ISSN 0121-1129

Algoritmo PSO-Híbrido para solucionar el problema de ruteo de vehículos con entrega y recolección simultáneas

PSO-Hybrid algorithm for solving the vehicle routing problem with simultaneous pickup and delivery

Fecha de Recepción: 17 de octubre de 2013
Fecha de Aprobación: 07 de noviembre de 2013

Henry Lamos Diaz*
Silvia Adriana Galvan Nuñez**
Ludy Juliana Gonzalez Villamizar***
Camilo Cruz Jimenez****

Resumen

Se presenta la metaheurística de Optimización de Enjambre de Partículas (PSO) para la solución del Problema de Ruteo de Vehículos con Entrega y Recolección Simultáneas (VRPSPD). Se aplica una representación de la solución y un método de decodificación para implementar el PSO al VRPSPD. El método de decodificación inicia transformando una partícula en una lista de prioridades de clientes para entrar a las rutas y en una matriz de prioridades 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 validado usando 18 instancias

Abstract

In this paper, the Particle Swarm Optimization (PSO) algorithm for solving the Vehicle Routing Problem with Simultaneous Pickup and Delivery (VRPSPD) is presented. A solution representation is applied with random elements and a decoding method for implementing PSO for VRPSPD. The decoding method begins by transforming a particle into a customer's priority list to enter the routes and the vehicles priority matrix to serve each customer. Vehicle routes are constructed based on the customer's priority list and the vehicles priority matrix. The algorithm is tested using 18 instances available from the literature. Computational results show that the

* Ph.D. Universidad Industrial de Santander (Colombia). hlamos@uis.edu.co

** M.Sc. Universidad Industrial de Santander (Colombia). silvia.galvan@correo.uis.edu.co

*** Universidad Industrial de Santander (Colombia). ludy.gonzalez@correo.uis.edu.co

**** Universidad Industrial de Santander (Colombia). camilo.cruz@correo.uis.edu.co

disponibles en la literatura para problemas de 100, 200 y 400 clientes.

Palabras clave: PSO, VRPSPD, Ruteo de vehículos, Enjambre de partículas, Técnicas metaheurísticas, Técnicas heurísticas, Logística inversa.

proposed method gives good solutions for 100, 200 and 400 customers problems.

Keywords: PSO, Particle Swarm Optimization, VRPSPD**,** Particle Swarm, Vehicle Routing, Metaheuristics technics, Reverse Logistics.

I. Introducción

El Problema Ruteo de Vehículos (VRP, su sigla en inglés) consiste en la construcción de rutas desde un depósito hasta un número determinado de clientes ubicados en un espacio geográfico, con el fin de ofrecerles algún tipo de servicio con una flota de vehículos, siendo la entrega de bienes el más común. El diseño de sistemas de distribución efectivos puede representar un significativo ahorro de costos a las empresas [1].

Dada su vigencia en la gestión empresarial del ambiente económico actual, el cual exige la nivelación adecuada de flujos de bienes entre los diferentes agentes de la cadena de suministros para alcanzar mejores indicadores de competitividad, han surgido variaciones del problema, relacionadas con el tipo de actividad logística llevada a cabo en la asignación de rutas de transporte. Algunas de las mayores exigencias a los sistemas de distribución están dadas por las crecientes preocupaciones medioambientales, que han dirigido esfuerzos hacia la protección del ambiente tanto en la industria como en la legislación. Un mejor manejo de desperdicios y de energía consumida, que permita directamente una reducción de costos, ha generado que el desensamble, el reciclaje parcial y completo, la remanufactura y la reutilización de bienes formen parte de los nuevos enfoques de producción y administración de la cadena de suministros [2].

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 (VRPSPD, su sigla en inglés), propuesta por Min [3], quien se interesó por un transporte de libros en una biblioteca. En el VRPSPD, 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 restricciones de capacidad y tiempo, con el objetivo de minimizar costos, distancias o tiempos de viaje.

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), las cuales son atendidas en una sola visita del proveedor. Con esto, los minoristas pueden negociar la devolución a los productores del exceso de productos no vendidos, con efectos beneficiosos para ambas partes. Esto también ocurre en industrias que aplican el uso de estibas o contenedores para el transporte de mercancías, en las cuales se reutilizan estos elementos. Además, dado que las leyes gubernamentales fuerzan a las empresas a tomar responsabilidad por la vida útil de sus productos, estas se ven obligadas a colectar 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 [2].

El VRPSPD es un problema de optimización combinatoria de tipo NP-hard. Encontrar soluciones exactas requiere de un tiempo de cómputo significativamente grande, debido a la complejidad algorítmica de tipo exponencial del problema [4]. Si bien el problema puede ser abordado mediante métodos exactos para instancias pequeñas, los casos reales de aplicación del VRPSPD son siempre problemas de gran tamaño [5]. Por tal motivo, se han aplicado técnicas heurísticas y metaheurísticas que proporcionan buenas soluciones al problema de optimización en tiempos de cómputo razonables, las cuales se presentan a continuación.

Dethloff [2] y Nagy y Salhi [6] propusieron heurísticas de inserción de clientes. Nagy y Salhi [7] propusieron una heurística de búsqueda local con cuatro fases para resolver el VRPSPD. Tang y Galvao [8] desarrollaron un algoritmo de Búsqueda Tabú que combina heurísticas de reubicación

de clientes dentro y entre rutas como el 2-opt. Dell'Amico *et al.* [9] publicaron por primera vez un trabajo de investigación en métodos exactos para resolver el VRPSPD, basado en Generación de Columnas, programación dinámica y el método de Branch-and-Price (híbrido de branch-and-bound y generación de columnas), para problemas de menos de 40 clientes.

Wassan *et al.* [10] diseñaron un método de Búsqueda Tabú Reactivo que consiste en una fase de construcción mediante una modificación del método de barrido y una segunda fase de mejoramiento basada en la heurística *1-intercambio.*

Biancessi y Righini [11] 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.

Gajpal y Abad [12] presentaron una metaheurística de optimización de colonias 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 multirruta, e intercambio de subsecuencias multirruta.

Ai y Kachitvichyanukul [13] propusieron una estrategia de PSO con múltiples estructuras sociales aplicada al VRPSPD, en la que se construyen rutas de acuerdo con listas de prioridad de clientes para entrar a las rutas de acuerdo con la posición de las partículas del enjambre generadas en cada iteración.

Zachariadis y Kiranoudis [14] 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. Con esta metodología se mejoraron resultados obtenidos mediante otros métodos.

En el presente trabajo de investigación se propone un algoritmo híbrido para la solución VRPSPD, utilizando un enfoque de optimización de enjambre de partículas (PSO); para ello se implementa un método de decodificación que consiste en la creación de listas de prioridad de clientes para ser asignados a las rutas y una matriz de prioridad de vehículos para servir a los clientes, a partir de la posición de las partículas del enjambre. Con base en esas prioridades, se utiliza la técnica heurística de construcción de rutas *cheapest insertion heuristic.* Adicionalmente, cada ruta emergente es mejorada con la heurística de intercambios *2-opt*.

El algoritmo propuesto fue implementado en MatLab®, y para evaluar su desempeño se utilizaron 18 instancias disponibles en la literatura, las cuales fueron adaptadas por Tang y Galvão [8] al VRPSPD.

El artículo está organizado de la siguiente manera: en la primera sección se presenta una generalización del VRPSPD y su aplicación a problemas de la industria; en la segunda sección se define el método PSO y su implementación para solucionar el VRPSPD; en la tercera sección se presentan los resultados computacionales, y, finalmente, en la cuarta sección son presentadas las conclusiones y recomendaciones.

II. Formulación del problema VRPSPD

El problema VRPSPD se puede formular de la siguiente manera [13]:

Sea ***G=(V,A)*** un grafo en donde el conjunto $V=\{v_0, v_1, \dots, v_n\}$ representa a los clientes y al depósito, $A= \{(v_i, v_j) | v_i, v_j \in V, i \neq j\}$ es el

conjunto de arcos que conectan al cliente v_i con el cliente v_j. Además, las distancias y los tiempos de viaje entre v_i y v_j se denotan como (d_{ij}) y (t_{ij}), *respectivamente*. 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 $\boldsymbol{p_i}$, de entrega $\boldsymbol{q_i}$ y un tiempo de servicio $\boldsymbol{s_i}$. Cada vehículo tiene un límite de duración de servicio ***D***. El VRPSPD consiste en el diseño de un grupo de máximo ***m*** rutas recorridas por vehículos de capacidad ***Q,*** satisfaciendo la demanda de los clientes y minimizando el costo total de transporte.

De acuerdo con [13], el VRPSPD también se puede formular como el siguiente problema de programación lineal entera mixta.

Minimizar:

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

sujeto a:

$$\sum_{k=1}^{m} x_{ijk} = 1 \text{ para } 1 \le j \le n \qquad (2)$$

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

$$\sum_{j=1}^{n} x_{0jk} \le 1 \text{ para } 1 \le k \le m \qquad (4)$$

$$\delta_{ik} + S_i + t_{ij} + \delta_{jk} \le (1 - x_{ijk})M \text{ para } 0 \le i \le n,\ 1 \le j \le n,\ 1 \le k \le m \quad (5)$$

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

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

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

$$\sum_{i=0}^{n} y_{ijk} + (p_j - q_j) * \sum_{i=0}^{n} x_{ijk} = \sum_{i=1}^{n+1} y_{ijk} \text{ para } 1 \le j \le n,\ 1 \le k \le m \quad (9)$$

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

$$y_{ijk} \ge 0, \text{ para } 0 \le i \le n,\ 1 \le j \le n+1,\ 1 \le k \le m \qquad (11)$$

$$\delta_{ijk} \ge 0, \text{ para } 0 \le i \le n,\ 1 \le j \le n+1,\ 1 \le k \le m \qquad (12)$$

donde,

x_{ijk} es una variable binaria que indica si el arco (v_i,v_j) es recorrido o no por el vehículo ***k**, es decir,*

$x_{ijk} = 1$ *si el vehículo* ***k*** *recorre el arco* (v_i,v_j),

$x_{ijk} = 0$ *si el vehículo* ***k*** *no recorre el arco* (v_i,v_j),

y_{ijk} es la carga del vehículo ***k*** que recorre el arco (v_i,v_j),

δ_{ik} es el tiempo de inicio de servicio del cliente ***i*** efectuado por el vehículo ***k.***

Donde la función objetivo (1) representa el costo total de transporte, teniendo en cuenta el costo fijo ***f*** por la utilización de cada vehículo, y ***g*** es el costo variable por unidad de distancia recorrida.

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. 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. La restricción (4) indica que un vehículo puede servir como máximo a una ruta. La restricción (5) relaciona el tiempo de inicio de servicio entre un cliente y otro. 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*** después de 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.

III. Optimización de enjambre de partículas para el VRPSPD

En esta sección se presenta el algoritmo de optimización de enjambre de partículas para resolver el VRPSPD descrito previamente, la representación de la solución y el método de decodificación de la solución [13].

A. Algoritmo PSO

El PSO es un algoritmo poblacional de búsqueda basado en la simulación del comportamiento social de aves de una bandada [15]. 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 individuos resulta en estrategias colectivas de solución de problemas.

En el PSO los individuos se conocen como partículas que pertenecen a un espacio de búsqueda $X \in °^{n}$. Los cambios de posición de las partículas dentro del espacio de búsqueda están basados en las tendencias psicosociales de individuos de emular el éxito de otros individuos, es decir, influenciados por la experiencia o conocimiento de sus vecinos. Como resultado, el comportamiento colectivo que surge es que todo el enjambre converge al estado que es mejor para todos sus miembros [16].

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 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 de la 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 esta desarrolla en la medida en que se desplaza por el espacio de búsqueda, conocido como *pbest*, y se denota como el vector

$$\Psi_l = [\psi_{l1}, \psi_{l2}, \dots, \psi_{lH}]. (13)$$

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,* y se denota como el vector

$$\Psi_g = [\psi_{g1}, \psi_{g2}, \dots, \psi_{gH}]. (14)$$

Este vector es denominado *comportamiento social*, puesto 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 una partícula en la iteración es el vector que se denota como sigue en la ecuación (15), donde y $l = 1, \dots, L$ y $t = 1, \dots, T$.

$$\Theta_l(\tau) = [\theta_{l1}, \theta_{l2}, \dots, \theta_{lH}] (15)$$

La actualización del vector de posición está asociada a la velocidad, que se compone de los términos de aprendizaje de la partícula y del peso inercial de esta. La velocidad para cada partícula del enjambre en la iteración es el vector

$$\Omega_l(\tau) = [\omega_{l1}, \omega_{l1}, \dots, \omega_{lH}]. (16)$$

De acuerdo con esto, la posición de la partícula cambia en cada iteración de la forma:

$$\Theta_l(\tau + 1) = \Theta_l(\tau) + \Omega_l(\tau + 1). (17)$$

Para utilizar el algoritmo aplicado a problemas de optimización combinatoria 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 direccione 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 algoritmo utilizado para resolver el VRPSPD se presenta a continuación. En primer lugar se inicializa el enjambre y se evalúa el desempeño de la función objetivo mediante el método de codificación; de esta manera se actualiza la información de los comportamientos del enjambre. Posteriormente se actualizan la velocidad y la posición, se evalúa el criterio de parada y, luego, se continúa con la siguiente iteración.

Sean

- u Número aleatorio distribución uniforme [0,1]
- $w(\tau)$ Peso inercial de la -ésima iteración
- c_p Constante de aceleración de la posición pbest
- c_g Constante de aceleración de la posición gbest
- θ_{max} Valor de posición máximo
- θ_{min} Valor de posición mínimo
- R_l El 1-ésimo conjunto de rutas de vehículos
- $Z(\Theta_l)$ Valor de desempeño de Θ_l
- ω_{lh} Componente de la velocidad de la partícula en la dimensión *h*.

De acuerdo con [13] y con estas variables se tiene la siguiente variante del algoritmo PSO.

Algoritmo 1. PSO para el VRPSPD

1. τ=*1, para todas las partículas inicializar* Θ_l, $\Omega_l = 0$ *y* $\Psi_l = \Theta_l$

2. Decodificar $\Theta_l(\tau)$ *en un conjunto de rutas* R_l

3. Guardar las mejores posiciones *pbest* y *gbest*

4. **Mientras** $(\tau < T)$

5. Actualizar *inercia, velocidad y posición de las partículas*

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

$$\omega_{lh}(\tau + 1) = w(\tau) * \omega_{lh}(\tau) + c_p u\left(\Psi_{lh} - \theta_{lh}(\tau)\right) + c_g u\left(\Psi_{gh} - \theta_{lh}(\tau)\right)$$

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

$$\theta \quad \tau + 1 \quad = \theta \quad \tau \quad + \omega \quad \tau + 1$$

6. Evaluar y decodificar *el nuevo* $\Theta_l(\tau)$ *en un conjunto de rutas* R_l

7. Actualizar y guardar *las mejores soluciones del enjambre pbest* y *gbest.*

8. $\tau = \tau + 1$

9. **Fin mientras**

1) Representación de la solución: La representación de la solución del VRPSPD a través del paradigma PSO consta 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 Fig. 1 se muestra la relación del área que recorre un vehículo representada por su punto de orientación.

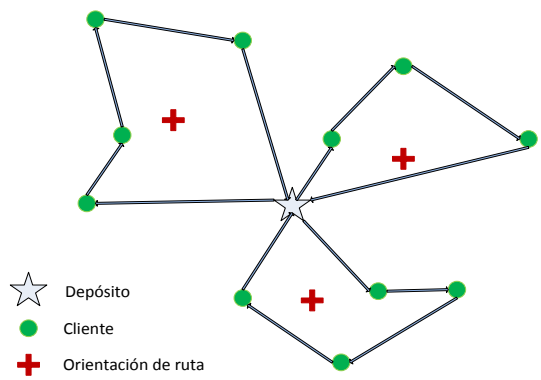

Fig. 1. Rutas y puntos de orientación de rutas

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.

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 VRPSPD para n clientes y m vehículos requiere partículas con dimensiones. Las primeras dimensiones corresponden al número de clientes y las siguientes dimensiones están 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 Fig. 2.

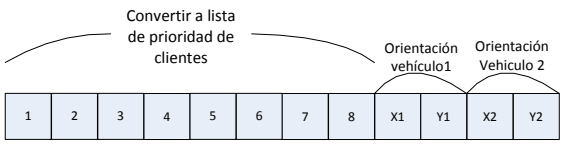

Fig. 2. Vector posición de una partícula del enjambre

2) *Método de decodificación:* Para convertir la codificación de los componentes de la heurística en una solución al problema original se utiliza un algoritmo de decodificación que consta de tres pasos:

- En el primer paso se construye la lista de prioridad de clientes, siguiendo la regla de mayor prioridad al cliente con menor valor a partir de las primeras n dimensiones de la partícula. La manera más simple de implementar esta regla es ordenando de forma ascendente los valores de posición de las n dimensiones y tomando el índice de dimensión como la lista de prioridades.
- 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.
- El último paso de decodificación es construir rutas con base en la lista de prioridad de clientes y en 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 2-opt de mejoramiento a la ruta emergente.

Este método es presentado de forma detallada en el siguiente algoritmo.

Algoritmo 2. Método de decodificación

A. Construir lista de prioridad de clientes (U)

1. Construir el conjunto $S = (1,2,\dots,n)$ y $U = \emptyset$

2. Mientras $S = \emptyset$
3. Seleccionar cliente *c* del conjunto *s* tal que corresponda a la menor dimensión de θ_{lh}
4. Agregar *c* a la última posición de *U*
5. Eliminar *c* del conjunto *S*
6. Fin mientras

B. Construir matriz de prioridad de vehículos (*W*)

1. Para $j = 1 \ldots m$, definir la posición de referencia de los m vehículos
2. $Xref_j = \theta_{i.n+2j-1}$ y $Yref_j = \theta_{i.n+2j}$
3. Para cada cliente i, $i = 1 \ldots n$
4. Calcular la distancia euclidiana entre el cliente i y los todos puntos de orientación
5. Mientras $S = \emptyset$
6. Construir conjunto $S = (1,2, \ldots, m)$ y $W_i = \emptyset$
7. Seleccionar vehículo *c* del conjunto *s* tal que tenga la menor distancia euclidiana al cliente *i*
8. Agregar *c* a la última posición del conjunto W_i
9. Fin mientras

C. Construir rutas de vehículos

1. *k=1*
2. **Mientras** *k=1*
3. Agregar clientes uno a uno a la ruta
4. $c = U_k$, $b = 1$
5. **Mientras** $b < m$
6. $j = W_{c,b}$
7. Hacer a *c* como candidato a insertar en la posición que genere el menor costo adicional en la ruta R_{lj}, según heurística de inserción más barata
8. Evaluar la carga y el tiempo de servicio de la nueva ruta
9. **Si** (cumple con condiciones de factibilidad)
10. Actualizar la ruta R_{lj} y reoptimizar con 2-opt

Fin_si

11. $b = b + 1$

Fin mientras

12. $k = k + 1$

Fin mientras

IV. Resultados computacionales

Los experimentos numéricos efectuados para evaluar el desempeño del PSO se llevaron a cabo aplicando el método en un conjunto de instancias desarrolladas para el VRPSPD. Se trata de 18 instancias de gran tamaño, introducidas por Tang y Galvão [8], quienes adaptaron instancias desarrolladas inicialmente para el CVRP por Solomon [17] y Gehring & Homberger [18].

Cada instancia está conformada por los siguientes elementos: la matriz ***d,*** que es la distancia euclidiana entre los clientes y el depósito; 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; el número de vehículos ***m*** es igual a la demanda total de entrega y recogida de los clientes, dividido sobre la capacidad de los vehículos disponibles. La Tabla 1 resume las características de las 18 instancias utilizadas, las cuales se encuentran disponibles en http://users.ntua.gr/ezach/

Tabla 1
Instancias para VRPSPD

Instancia	n	Q	$\sum p_i$	$\sum q_i$
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

Los parámetros utilizados para aplicar el algoritmo PSO se muestran en la Tabla 2. A partir de un análisis de sensibilidad se determinó considerar el número de iteraciones como el criterio de parada; además, todas las instancias se ejecutaron con 50 iteraciones.

De acuerdo con [19], los siguientes son los parámetros del algoritmo PSO.

Tabla 2
Parámetros para el PSO

Parámetros	Valor
Número de partículas	$N = 50$
Número de vecinos	$K = 5$
Peso inercial Inicial	$\omega(1) = 0.9$
Peso inercial final	$\omega(T) = 0.4$
Constante de Aceleración Pbest	$c_p = 1$
Constante de aceleración Gbest	$c_g = 1$

El algoritmo PSO fue implementado en MatLab, versión R2012a, en un equipo con procesador Intel Core i5 con 4 GB de memoria RAM instalada.

Las soluciones obtenidas fueron comparadas con el método Búsqueda Tabú (TS) propuesto por Tang-Montanè y Galvão [8]. La comparación de los resultados se realizó con base en la desviación de los valores de respuesta (Gap %) presentados en las Tablas 3, 4 y 5 para los grupos de instancias de 100, 200 y 400 clientes respectivamente.

Tabla 3
Resultados para instancias de 100 clientes

Problemas de 100 clientes							
Instancia	*Q*	***Mejor solución TS***		***Mejor solución PSO***		*Gap (%)*	
		Z	*k*	*Z*	*k*	*Z*	*k*
r101	200	1042.62	12	1095.7	13	5.091	8.33
r201	1000	671.03	3	671.6	3	0.085	0.0
c101	200	1259.79	17	1316.7	17	4.517	0.0
c201	700	666.01	5	668.68	5	0.401	0.0
rc101	200	1094.15	11	1140.9	11	4.273	0.0
rc201	1000	674.46	3	679.04	3	0.679	0.0

En los resultados obtenidos para las instancias de 100 clientes se pudo observar que la desviación entre los resultados obtenidos en el presente trabajo y los obtenidos por el método TS no superan el 6% 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. Para el caso del número de vehículos, se encontró la misma solución en todos los casos, excepto en la instancia r101, para la cual se utilizó 1 vehículo más que en la solución arrojada por la Búsqueda Tabú.

En los resultados obtenidos para las instancias de 200 clientes se pudo observar que la desviación no supera el 14% para la función de costos.

Tabla 4
Resultados para instancias de 200 clientes

Problemas de 200 clientes							
Instancia	***Q***	*Mejor solución TS*		*Mejor solución PSO*		***Gap (%)***	
		Z	***K***	***Z***	***k***	***Z***	***K***
R121	200	3447.2	23	3763.6	26	9.178	13.043
R221	1000	1690.67	5	1708.7	5	1.066	0.000
C121	200	3792.62	29	4197.8	30	10.683	3.448
C221	700	1767.58	9	1883.6	10	6.564	11.111
RC121	200	3427.19	24	3768.9	25	9.971	4.167
RC221	1000	1645.94	5	1864	5	13.248	0.000

El resultado con mayor desviación corresponde a la instancia RC221, para la cual, sin embargo, se utilizó el mismo número de vehículos que en la solución de TS. También se puede observar que los menores valores de desviación corresponden a los problemas C221 y R221, donde la capacidad de la flota es grande. Para el caso de los vehículos, la mayor desviación fue de 13.043%.

Tabla 5
Resultados para instancias de 400 clientes

Problemas de 400 clientes							
Instancia	*Q*	*Mejor solución TS*		*Mejor solución PSO*		*Gap (%)*	
		Z	*k*	*Z*	*k*	*Z*	*K*
R141	200	10027.81	54	10874	58	8.438	7.407
R241	1000	3695.26	10	4032.6	10	9.129	0.000
C141	200	11676.27	65	12846	68	10.018	4.615
C241	700	3732	15	4105.5	16	10.008	6.667
RC141	200	9883.31	52	10990.53	57	11.203	9.615
RC241	1000	3603.53	11	3939.8	11	9.332	0.000

Para instancias de 400 clientes, con capacidad de vehículos mayor, se encontró un número de vehículos más cercano a la solución de TS que para las instancias con capacidad de vehículos menor. En las instancias R241 y RC241 se utilizó el mismo número de vehículos que en las soluciones arrojadas por el método TS.

A partir de los resultados se realizaron procedimientos estadísticos para verificar el comportamiento del algoritmo cuando se varían dos parámetros del VRPSPD: la capacidad de los vehículos y el número de clientes. Se buscaba verificar si existía algún efecto significativo en el desempeño del algoritmo producido por alguno de ellos o por la interacción entre los dos, para lo cual se procedió de la siguiente manera.

En todos los casos, el desempeño del algoritmo se evaluó mediante el porcentaje de desviación con respecto a la solución obtenida por el método TS como

$$Desv_Z = \frac{Z_{pso} - Z_{TS}}{Z_{TS}} * 100\% \quad (18)$$

donde,

Desv_Z: porcentaje de desviación del valor Z respecto a la solución obtenida por el método TS

Z_{pso}: valor de la función objetivo obtenido por el método PSO

y Z_{TS} : valor de la función objetivo obtenido por el método Búsqueda Tabú.

Adicionalmente, se evaluó el impacto sobre la desviación en el número de vehículos asignados con respecto al número de vehículos arrojados por el método TS como

$$\text{Desv}_K = \frac{K_{pso} - K_{TS}}{K_{TS}} * 100\% \quad (19)$$

donde,

Desv_k: porcentaje de desviación del número de vehículos respecto a la solución obtenida por el método TS;

K_{pso}: número de vehículos obtenido por el método PSO

y K_{TS} : número de vehículos obtenido por el método Búsqueda Tabú.

V. Análisis de resultados

El análisis permitió establecer que la variación tanto en el número de clientes como en la capacidad de carga de los vehículos tiene impacto estadísticamente significativo sobre . El análisis de varianza y pruebas de rangos múltiples permitió establecer que el algoritmo arroja soluciones con menor desviación con respecto a los valores de referencia cuando las instancias tienen vehículos con capacidad grande (1000 u.), que cuando

tienen capacidad pequeña (200 u.). Así mismo, se obtiene que el funcionamiento del algoritmo tiene mayor aproximación a las soluciones de referencia cuando las instancias son pequeñas. En general, para instancias pequeñas y con capacidad de carga de los vehículos grande, se encontraron los mejores resultados.

Por otra parte, el análisis efectuado para permite establecer que el algoritmo es robusto frente a la variación de la capacidad de carga de los vehículos; es decir, que este factor no tiene un efecto significativo sobre el desempeño del algoritmo. La variación del número de clientes, en cambio, sí tiene un efecto significativo sobre el desempeño del algoritmo. Para instancias de 100 clientes se tienen soluciones con menor desviación con respecto a los valores de referencia, que para las instancias más grandes. Para aquellas instancias de 200 y 400 clientes, el algoritmo encontró soluciones con la misma desviación.

VI. Conclusiones y trabajos futuros

En el presente trabajo se resolvió el VRPSPD con la metaheurística PSO, utilizando un método de decodificación para la construcción de las rutas vehiculares. Adicionalmente, se efectuaron experimentos numéricos para evaluar el desempeño del PSO sobre un conjunto de instancias. Por último, se llevó a cabo un diseño factorial para determinar el impacto de los parámetros del VRPSPD sobre la función objetivo.

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 mayor dificultad en encontrar soluciones que tuvieran menor desviación en su costo, respecto a los valores de referencia.

También, se puede 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 se hace menor.

La interacción entre el tamaño del problema y la capacidad de los vehículos implica una variación significativa en la obtención de resultados con el algoritmo PSO. 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 este 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 con 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 interrutas, y no únicamente intrarrutas, para mejorar las secuencias emergentes.

Referencias

[1] J.-F. Chen and T.-H. Wu. "Vehicle routing problem with simultaneous deliveries and pickups". *J. Oper. Res. Soc.*, 57(5): 579–587, Jul. 2005.

[2] J. Dethloff. "Vehicle routing and reverse logistics: The vehicle routing problem with simultaneous delivery and pick-up". *OR Spektrum*, 23(1): 79-96, Feb. 2001.

[3] H. Min. "The multiple vehicle routing problem with simultaneous delivery and pickup points". *Transp. Res.*, 23: 337-386, 1989.

[4] J.-F. Cordeau, M. Gendreau, G. Laporte, J.-Y. Potvin, and F. Semet. "A guide to vehicle routing heuristics". *J. Oper. Res. Soc.*, 53(5): 512-522, May 2002.

[5] P. Toth and D. Vigo. "Models, relaxations and exact approaches for the capacitated vehicle routing problem". *Discret. Appl. Math.*, 123(1-3): 487-512, Nov. 2002.

[6] S. Salhi and G. Nagy. "A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling". *J. Oper. Res. Soc.*, 50(10): 1034-1042, Oct. 1999.

[7] 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, Apr-2005.

[8] F. A., Tang Montané and R. D. Galvão, "A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service". *Comput. Oper. Res.*, 33(3): 595-619, Mar. 2006.

[9] M. D. Amico, G. Righini, and M. Salani. *A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection*, 2005.

[10] N. A. Wassan, A. H. Wassan, and G. Nagy, "A reactive tabu search algorithm for the vehicle routing problem with simultaneous pickups and deliveries". *J. Comb. Optim.*, 15(4): 368-386, Jun. 2007.

[11] N. Bianchessi and G. Righini. "Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery". *Comput. Oper. Res.*, 34(2): 578-594, Feb. 2007.

[12] Y. Gajpal and P. Abad. "An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup". *Comput. Oper. Res.*, 36(12): 3215-3223, Dec. 2009.

[13] T. J. Ai and V. Kachitvichyanukul. "A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery". *Comput. Oper. Res.*, 36, 2009.

[14] E. E. Zachariadis and C. T. Kiranoudis. "A local search metaheuristic algorithm for the vehicle routing problem with simultaneous

pick-ups and deliveries". *Expert Syst. Appl.*, 38(3): 2717-2726, Mar. 2011.

[15] J. Kennedy and R. Eberhart. "Particle swarm optimization". *Proc. ICNN'95 - Int. Conf. Neural Networks*, 4: 1942-1948, 1995.

[16] 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.

[17] M. M. Solomon. "Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints". *Oper. Res.*, 35(2): 254-265, 1987.

[18] H. Gehring, J. Homberger, and D.- Hagen. *A Parallel Two-phase Metaheuristic for Routing Problems with Time Windows Abstract*. Hagen, Germany, 1999.