Optimización Basada en Colonias de Hormigas: Una aplicación a la distribución de sobres

Adrián Ojeda[1], Benjamín Barán[2 3] y Horacio Kuna[1 4]

[1] Universidad Nacional de Itapúa – UNI – Encarnación, Paraguay
adrianojedabenitez@gmail.com
[2] Universidad Nacional del Este – UNE – Ciudad del Este, Paraguay
[3] Universidad Nacional de Asunción – UNA – Asunción, Paraguay
bbaran@cba.com.py
[4] Universidad Nacional de Misiones – Posadas, Argentina
hdkuna@unam.edu.ar

Resumen. El presente trabajo propone un novedoso algoritmo basado en la reconocida meta-heurística de optimización por *Colonia de Hormigas* para resolver el problema de distribución de sobres de una empresa paraguaya distribuidora de extractos bancarios. El problema es modelado como un MTSP (*Multiple Traveling Salesman Problem)* multi-objetivo que optimiza cuatro objetivos simultáneamente, específicamente: (1) minimiza la flota de vehículos (número de motocicletas a utilizar), (2) minimiza la distancia total del recorrido, (3) minimiza el tiempo total de entrega y (4) maximiza la ganancia total de la empresa. Resultados Experimentales demuestran que el algoritmo propuesto resuelve eficientemente el problema logístico de distribución de sobres en la ciudad de Encarnación.

Palabras Claves: *Multiple Traveling Salesman Problem, Ant Colony Optimization,* Optimización Multi-Objetivo.

1 Introducción

Este trabajo propone resolver un problema logístico de distribución de sobres en el cual (*n-1)* clientes deben ser visitados por hasta *m* conductores que parten desde el depósito central de una empresa, utilizando una flota homogénea de vehículos. A este problema se le puede modelar como un MTSP (*Multiple Traveling Salesman Problem*) [1]. El MTSP es una generalización del paradigmático problema TSP - *Traveling Salesman Problem* en el que se consideran varios viajantes que se distribuyen la tarea de visitar a todos los clientes asignados. El TSP puede ser simétrico o asimétrico [2]. La solución del TSP necesariamente resulta en un ciclo Hamiltoniano [3]; es decir, un ciclo que partiendo de un nodo (típicamente, el depósito central de una empresa) pase por todos los nodos (clientes) restantes, una única vez por cada nodo, regresando finalmente al nodo de origen.

Por analogía con el TSP, el problema del MTSP consiste en visitar a los (*n-1*) clientes, utilizando ahora hasta *m* vendedores que parten y regresan al depósito central, minimizando el costo total de las *m* rutas [1]. Típicamente, la distancia entre ciudades es considerada como el costo de cada tramo entre ciudades, por lo que una matriz de costos D = $\{d_{ij}\}$ es dato del problema. En este problema no se tiene en cuenta la capacidad de cada vehículo utilizado por los *m* vendedores [4].

Cabe aquí mencionar que el problema MTSP es tradicionalmente planteado como mono-objetivo [1], minimizando una única función de costo, aunque resulta fácil generalizar el problema a un contexto multi-objetivo al considerar la minimización simultánea de diferentes matrices de costo representando objetivos diversos como distancia entre ciudades, tiempo de viaje, o costo del viaje (gasolina + peajes), entre otras posibilidades, e incluso considerando también la minimización de la flota de vehículos a ser utilizada.

Para resolver el problema de distribución de sobres en un contexto multi-objetivo, se propone utilizar una de las meta-heurísticas más utilizadas para resolver problemas de ruteo, la Optimización basada en Colonias de Hormigas (ACO – *Ant Colony Optimization*) [5] [6], reconocida en la literatura especializada por sus excelentes resultados experimentales [7] [8].

Para este trabajo, se propone una variante del MMAS - *Max-Min Ant System*, propuesta por Stützle y Hoos [9], que demostró excelentes resultados al resolver el TSP tradicional que considera un solo objetivo [10]. La variante propuesta en este trabajo modifica la propuesta original de Stützle y Hoos [9] para abordar el problema multi-objetivo, combinando algunas de las ideas originalmente propuestas por Pinto y Barán, quienes propusieron el M3AS – *Multiobjective Max-Min Ant System* [11], junto a varias mejoras propuestas por Barán y Schaerer en [12].

Para finalizar, cabe señalar que este trabajo propone resolver el problema logístico concreto de una empresa distribuidora de sobres que necesita optimizar simultáneamente cuatro funciones objetivo: (1) minimizar la cantidad total de vehículos a ser utilizados, (2) minimizar la distancia total del recorrido, (3) minimizar el tiempo total de la entrega y (4) maximizar la ganancia total de la empresa. El resto del artículo queda organizado de la siguiente manera: la sección II presenta una breve introducción a la Optimización Multi-Objetivo, mientras que la sección III propone el modelo matemático del Problema de Distribución de Sobres. La sección IV describe la solución propuesta mientras que la sección V exhibe los resultados experimentales y finalmente la sección VI presenta las conclusiones y propone algunos trabajos futuros.

2 Optimización Multi-Objetivo

En la sección anterior se ha mencionado que se desean optimizar cuatro funciones objetivo que serán denotadas como $F_1(\Psi)$, $F_2(\Psi)$, $F_3(\Psi)$ y $F_4(\Psi)$ respectivamente, donde Ψ representa una posible solución del problema. Para encontrar una solución óptima en un contexto multi-objetivo se debe encontrar una solución Ψ que optimice simultáneamente todas las funciones objetivo; es decir, una solución que no pueda ser mejorada en ninguna función objetivo sin deteriorar otros objetivos, lo que se conoce como solución *no-dominada* o solución *Pareto óptima* [12], por lo que se pasa a definir el concepto de *Dominancia Pareto*.

En un contexto multi-objetivo, dadas dos soluciones factibles *u* y *v*, se dice que *u* domina a *v* si es mejor o igual que *v* en cada uno de los objetivos considerados además de ser estrictamente mejor que *v* en al menos un objetivo. Lógicamente, al resolver un problema de optimización multi-objetiva, solo nos interesa encontrar las soluciones *no-dominadas*. Es relevante mencionar que si los objetivos son contradictorios, no es posible encontrar una única solución óptima que domine a todas las demás, por lo que en la absoluta mayoría de los problemas prácticos se obtiene un conjunto de soluciones *no-dominadas* que se conoce como *Conjunto Pareto*, cuya imagen en el espacio objetivo se conoce como *Frente Pareto* [13]. Antes de concluir, se recuerda que dadas dos soluciones *u* y *v*, se dice que *u* es *no-comparable* con *v* (y vice versa) si ninguna domina a la otra. En consecuencia, todas las soluciones que conforman el *Conjunto Pareto*, son *no-comparables* entre sí [13].

3 Modelo Matemático

Sea:

$V = \{v_0, v_1, v_2 \dots v_{n-1}\}$	el conjunto de nodos de un grafo $G\ (V, E)$, donde v_0 representa al depósito.
$\bar{V} = \{\bar{v}_1, \bar{v}_2 \dots \bar{v}_{n-1}\}$	el conjunto de $(n-1)$ clientes a visitar donde $\bar{v}_i$ representa a un cliente, $i \in \{1, 2, \dots, (n-1)\}$.
$K = \{k_1, k_2 \dots k_m\}$	el conjunto de vehículos disponibles para realizar la entrega, donde k_i representa a un vehículo, $i \in \{1, 2, \dots, m\}$.
d_{ij}	la distancia entre el nodo v_i y v_j., donde se puede dar que $d_{ij} \neq d_{ji}$ dado que este trabajo considera el caso más general (no simétrico).
t_{ij}	el tiempo entre el nodo v_i y v_j, donde nuevamente se puede dar que $t_{ij} \neq t_{ji}$.
C_{nafta}	una constante que indica el consumo aproximado de combustible.
C_{jornal}	una constante que indica el pago al chofer (premio en efectivo) por cada sobre debidamente entregado.
CostoFijo	una constante que indica el costo fijo de cada vehículo (mantenimiento, cambio de aceite, etc.).
G_i	la ganancia por sobre entregado, donde $i \in \{1, 2, \dots, (n-1)\}$.
M_i	multa por sobre no entregado, donde $i \in \{1, 2, \dots, (n-1)\}$.
W	tiempo máximo de trabajo permitido por día.
Ψ	una solución del problema indicando el recorrido a seguir por cada uno de los *m* vehículos disponibles.

donde además se utilizan las variables binarias (*flags*):

$$x_{ijk} = \begin{cases} 1 \ si\ el\ vehiculo\ k\ viaja\ de\ i\ a\ j. \\ 0\ en\ caso\ contrario. \end{cases}$$

$$\bar{x}_i = \begin{cases} 1 \; si \; se \; entrega \; el \; sobre \; i. \\ 0 \; en \; caso \; contrario. \end{cases}$$

Se hace notar que dada una solución Ψ, las variables x_{ijk} así como las variables $\bar{x}_i$ quedan unívocamente definidas, por lo que da lo mismo hablar de una solución Ψ, o de sus correspondientes variables x_{ijk} y $\bar{x}_i$, lo que permite expresar las cuatro funciones objetivo consideradas, como se indica a continuación.

1. Número de vehículos

$$F_1\,(\Psi) = \sum_{k=1}^{m} \max_{i,j}\,(x_{ijk}) \qquad \forall i,j \in \bar{V} \tag{3.1}$$

donde $F_1\,(\Psi)$ representa la cantidad de vehículos a ser utilizados.

2. Distancia total del recorrido

$$F_2\,(\Psi) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \sum_{k=1}^{m} x_{ijk}\, d_{ij} \tag{3.2}$$

donde $F_2\,(\Psi)$ representa la distancia total recorrida por los *m* vehículos utilizados para repartir los sobres.

3. Tiempo total del recorrido

$$F_3\,(\Psi) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \sum_{k=1}^{m} x_{ijk}\, t_{ij} \tag{3.3}$$

donde $F_3(\Psi)$ representa el tiempo total utilizado por los *m* vehículos en sus recorridos.

4. Ganancia total de la entrega

$$F_4\,(\Psi) = \sum_{i=1}^{n-1} \bar{x}_i G_i - \sum_{i=1}^{n-1} (1-\bar{x}_i) M_i - Cnafta\, F_2(\Psi) - \sum_{i=1}^{n-1} \bar{x}_i . Cjornal - Costofijo \tag{3.4}$$

donde $F_4(\Psi)$ representa la ganancia total de la entrega.

sujeto a:

$$\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} x_{ijk}\, t_{ij} \;\le\; W \qquad \forall k \in \{1, \dots, m\} \tag{3.5}$$

Esta restricción impide que un vehículo (*k)* trabaje un número de horas mayor al máximo número de horas (*W*) permitido en una solución.

4 Solución Propuesta

En este apartado se plantea la solución propuesta para un negocio de distribución de extractos de tarjetas de crédito de la localidad de Encarnación-Paraguay. El problema se plantea como un MTSP para el caso asimétrico. Como fuera adelantado, la solución propuesta se basa en el *Max-Min Ant System* [9], debidamente modificado con ideas presentadas en [8], [10] y [11] de forma a poder utilizarlo en el contexto multi-objetivo en que se plantea el problema de distribución de sobres.

En general, la idea detrás de la meta-heurística ACO es utilizar agentes artificiales (llamados *hormigas*) que vayan construyendo una solución factible eligiendo la siguiente ciudad a visitar en forma aleatoria, utilizando una distribución de probabilidad como la presentada en (4.1), basada en la utilización de dos matrices: (1) la matriz de feromonas $\boldsymbol{\tau} = \{\tau_{ij}\}$ que se utiliza para guardar la información aprendida y (2) la matriz de visibilidad $\boldsymbol{\eta} = \{\eta_{ij}\}$ que guarda información heurística (cuan deseable es a priori ir de una ciudad a otra). Considerando que este trabajo considera un contexto multi-objetivo, se propone utilizar dos visibilidades, inspiradas en Pinto y Barán [11] y basadas en los dos objetivos considerados a priori principales: minimizar el tiempo y la distancia total de un recorrido, objetivos de los cuales dependen de alguna forma los otros dos objetivos restantes.

La ecuación propuesta en este trabajo, utilizando dos visibilidades es entonces:

$$p_{ij} = \begin{cases} \dfrac{\tau_{ij}^{\alpha} * \left[\eta_{d_{ij}}^{\beta_1} * \eta_{t_{ij}}^{\beta_2}\right]}{\sum\limits_{\forall j \notin tabu_k} \tau_{ij}^{\alpha} * \left[\eta_{d_{ij}}^{\beta_1} * \eta_{t_{ij}}^{\beta_2}\right]} & si\ j \notin tabu_k \\ 0\ en\ caso\ contrario \end{cases} \qquad \textbf{(4.1)}$$

donde:

$\eta_{dij} = \frac{1}{d_{ij}}$ es la visibilidad relativa a la distancia d_{ij}.

$\eta_{tij} = \frac{1}{t_{ij}}$ es la visibilidad relativa al tiempo t_{ij}.

Los valores de α, β_1 y β_2 son parámetros del algoritmo que se definen a priori, según sea la importancia relativa de los diferentes argumentos considerados. El valor de $\alpha = 1$ es utilizado por defecto. Los valores de β están comprendidos entre 0 y 1 dependiendo de la influencia que se quiera atribuir a una visibilidad especifica.

Las fórmulas de actualización de feromonas del MMAS original no sufren alteración alguna y se propone utilizar la función objetivo con mayor relevancia, siguiendo los delineamientos propuestos por Barán et al. en [8]. Los valores máximos y mínimos que pueden tomar las variables τ_{ij} están dados por [9]:

$$\tau^{max}{}_{ij} = \frac{1}{1-p} \cdot \frac{1}{F_3(mejor\_solucion\_parcial)} \qquad \textbf{(4.2)}$$

$$\tau^{min}{}_{ij} = \frac{\tau^{max}\,(1-\sqrt[n]{pBest})}{(avg-1)\sqrt[n]{pBest}} \qquad \textbf{(4.3)}$$

donde p es un factor entre 0 y 1 que representa la *tasa de evaporación de la feromona* y F_3*(mejor_solucion_parcial)* representa al mejor tiempo total encontrado hasta la iteración actual. Minimizar el tiempo total del recorrido es el objetivo considerado de mayor relevancia en este trabajo, aunque la formulación general no cambie al considerar algún otro objetivo, como la ganancia de la entrega. La variable *avg* se obtiene como $avg = n / 2$, donde n es la cantidad de nodos y *pBest* es una constante.

Siguiendo los delineamientos tradicionales utilizados en la mayoría de los trabajos referenciados [8 - 11], la actualización de feromonas se realiza utilizando las ecuaciones:

$$\Delta\tau_{ij} = \frac{1}{F_3(\psi)} \qquad \text{si el arco } (i, j) \text{ es utilizado en la solución } \psi \tag{4.4}$$

$$\tau_{ij} = p.\tau_{ij} + \Delta\tau_{ij} \qquad \text{donde:} \quad \tau^{min}{}_{ij} \leq \tau_{ij} \leq \tau^{max}{}_{ij} \tag{4.5}$$

Para obtener el valor de $\Delta\tau_{ij}$ en **(4.4),** se utiliza en este trabajo el valor de la función objetivo $F_3(\psi)$, considerada como la de mayor relevancia. El valor de la tasa de evaporación p es un parámetro de entrada del algoritmo (típicamente cercano al valor 0.9) y τ_{ij} representa la cantidad de feromonas en el arco (i, j).

Pseudocódigo del Algoritmo Propuesto

```
Inicialización de Parámetros
s=0;
Desde nc = 1 hasta Número_de_Iteraciones
    Desde v=1 hasta m              /* cantidad de vehículos disponibles */
        s=s+1;
        Desde t=1 hasta n-1        /* recorrer grafo */
            Desde i=0 hasta n-1
                Desde k=1 hasta v      /*por cada vehículo*/
                    Elegir la ciudad j a mover, con probabilidad p_ij dada por (4.1)
                    Insertar la ciudad j en la lista tabu_k(s)
                Fin Desde k
            Fin Desde i
        Fin Desde t
        Calcular funciones objetivo F(Ψ)
        Guardar soluciones válidas en el Conjunto_de_Soluciones
        Actualizar matriz de feromonas utilizando (4.5)
        Vaciar todas las listas tabú
        s = 0;
    Fin Desde v
    Borrar soluciones dominadas dentro del Conjunto_de_Soluciones
    Actualizar Conjunto_Pareto
Fin Desde nc
```

A continuación se presenta la función utilizada para marcar y eliminar las soluciones dominadas.

Entrada.

Conjunto de soluciones $\{ \psi, \ldots \psi_{n\_sol} \}$.

Pseudocódigo.

Desde *h=1* ***hasta*** *n_sol* /*n_sol es la cantidad de soluciones encontradas*/
    Desde *h'=1* ***hasta*** *n_sol*
        Si $((F_1(\psi_h) \leq F_1(\psi_{h'}))$ ***Y*** $(F_2(\psi_h) \leq F_2(\psi_{h'}))$ ***Y*** $(F_3(\psi_h) \leq F_3(\psi_{h'}))$ ***Y*** $(F_4(\psi_h) \geq F_4(\psi_{h'}))$ ***Entonces***
            Si $((F_1(\psi_h) < F_1(\psi_{h'}))$ ***O*** $(F_2(\psi_h) < F_2(\psi_{h'}))$ ***O*** $(F_3(\psi_h) < F_3(\psi_{h'}))$ ***O*** $(F_4(\psi_h) > F_4(\psi_{h'}))$ ***Entonces***
                Marcar solución $\psi_{h'}$ *como dominada*
            Fin Si
        Fin Si
    Fin Desde *h'*
Fin Desde *h*
Eliminar soluciones dominadas

Salida.

Conjunto Pareto conteniendo las soluciones no dominadas.

5 Resultados Experimentales

Para validar las soluciones obtenidas por el algoritmo propuesto se ha desarrollado un algoritmo de búsqueda exhaustiva (BE) que se encargue de explorar íntegramente todo el espacio de soluciones hasta encontrar las mejores [14]. Este algoritmo BE implementado puede no ser el más eficiente debido a su alta complejidad computacional, pero ha sido útil para garantizar que las soluciones encontradas por el algoritmo propuesto en efecto pertenecen al *Conjunto Pareto*. En las Tablas 4, 5 y 6 se muestran los resultados comparativos entre el algoritmo propuesto y un algoritmo BE, incluyendo información relativa al tiempo de ejecución y las iteraciones realizadas por cada algoritmo, considerando 3 escenario distintos, todos correspondientes a la ciudad de Encarnación. Para estimar el tiempo y la distancia se ha utilizado el *API* de *Google Maps*.

Cada algoritmo ha sido implementado en una aplicación web construida con el lenguaje de programación PHP, utilizando Zend Framework. La aplicación ha sido alojada en un Servidor Web Apache/2.2.26 (Unix) DAV/2 PHP/5.4.30 y ejecutada en un ordenador con Sistema Operativo OS X 10.9, con memoria de 4 GB 1600 MHz DDR3, procesador de 1.3 GHz Intel Core i5 y disco duro SSD.

Para el primer escenario se utilizaron los siguientes parámetros: α, β_1 y $\beta_2 = 1$; n = 4 nodos; *NCMAX* = 10 iteraciones; m = 1 vehículos; W = 480 minutos; M = 350 Gs; G = 2.000 Gs; Cnafta = 0.5 Gs; Cjornal = 1500 gs y CostoFijo = 0.3 Gs., destacándose que al momento de redactar este documento un dólar americano equivaldría a aproximadamente cinco mil guaraníes (Gs).

En la Tabla 1 se puede observar que la Solución 3 es la de menor tiempo total de recorrido, pero al mismo tiempo es la de mayor distancia total y además, la ganancia es menor a las Soluciones 1 y 2 (ambas con mayor tiempo pero menor distancia total). Las Soluciones 1 y 2 tampoco pueden ser mejoradas entre si, considerando que la primera utiliza un menor tiempo pero la distancia es mayor. Con estos resultados se demuestra que las tres Soluciones son *no-dominadas* entre sí, teniendo en cuenta que ninguna puede mejorar a la otra al considerar los 4 objetivos al mismo tiempo.

Para el segundo escenario los parámetros fueron configurados de la siguiente manera: α, β_1 y β_2 = 1; n = 9 nodos; $NCMAX$ = 2.500 iteraciones; m = 2 vehículos; W = 480 minutos; M = 350 Gs; G = 2.000 Gs; Cnafta = 0.5 Gs; Cjornal = 1500 Gs y CostoFijo = 0.3 Gs. En este caso, la Tabla 2 muestra dos soluciones contenidas en el Conjunto Pareto. Es importante destacar que para este escenario se utilizan 2 vehículos. Como el tiempo total del recorrido es menor al límite máximo de trabajo diario permitido, el algoritmo utilizó solo un vehículo para encontrar la solución mostrada, adecuándose al objetivo de utilizar la mínima cantidad de vehículos que puedan ser necesarios. Ambas soluciones son válidas pues ninguna puede mejorar a la otra, es decir, se trata de soluciones *no-comparables* entre sí.

Finalmente, para el tercer escenario se utilizaron los siguientes parámetros: α, β_1 y β_2 = 1; n= 16 nodos; $NCMAX$ = 30.000 iteraciones; m = 2 vehículos; W = 100 minutos; M = 350 Gs; G = 2.000 Gs; Cnafta = 0.5 Gs; Cjornal = 1500 Gs y CostoFijo = 0.3 Gs. El algoritmo propuesto encontró una única solución y fueron necesarios utilizar todos los vehículos disponibles. El tiempo máximo diario permitido por cada vehículo fue de solo 100 minutos, por lo que no se encontró ninguna solución con **un solo vehículo** que realice la entrega en un tiempo menor al establecido, lo que llevó a utilizar ambos vehículos. Los valores obtenidos se presentan en la Tabla 3.

Tabla 1. Resultados del primer escenario. Conjunto Pareto de soluciones.

Solución Nro.	Vehículos	Tiempo	Distancia	Ganancia	¿Dominada?
Solución 1	1	33,38 min	16,76 km	1.491,32 Gs	No
Solución 2	1	34,13 min	16,54 km	1.491,43 Gs	No
Solución 3	1	33,12 min	17,49 km	1.490,96 Gs	No

Tabla 2. Resultados del segundo escenario. Conjunto Pareto de soluciones.

Solución Nro.	Vehículos	Tiempo	Distancia	Ganancia	¿Dominada?
Solución 1	1	97,36 min	67,73 km	3.965,85 Gs	No
Solución 2	1	94,90 min	68,20 km	3.965,60 Gs	No

Tabla 3. Resultados del tercer escenario. Conjunto Pareto de soluciones.

Solución Nro.	Vehículos	Tiempo	Distancia	Ganancia	¿Dominada?
Solución 1	2	150,81 min	103,48 km	7.447,66 Gs	No

El algoritmo de BE fue ejecutado con los parámetros correspondientes a cada escenario de prueba al igual que el algoritmo propuesto en este trabajo y. Los resultados se resumen a continuación:

Tabla 4. Resultados comparativos. Primer escenario.

Algoritmos	Iteraciones	Tiempo	¿Mismas Soluciones?
Algoritmo Propuesto	8	~1 seg.	Si
Búsqueda Exhaustiva	10	~1.5 seg.	Si

Tabla 5. Resultados comparativos. Segundo escenario.

Algoritmos	Iteraciones	Tiempo	¿Mismas Soluciones?
Algoritmo Propuesto	2.500	~2 min.	Si
Búsqueda Exhaustiva	10.000	~30 min.	Si

Tabla 6. Resultados comparativos. Tercer escenario.

Algoritmos	Iteraciones	Tiempo	¿Mismas Soluciones?
Algoritmo Propuesto	30.000	~5.5 min.	Si
Búsqueda Exhaustiva	1.4E+12	~3 hs. 15 min.	Si

En el primer escenario, la diferencia del tiempo empleado y la cantidad de iteraciones realizadas por cada método, no es significativa, dada la simplicidad del problema, utilizado principalmente para validar los resultados. Lógicamente, la diferencia se va haciendo significativa en la medida que el número de nodos *n* aumenta. Así, en el segundo escenario la diferencia entre ambos algoritmos es bastante significativa. El Algoritmo Propuesto precisó 2.500 iteraciones para encontrar una solución factible, mientras que el algoritmo de BE realizó 10.000 iteraciones, con una considerable diferencia de tiempo, dado que el Algoritmo propuesto se ejecutó durante unos 2 minutos mientras el algoritmo BE requirió aproximadamente media hora. Lógicamente, en el tercer escenario la diferencia entre los algoritmos es todavía más amplia, llegando al orden del billón de iteraciones, lo que implicó como 3 horas de diferencia en el tiempo de cálculo.

6 Conclusión

El problema de distribución de extractos bancarios es un caso real de una empresa de la ciudad de Encarnación - Paraguay, que pudo ser modelado matemáticamente utilizando el MTSP. Para resolver este problema en un contexto multi-objetivo, se implementó una variante de la meta-heurística de Colonia de Hormigas (ACO) capaz de resolver el problema en un contexto multi-objetivo que considera la optimización simultánea de cuatro funciones objetivo. Para probar el algoritmo propuesto, fueron considerados tres escenarios de prueba. Resultados experimentales comparados con una búsqueda exhaustiva (BE) demostraron que el algoritmo propuesto funciona correctamente y que lógicamente resulta mucho más eficiente que la BE de referencia, claramente no viable para problemas con un gran número de clientes.

El algoritmo propuesto demostró estabilidad al momento de encontrar una solución Pareto óptima. Esto puede deducirse al observar los resultados del segundo y tercer escenario, donde precisó un porcentaje bastante reducido de la cantidad total de iteraciones utilizadas por el algoritmo de BE.

Como trabajos futuros se propone: replantear el problema teniendo en cuenta la capacidad de cada vehículo y el tamaño de las mercaderías; considerar problemas de mayor tamaño; implementar otras meta-heurísticas además de ACO y comparar los resultados; así como considerar la posibilidad de utilizar más de un depósito al que tengan que regresar los conductores una vez concluido el recorrido diario.

Referencias

1. Davendra, D: Traveling Salesman Problem, Theory and Applications. inTech, Rijeka (2010)
2. Toth, P., Vigo D.: The Vehicle Routing Problem. Society for Industrial and Applied Mathematics, Philadelphia (2001)
3. Gómez O., Barán B.: Reasons of ACO's Success in TSP. Ant Colony Optimzation and Swarm Intell SE - 20, 226-237 (2004)
4. Oberlin P., Rathinam S., Darbha S.: A Transformation for a Multiple Depot, Multiple Traveling Salesman Problem. Am. Control Conf. Hyatt Regency Riverfront, 2636-2641 (2009)
5. Dorigo M., Maniezzo V., Colorni A.: Positive Feedback as a Search Strategy. Dipartimento di Elettronica - Politecnico di Milano (1991)
6. Dorigo M.: The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances. International Series in Operations Research and Management Science, 1-42 (2001)
7. Hermosilla A., Barán B.: Comparación de un sistema de colonias de hormigas y una estrategia evolutiva para un Problema Multiobjetivo de Ruteo de Vehículos con Ventanas de Tiempo. 30ma Conf. Latinoam. Informática, 379-388 (2004)
8. Barán B., Laufer M., Insaurralde M. G.: Generalización del algoritmo MOACS para Varios Objetivos: Una aplicación a la distribución de motocicletas. XL Conf. Latinoam. Informática - CLEI (2014)
9. Stützle T., Hoos H. H.: MAX – MIN Ant System. Future Generation Computer Systems, 889-914 (2000)
10. Rodríguez García J.: Análisis de algoritmos basados en colonia de hormigas en problemas de camino mínimo. Universidad Carlos III De Madrid (2010)
11. Pinto D., Barán B.: Multiobjective Max-Min Ant System: An application to Multicast Traffic Engineering. 7th Argentine Symp. Artif. Intell. – ASAI (2005)
12. Barán B., Schaerer M.: A MultiObjective Ant Colony System for Vehicle Routing Problem With Time Windows. 21st IASTED Int. Conf. Appl. Informatics (2003)
13. Van Veldhuizen D.: Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations. IRE Trans. Educ., vol. 5, n.º 2 (1999)
14. Carbonell A., Yepes V., González Vidosa, F.: Búsqueda exhaustiva por entornos aplicada al diseño económico de bóvedas de hormigón armado. Rev. Int. Métodos Numéricos para Cálculo y Diseño en Ing., vol. 27, n.º 3, 227-235 (2011)
15. Ormel, C. W., Spaans M.: Monte Carlo Simulation of Particle Interactions at High Dynamic Range: Advancing beyond the Googol. Astrophys. J., vol. 684, n.º 2, 1291 (2008)