error Estd.</b>
Promedio	5100,71	71,9555
A: $n$	3049,55	143,911
B: $P$	-973,036	143,911
C: $D_i$	-175,967	143,911
D: $S_i$	1001,36	143,911
E: $r$	-3027,87	143,911
F: $h_i$	-23,452	143,911

Los efectos se grafican en un diagrama de Pareto para visualizar cuáles tienen un mayor impacto sobre la variable respuesta. Los efectos se muestran por orden de importancia, aquí se observa que los factores  $n$  y  $S_i$  tienen un efecto positivo en la función objetivo; esto quiere decir que al pasar del nivel bajo (Nivel 1) al nivel alto (Nivel 2), el valor de la función objetivo aumenta. Por su parte, el efecto de los factores  $r$  y  $P$  es negativo sobre la función objetivo al pasar del nivel bajo al nivel alto; lo anterior significa que cuando la capacidad del vehículo está en el nivel alto, la cobertura de clientes atendidos antes de la descarga preventiva aumenta, disminuyendo de esta manera el número de viajes al depósito por concepto de descargas preventivos o por fallas en la ruta. Asimismo, cuando los clientes están ubicados alrededor de dos centros, el valor de la función objetivo es menor a si éstos estuvieran ubicados aleatoriamente con distribución uniforme en un área dada.

Figura 6. Diagrama Pareto para los efectos sobre la función objetivo



Fuente: Autor

El análisis de varianza de la Tabla 6 se usa para confirmar la magnitud de estos efectos. En la tabla se observa que existen cuatro efectos, dos de ellos tienen valores P menores que 0,05; mientras que los dos restantes son significativos a

nivel menor de 0,1. Los efectos principales que resultan significativos son los asociados a los factores  $n$ ,  $r$ ,  $P$  y  $S_i$ .

Tabla 6. Análisis de Varianza para la función objetivo

Fuente	Suma de Cuadrados	GI	Cuadrado Medio	Razón-F	Valor-P
A: $n$	1,85995E7	1	1,85995E7	449,04	0,0300
B: $P$	1,8936E6	1	1,8936E6	45,72	0,0935
C: $D_i$	61928,8	1	61928,8	1,50	0,4364
D: $S_i$	2,00543E6	1	2,00543E6	48,42	0,0909
E: $r$	1,8336E7	1	1,8336E7	442,68	0,0302
F: $h_i$	1099,99	1	1099,99	0,03	0,8972
Error total	41420,8	1	41420,8		
Total (corr.)	4,0939E7	7			

R-cuadrada = 99,8988 %

## 5.2 COMPARACIÓN DEL RENDIMIENTO DEL ALGORITMO

Como se mencionó en la Sección 4.2, en la literatura no se realizan comparaciones con instancias benchmark para el VRPSD. Por tal motivo, en el presente trabajo se llevó a cabo un análisis adicional de la calidad de los valores obtenidos en los experimentos comparándolos con la versión básica del PSO. Para ello se implementó el PSO para las ocho instancias evaluadas en el EPSO. La capacidad de búsqueda y el desempeño de los dos métodos se presentan en la Tabla 7.

Como puede apreciarse, las soluciones obtenidas con el EPSO son superiores respecto al PSO para las ocho instancias comparadas. El éxito del EPSO sobre el PSO radica en la capacidad que tiene el primero en balancear tanto la explotación como la exploración en el espacio de búsqueda ya que hace que el enjambre

“evolucione” a través de los operadores de cruce y mutación empleados, garantizando una mayor exploración en el espacio de búsqueda, aumentando de esta manera la posibilidad de encontrar mejores soluciones.

Tabla 7. Resultados obtenidos con EPSO y PSO

<b>INSTANCIA</b>	<b>EPSO</b>	<b>PSO</b>
1	5103,46	5123,95
2	2877,53	2894,25
3	2257,82	2282,30
4	4074,92	4136,47
5	9274,82	9497,70
6	5093,08	5125,84
7	4128,65	4152,73
8	8005,36	8015,54

## 6. CONCLUSIONES

En el presente trabajo se propuso el framework SIM-EPSO basado en el enfoque de optimización por simulación para el VRPSD con descargas preventivas para el caso de un solo vehículo, utilizando el algoritmo híbrido EPSO y simulación de Monte Carlo para calcular el costo esperado de la función objetivo.

Con el fin de establecer el impacto de algunos parámetros propios del VRPSD sobre la función objetivo, se llevó a cabo un diseño experimental sobre un test bed cuidadosamente generado. Para ello, se definieron seis factores, los cuales variaban en dos niveles. Los resultados de los experimentos arrojaron que factores como número de clientes y número promedio de clientes atendidos antes de la descarga preventiva, tienen una participación importante en el valor de la función objetivo.

Se comparó el desempeño del EPSO con el PSO en donde se observó la superioridad del framework propuesto con respecto al segundo método de solución; revelando de esta manera, la capacidad que tiene el EPSO de realizar mejores exploraciones en la región factible y diversificaciones en las soluciones gracias a los componentes de inteligencia de enjambre, propios del PSO y a los operadores evolutivos incorporados.

## **7. RECOMENDACIONES**

Como recomendaciones para futuras investigaciones, este trabajo puede ser adaptado fácilmente hacia la implementación de la política a priori. Por otro lado, se puede extender para el caso de múltiples vehículos y/o incorporando otros elementos aleatorios al modelo tales como los clientes o los tiempos de viaje de los vehículos. Respecto al método EPSO, sería interesante ahondar en nuevos métodos de decodificación de la solución para el caso de múltiples vehículos y hacer un análisis más detallado para encontrar mejores soluciones iniciales.

## REFERENCIAS BIBLIOGRÁFICAS

AI, The Jin y KACHITVICHYANUKUL, Voratas. A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. En: Computers & Operations Research. 2009. Vol. 36.

AI, The Jin y KACHITVICHYANUKUL, Voratas. Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem. En: Computers & Industrial Engineering [en línea]. 2009. Vol. 56, no. 1, pp. 380–387. DOI 10.1016/j.cie.2008.06.012. Disponible en: <http://dx.doi.org/10.1016/j.cie.2008.06.012>.

AK, A. y ERERA, L,. A Paired-Vehicle Recourse Strategy for the Vehicle-Routing Problem with Stochastic Demands. En: Transportation Science [en línea]. Mayo 2007. Vol. 41, no. 2, pp. 222–237. DOI 10.1287/trsc.1060.0180. Disponible en: <http://transci.journal.informs.org/cgi/doi/10.1287/trsc.1060.0180>.

ANDREATTA, G. y LULLI, G. A multi-period TSP with stochastic regular and urgent demands. En: European Journal of Operational Research [en línea]. Febrero 2008. Vol. 185, no. 1, pp. 122–132. [consultado 11 Abril 2011]. DOI 10.1016/j.ejor.2006.12.040. Disponible en: <http://linkinghub.elsevier.com/retrieve/pii/S0377221707000707>.

APRIL, Jay., et al. Practical introduction to simulation optimization. En: Proceedings of the 2003 Winter Simulation Conference. S.I.: s.n. 2003.

BERTSIMAS, D. J. A Vehicle Routing Problem with Stochastic Demand. En: Operations Research [en línea]. Mayo 1992. Vol. 40, no. 3, pp. 574–585. DOI 10.1287/opre.40.3.574. Disponible en: <http://or.journal.informs.org/cgi/doi/10.1287/opre.40.3.574>.

BERTSIMAS, D. J., JAILLET, P. y ODoni, a. R. A Priori Optimization. En: Operations Research [en línea]. Noviembre 1990. Vol. 38, no. 6, pp. 1019–1033. DOI 10.1287/opre.38.6.1019. Disponible en: <http://or.journal.informs.org/cgi/doi/10.1287/opre.38.6.1019>.

BIANCHI, Leonora. Ant Colony optimization and local search for the probabilistic traveling salesman problem: A case of study in stochastic combinatorial optimization. S.l.: PhD Thesis, Université Libre de Bruxelles. 2006.

BIANCHI, Leonora., et al. Metaheuristics for the Vehicle Routing Problem with Stochastic Demands. En: Journal of Mathematical Modelling and Algorithms [en línea]. Diciembre 2005. Vol. 5, no. 1, pp. 91–110. DOI 10.1007/s10852-005-9033-y. Disponible en: <http://www.springerlink.com/index/10.1007/s10852-005-9033-y>.

BIANCHI, Leonora., et al. Hybrid Metaheuristics for the Vehicle Routing Problem with Stochastic Demands. Reporte Técnico. S.l. 2005.

CHEPURI, Krishna y HOMEM-DE-MELLO, Tito. Solving the Vehicle Routing Problem with Stochastic Demands using the Cross-Entropy Method. En: Annals of Operations Research [en línea]. Febrero 2005. Vol. 134, no. 1, pp. 153–181. DOI 10.1007/s10479-005-5729-7. Disponible en: <http://www.springerlink.com/index/10.1007/s10479-005-5729-7>.

CHRISTIANSEN, C y LYSGAARD, J. A branch-and-price algorithm for the capacitated vehicle routing problem with stochastic demands. En: Operations Research Letters [en línea]. Noviembre 2007. Vol. 35, no. 6, pp. 773–781. [Consultado 11 Abril 2011]. DOI 10.1016/j.orl.2006.12.009. Disponible en: <http://linkinghub.elsevier.com/retrieve/pii/S0167637707000168>.

CORDEAU, Jean-françois., et al. Vehicle Routing. In: Handbook in OR & MS. 14. S.l.: s.n. pp. 367–428. 1995.

ENGELBRECHT, Andries. Computational Intelligence.pdf. En: Computational Intelligence, An Introduction. Second Edi. S.I.: John Wiley & Sons, Ltd. pp. 285–357. 2007.

FAN, Jianhua y WANG, Xiufeng. A Multiple Vehicles Routing Problem Algorithm with Stochastic Demand. En: 6th World Congress on Intelligent Control and Automation [en línea]. S.I.: Ieee. 2006. pp. 1688–1692. Disponible en: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1712640>.

FARHANG, Babak, RUIZ, Rubén y JAFAR, Seyed. Vehicle routing problem with uncertain demands: An advanced particle swarm algorithm. En: Computers & Industrial Engineering [en línea]. 2011. pp. 1–12. DOI 10.1016/j.cie.2011.10.001. Disponible en: <http://dx.doi.org/10.1016/j.cie.2011.10.001>.

FISHER, Marshall. Vehicle Routing. In: Handbook in OR & MS, vol 8. S.I.: s.n. 1995

GEETHA, S, GANESAN, Poonthalir y VANATHI, P T. A Hybrid Particle Swarm Optimization with Genetic Operators for Vehicle Routing Problem. En: Journal of Advances in Information Technology. 2010. Vol. 1, no. 4, pp. 181–188. DOI 10.4304/jait.1.4.181-188.

GENDREAU, Michel, LAPORTE, Gilbert y SÉGUIN, René, 1996. Stochastic vehicle routing. En: European Journal of Operational Research [en línea]. Enero 6 de 1996. Vol. 88, no. 1, pp. 3–12. DOI 10.1016/0377-2217(95)00050-X. Disponible en: <http://linkinghub.elsevier.com/retrieve/pii/037722179500050X>.

GENDREAU, Michel., et al. An Exact Algorithm for the Vehicle Routing Problem with Stochastic Demands and Customers. En: Transportation Science. 1995. Vol. 29, no. 2, pp. 143–155.

HJORRING, Curt y HOLT, John. New optimality cuts for a single-vehicle stochastic routing problem. En: Annals of Operations Research. 1999. Vol. 86, pp. 569–584.

ISMAIL, Z . y IRHAMAH. Solving the Vehicle Routing Problem with Stochastic Demands via Hybrid Genetic Algorithm-Tabu Search. En: Journal of Mathematics and Statistics. 2008. Vol. 4, no. 3, pp. 161–167.

JUAN, a., FAULIN, J., GRASMAN, S., RIERA, D., MARULL, J. y MENDEZ, C., 2010. Using safety stocks and simulation to solve the vehicle routing problem with stochastic demands. En: *Transportation Research Part C: Emerging Technologies* [en línea]. Octubre 2010. [Accessed 4 November 2010]. DOI 10.1016/j.trc.2010.09.007. Available from: <http://linkinghub.elsevier.com/retrieve/pii/S0968090X10001439>.

KENNEDY, James y EBERHART, Russell. Particle Swarm Optimization. En: Natural Networks, 1995. Proceedings., IEEE International Conference. S.l.: s.n. 1995. pp. 1942–1948.

KENYON, Astrid S. y MORTON, David P. Stochastic Vehicle Routing with Random Travel Times. En: Transportation Science [en línea]. Febrero 2003. Vol. 37, no. 1, pp. 69–82. DOI 10.1287/trsc.37.1.69.12820. Disponible en: <http://transci.journal.informs.org/cgi/doi/10.1287/trsc.37.1.69.12820>.

LAPORTE, G, LOUVEAUX, Francois y VAN HAMME, Luc. An integer L-Shaped algorithm for the capacitated vehicle routing problem with stochastic demands. En: Operations Research. 2001. Vol. 50, no. 3, pp. 415–423.

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

LIU, D S., et al. On solving multiobjective bin packing problems using evolutionary particle swarm optimization. En: European Journal of Operational Research. 2008. Vol. 190, pp. 357–382. DOI 10.1016/j.ejor.2007.06.032.

MANFRIN, Max. Ant Colony Optimization for the Vehicle Routing Problem. S.I.: Université Libre de Bruxelles. 2004.

MARINAKIS, Yannis y MARINAKI, Magdalene. A hybrid genetic – Particle Swarm Optimization Algorithm for the vehicle routing problem. En: Expert Systems With Applications [en línea]. 2010. Vol. 37, no. 2, pp. 1446–1455. DOI 10.1016/j.eswa.2009.06.085. Disponible en: <http://dx.doi.org/10.1016/j.eswa.2009.06.085>.

MENDOZA, Jorge E, CASTANIER, Bruno y GUÉRET, Christelle. Constructive Heuristics for the Multicompartment Vehicle Routing Problem with Stochastic Demands. En: Transportation Science. 2011. pp. 1–18. DOI 10.1287/trsc.1100.0353.

MENDOZA, Jorge E., et al. A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands. En: Computers & Operations Research [en línea]. Noviembre 2010. Vol. 37, no. 11, pp. 1886–1898. [Consultado Marzo 2011]. DOI 10.1016/j.cor.2009.06.015. Disponible en: <http://linkinghub.elsevier.com/retrieve/pii/S0305054809001713>.

NOVOA, C y STORER, R. An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. En: European Journal of Operational Research [en línea]. Julio 2009. Vol. 196, no. 2, pp. 509–515. [Consultado Junio 2010]. DOI 10.1016/j.ejor.2008.03.023. Disponible en: <http://linkinghub.elsevier.com/retrieve/pii/S0377221708003172>.

ÓLAFSSON, Sigurdur y KIM, Jumi. Simulation Optimization. En: Proceedings of the 2002 Winter Simulation Conference E. Yiicesan, C.-H. Chen, J. L. Snowdon, and J. M. Chames, eds. S.I.: IEEE. 2002. pp. 79–84.

REI, W., GENDREAU, M. y SORIANO, P. A Hybrid Monte Carlo Local Branching Algorithm for the Single Vehicle Routing Problem with Stochastic Demands. En: *Transportation Science* [en línea]. 22 Enero 2010. Vol. 44, no. 1, pp. 136–146. [Consultado Julio 2012]. DOI 10.1287/trsc.1090.0295. Disponible en: <http://transci.journal.informs.org/cgi/doi/10.1287/trsc.1090.0295>.

SECOMANDI, Nicola. Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands. En: *Computers & Operations Research* [en línea]. Septiembre 2000. Vol. 27, no. 11-12, pp. 1201–1225. DOI 10.1016/S0305-0548(99)00146-X. Disponible en: <http://linkinghub.elsevier.com/retrieve/pii/S030505489900146X>.

SECOMANDI, Nicola. y MARGOT, F. Reoptimization Approaches for the Vehicle-Routing Problem with Stochastic Demands. En: *Operations Research* [en línea]. 1 Enero 2009. Vol. 57, no. 1, pp. 214–230. [Consultado Julio 2012]. DOI 10.1287/opre.1080.0520. Disponible en: <http://or.journal.informs.org/cgi/doi/10.1287/opre.1080.0520>.

SECOMANDI, Nicola. A rollout policy for the vehicle routing problem with stochastic demands. En: *Operations Research*. 2001. Vol. 49, no. 5, pp. 796–802.

SHANMUGAM, Geetha, GANESAN, Poonthalir y VANATHI, P T. Meta Heuristic Algorithms for Vehicle Routing Problem with Stochastic Demands. En: *Journal of Computer Science*. 2011. Vol. 7, no. 4, pp. 533–542.

TAN, K, CHEONG, C y GOH, C. Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation. En: *European Journal of Operational Research* [en línea]. 1 Marzo 2007. Vol. 177, no. 2, pp. 813–839. [Accessed 28 January 2011]. DOI 10.1016/j.ejor.2005.12.029. Disponible en: <http://linkinghub.elsevier.com/retrieve/pii/S0377221706000208>.

TILLMAN, Frank A. The Multiple Terminal Delivery Problem with Probabilistic Demands The Multiple Terminal Delivery Problem with Probabilistic Demands. En: Transportation Science. 1969. Vol. 3, no. 3, pp. 192–204.

TRIPATHI, Mukul, KURIGER, Glenn y WAN, Hung-da. An Ant Based Simulation Optimization for the Vehicle Routing Problem with Stochastic Demands. En: Proceedings of the 2009 Winter Simulation Conference, IEEE. S.I.: s.n. 2009. pp. 2476–2487.

YANG, Wen-Huei, MATHUR, K y BALLOU, R. Stochastic Vehicle Routing Problem with Restocking. En: Transportation Science [en línea]. Abril 2000. Vol. 34, no. 1, pp. 99–112. Disponible en: <http://www.ncbi.nlm.nih.gov/pubmed/20422950>.