

Optimization based on Multi-type Ants for the Traveling Salesman Problem

Y.J. Costa Salas*, N. J. Castaño Pérez[^], J. F. Betancur*

Abstract—This paper proposes an algorithm based on multi-type ants (so-called Multi-type Ant Colony System, M-ACS) for solving the Traveling Salesman Problem (TSP). Multiple ant types cooperate (through pheromone exchange) and compete (by mean of repulsion mechanism) among them in order to increase the efficacy in the search process. The obtained experimental results has been compared against benchmark results from OR literature. In particular for large scale TSP, the algorithmic proposal M-ACS shows competitive results regarding to the efficacy.

Keywords—optimization, bioinspired computation, ant algorithms, multi-type ants, traveling salesman

I. INTRODUCCIÓN

Indudablemente, el problema del agente viajero o vendedor (*Traveling Salesman Problem*, TSP), es uno de los problemas combinatorios más estudiados en el área de la Investigación de Operaciones [1]. Múltiples artículos han abordado estrategias de solución para este problema [2]. El TSP constituye el punto de partida, en términos de modelación matemática, para varias decisiones de planeación en la industria y los servicios [3] y [4]. En el sentido de lo anterior pueden citarse como ejemplos fehacientes: la secuenciación de tareas (*Scheduling Problems*) [5], asignación de recursos (*Resource Assignments*) [6], planeación de rutas (*Route Planning*) en diversos contexto de aplicación [7-9], entre otros.

El TSP, como problema combinatorio discreto, consiste en encontrar una ruta hamiltoniana (ruta que inicia y culmina en el mismo nodo origen) para el vendedor de forma tal que todos los nodos o punto destinos, asignados a dicho vendedor, sean visitados una sola vez [10].

Distintos algoritmos de solución han sido propuestos para resolver el TSP, dentro de los que se encuentran dos grandes familias de algoritmos: los exactos y los aproximados [11]. La lista de algoritmos, provenientes de ambas familias, implica más de 50 años de estudios en este campo por parte de la comunidad científica. Por tales

motivos, en la Tabla I se muestra un resumen de las alternativas de solución más conocidas para el TSP.

Una mirada crítica a la Tabla I permite apreciar como las meta-heurísticas presentan mayor auge en los últimos tiempos. Específicamente, aquellas que se inspiran en el comportamiento de los seres vivos. Como resultado de la aplicación de este tipo de algoritmos, las mejores soluciones encontradas para TSP se han acercado significativamente al óptimo absoluto, resultando bien interesante, la evidente reducción del tiempo de ejecución que presentan las meta-heurísticas.

TABLA I
ENFOQUES ALGORÍTMICOS PARA EL TSP

Enfoque	Tipo de Algoritmo	Contexto de Aplicación	Autores
Exacto	<i>Branch and Bound</i> <i>Branch and Cut</i>	Bioinformática	[12]
	<i>Constraint Programming</i>	TSP con ventanas de tiempo	[13]
	<i>Branch and Cut</i>	Problemas de carga y descarga	[14]
Aproximado	<i>Lin-Kernighan heuristic</i>	TSP generalizado	[15]
	Genetic Algorithm	TSP multiobjetivo y bajo incertidumbre	[16]
	Particle Swarm Optimization	TSP bajo competencia	[17]
	Genetic Algorithm Ant Algorithm	TSP clásico	[18]
	Ant Algorithm	TSP dinámico con factor de tráfico	[19]

El presente artículo propone una variante algorítmica basada en el comportamiento real de las hormigas: el Sistema Multi-tipos de Colonias de Hormigas (*Multi-type Ant Colony System*, M-ACS). El M-ACS introduce la utilización de múltiples tipos de hormigas asociadas en colonias, donde cada una construye una solución global de problema y durante el proceso constructivo las hormigas cooperan, mediante el intercambio “frecuente” de feromona, además de competir a través de un mecanismo de repulsión. Dicha variante puede ser encontrada en trabajos previos, con aplicación particular al problema de los múltiples agentes vendedores (*Multiple Traveling Salesman Problem*, mTSP) [20]. Como elemento distintivo, la presente investigación propone el análisis del desempeño de la propuesta algorítmica (M-ACS) para el TSP de gran escala. En este sentido, la eficacia del M-ACS es comparada con otros enfoques reconocidos en la literatura especialidad, usando como bases diversos problemas de

*Y. J. Costa Salas is with the Department of Economics, Universidad de Manizales, Manizales, Road 9 #19-03 Colombia (email: yasel.costa@umanizales.edu.co).

[^]N. J. Castaño Pérez is with the Department of Engineering, Universidad de Manizales, Manizales, Road 9 #19-03 Colombia (email: njcastanop@gmail.com).

*J. F. Betancur is with the Department of Economics, Universidad de Manizales, Manizales, Road 9 #19-03 Colombia (email: jbetancur@umanizales.edu.co).

gran escala, los cuales pueden ser encontrados en *OR-Library* [21].

Los resultados experimentales permitieron demostrar que el enfoque propuesto es competitivo, y en algunos casos, se desempeñó mejor en términos de eficacia (valor de la función objetivo).

Para su presentación, este artículo se encuentra estructurado de la forma siguiente: en la Sección II se presenta, de manera resumida, la formulación matemática del problema objeto de estudio (el TSP), luego en la Sección III se describe el enfoque algorítmico propuesto (M-ACS) para el TSP. La Sección IV del presente artículo está dedicada a mostrar algunas experiencias computacionales a través de la aplicación del algoritmo propuesto en instancias de gran escala del TSP. Finalmente, las conclusiones de la investigación aparecen en la Sección V.

II. FORMULACIÓN MATEMÁTICA DEL TSP

El TSP puede definirse en un grafo $G = (V, A)$, donde V resulta un conjunto de n nodos y A representa el conjunto de arcos que conectan dichos nodos. Asociado a cada arco del grafo existe una matriz $C = (c_{ij})$, donde c_{ij} indica mayormente la longitud del arco en unidades de distancia o tiempo. Se define a la matriz C como simétrica [21] cuando $c_{ij} = c_{ji}, \forall (i, j) \in A$ y asimétrica para el caso contrario. El objetivo del problema consiste en encontrar un camino hamiltoniano (comienza y culmina en el nodo origen, nodo 1 en la formulación), el cual debe realizarse visitando los nodos una sola vez.

La variable de decisión asociada al modelo matemático del TSP queda definida de la forma siguiente:

$$x_{ij} = \begin{cases} 1 & \text{si el arco}(i, j) \text{ es usado en una ruta,} \\ 0 & \text{de lo contrario.} \end{cases} \quad (1)$$

De manera general la formulación lineal entera del TSP puede ser establecida como sigue:

$$\text{minimizar } \sum_{i=1}^n \sum_{j=1}^n c_{ij} \cdot x_{ij} \quad (2)$$

$$\sum_{i=1}^n x_{ij} = 1, \quad j = 2, \dots, n, \quad (3)$$

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 2, \dots, n, \quad (4)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, \quad \forall S \subseteq V \setminus \{1\}, S \neq \emptyset, \quad (5)$$

$$x_{ij} \in \{0,1\}, \forall (i, j) \in A \quad (6)$$

La expresión (2) describe la función objetivo del problema, la cual consiste en minimizar la sumatoria de las longitudes asociadas a cada arco (i, j) . Las restricciones (3) y (4) aseguran que el vendedor visite sus correspondientes nodos solo una vez. Finalmente, las restricciones que se generan con la expresión (5) previenen la creación de subcircuitos en el grafo.

III. EL ALGORITMO M-ACS PARA RESOLVER EL TSP

Partiendo la propuesta inicial del M-ACS reportada en [20], el enfoque algoritmo está basado en la siguiente idea: sea CO un conjunto de colonias de hormigas artificiales, las cuales representan posibles soluciones globales para el problema formalizado en la Sección II (ver Fig. 1). Cada hormiga perteneciente a la colonia construye una solución global mediante la aplicación del algoritmo Sistema de Colonias de Hormigas (*Ant Colony System, ACS*) [22], pero durante el proceso de construcción de la solución, las hormigas de un mismo tipo cooperan compartiendo entre ellas información sobre las experiencias alcanzadas hasta una iteración determinada. Dicha cooperación se realiza mediante “frecuentes” intercambios de feromona. Simultáneamente, las hormigas de tipos diferentes establecen un mecanismo de repulsión, el cual fomenta la competencia entre hormigas (de tipos diferentes) por incorporar mejores nodos a la solución global. De este modo un conjunto de soluciones es construida por las colonias que sean definidas, seleccionado la mejor solución posteriormente a la última iteración.

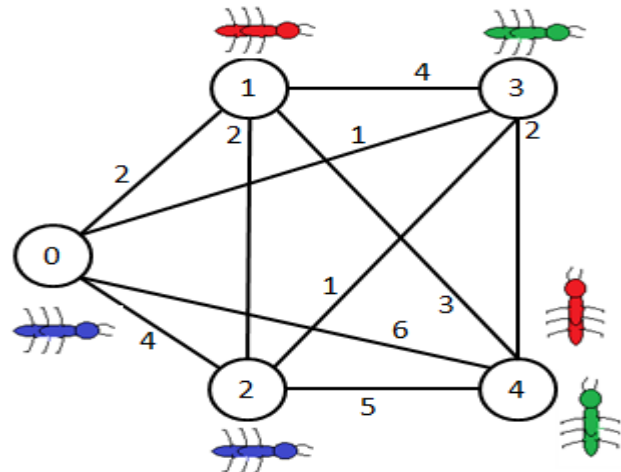


Fig. 1 Construcción de las soluciones en el M-ACS.

A. CONSTRUCCIÓN DE LA RUTA

Las hormigas artificiales construyen una ruta mediante la selección sucesiva pseudo-aleatoria e inteligente hasta que todos los nodos hayan sido visitados. Los nodos deben ser visitados una sola vez. En el algoritmo propuesto, la selección de un nodo no visitado por el viajero se realiza considerando tres aspectos: qué tan buena fue la elección de dicha ciudad hasta la interacción actual (τ_{rs} , rastro de feromona), qué tan promisorio resulta la ciudad analizada (η_{rs} , medida de deseabilidad) y qué tan buena fueron las elecciones de dicha ciudad para las hormigas que

pertenecientes a las colonias de tipos diferentes ($\phi_a(r, s)$, rastro de feromona de la colonia que integran las hormigas distintas al tipo a). A diferencia del clásico ACS, en el M-ACS, cada hormiga se mueve desde un nodo actual r hasta el siguiente nodo s mediante la regla pseudo-aleatoria siguiente:

$$s = \begin{cases} \arg \max_{u \in J_k(r)} \{ \text{EXPT} \} & \text{if } q \leq q_0 \\ S & \text{otherwise} \end{cases} \quad (7)$$

$$\text{EXPT} = \{ \tau(r, u) \cdot [\eta(r, u)]^\beta \cdot [\phi_a(r, s)]^{-\gamma} \}$$

$$S: p_k(s) = \begin{cases} \{ \text{EXPR} \} & \text{if } s \in J_k(s) \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

$$\text{EXPR} = \frac{[\tau(r, s)] \cdot [\eta(r, s)]^\beta \cdot [\phi_a(r, s)]^{-\gamma}}{\sum_{u \in J_k(s)} [\tau(r, u)] \cdot [\eta(r, u)]^\beta \cdot [\phi_a(r, u)]^{-\gamma}}$$

donde $\phi_a(r, s)$, para la presente propuesta algorítmica, indica el valor promedio de feromona en el arco r , alcanzado por todas las hormigas distintas del tipo a . El parámetro γ es incorporado al algoritmo, dicho parámetro expresa la tendencia existente en cada colonia hacia el uso de su propia experiencia ($\gamma = 0$) o hacia el uso de las experiencia de las colonias con hormigas de otro tipo ($\gamma > 0$). El término $J_k(s)$ indica el conjunto de nodos que faltan por visitar a la hormiga k estando en el nodo s . Los parámetros α y β determinan la influencia relativa de la feromona versus la distancia ($\alpha, \beta > 0$). En el algoritmo se define además el parámetro q_0 , el cual se distribuye uniformemente $[0, 1]$ y determina la importancia relativa de la explotación (expresión 7) versus la exploración (expresión 8). El número aleatorio q se genera uniformemente $[0, 1]$, y en caso de que $q \leq q_0$ el próximo nodo se incorpora mediante la expresión (8), de lo contrario mediante la expresión (9).

La medida de deseabilidad $\eta(r, s)$, específicamente para el TSP, puede ser calculada según la expresión siguiente:

$$\eta_{rs} = \frac{1}{d_{rs}} \quad (9)$$

donde d_{rs} representa la distancia que existe entre el nodo r y el nodo s (nodo r y s son equivalentes a los nodos i y j en las formulación matemática de la Sección II).

B. ACTUALIZACIÓN DE LA FEROMONA

El proceso de actualización de la feromona para el M-ACS se desarrolla bajo las mismas reglas que el algoritmo clásico ‘‘Sistema de Colonias de Hormigas’’ (*Ant Colony System*, ACS)[38]. La gran diferencia del algoritmo propuesto radica en la creación de una matriz de feromona por tipo de hormiga. Esto significa que las reglas de actualización, tanto la local, como la global, modifican los valores de las matrices según el tipo de hormiga.

La regla de actualización local (ver expresión 10) se aplica para cambiar el nivel de feromona de cada arco mientras cada hormiga artificial construye una solución.

$$\tau_{rs}^{new} = (1 - \rho)\tau_{rs}^{old} + \rho\tau_0 \quad (10)$$

donde ρ se define como el coeficiente de evaporación ($0 \leq \rho \leq 1$) y $(1 - \rho)$ indica el nivel de evaporación de la feromona. El nivel inicial de feromona (τ_0) es alcanzado a través de la clásica heurística del Vecino más Cercano (*Nearest Neighbor*, NN) y se determina mediante la siguiente expresión:

$$\tau_0 = (n \cdot L_{nn})^{-1} \quad (11)$$

donde n denota el número de ciudades y L_{nn} representa el recorrido total aplicando la heurística NN. Posterior a la primera iteración, se realiza la actualización global de la feromona (expresión 12) para aquellos arcos que conformaban la mejor solución alcanzada en la primera iteración. Es decir, cada colonia representa un tipo o clase particular de hormigas, y cada hormiga construye una solución. Por lo tanto, los arcos que forman parte de la mejor solución encontrada, por tipo de hormiga, reciben una cantidad adicional de feromona (actualización global se aplica para cada matriz de feromona, según los tipos de hormigas que existan), lográndose de este modo que, para la iteración siguiente, exista mayor probabilidad de que se preserve la mejor solución por cada colonia.

$$\tau_{rs}^{new} = (1 - \rho)\tau_{rs}^{old} + \frac{\rho}{L_{Best(a)}} \quad \forall (r, s) \in BestSol \quad (12)$$

donde $L_{best(a)}$ es la distancia total recorrida de la mejor solución (*BestSol*) en la iteración, para la colonia tipo a .

III. C. COOPERACIÓN Y COMPETENCIA ENTRE HORMIGAS

El algoritmo M-ACS (ver en Pseudo-códigos 1 y 2), como enfoque perteneciente a la inteligencia colectiva (*Swarm Intelligence*), presenta un mecanismo de cooperación. El proceso de cooperación, inspirado en [23], consiste en los ‘‘frecuentes’’ intercambios del rastro de feromona alzado por las hormigas que pertenecen a cada colonia. Cada colonia tiene su propia matriz de feromona. La idea de intercambio es compartir la experiencia entre colonias (entre hormigas de diferentes tipos). Los ‘‘frecuentes’’ intercambios de feromona son realizados después de una cantidad de iteraciones predefinidas F . Dicho parámetro puede definirse porcentualmente, indicando que cuando el número de iteraciones haya alcanzado el valor porcentual F con respecto al total de iteraciones N se debe realizar el intercambio de feromona entre las colonias. De este modo, los nuevos valores de feromonas, para cada arco de las matrices asociadas a los tipos de colonia, pueden ser calculados de la manera siguiente:

$$\phi_a(r, s) = \frac{\sum_{\forall c \in CO; c \neq a} \phi_c(r, s)}{CO - 1} \quad (13)$$

donde a representa la colonia actual que recibe el valor del intercambio, la cual toma el valor promedio de los rastros de feromona de las otras colonia, excluyendo la colonia actual (se emplea el índice temporal c para indicar todos los otros tipos de colonias exceptuando la colonia tipo a).

Pseudo-código 1: procedimiento general del M-ACS

```

Inicializar parámetros
Obtener la solución inicial mediante NN ( $\psi^m$ )
 $\psi^{gb} \leftarrow \psi^m$ 
 $L_{gb} \leftarrow L_m$ 
For cada colonia  $a$ 
  For cada arco  $(r, s)$ 
     $\tau_{rs} = \tau_0$  (Ecuación 11)
  EndFor
Do Until  $IT = N$ 
  If  $IT \% N = F$  Then
    Intercambiar rastros de feromona entre colonias
    (Ecuación 13)
  End If
For cada colonia  $a$ 
  For cada hormiga  $k$ 
    Construir una solución ( $\psi^k$ ) mediante (solución-
    nueva-hormiga)
    If  $L_k \leq L_{gb}$  Then
       $L_{gb} = L_k$ ;  $\psi^{gb} = \psi^k$ 
    EndIf
  EndFor
EndFor
 $IT = IT + 1$ 
Loop
For cada arco  $(r, s)$  de la colonia  $a$  en solución  $BestSol$ 
  Actualizar rastro de feromona  $\tau_{rs}$  (Ecuación 12)
EndFor

```

El procedimiento **solución-nueva-hormiga** se describe como sigue en el pseudo-código siguiente:

Pseudo-código 2: solución-nueva-hormiga

```

Inicializar parámetros
Localizar hormiga  $k$  en nodo origen (random en TSP)
Inicializar la distancia recorrida:  $L^k \leftarrow 0$ 
For cada colonia  $a \in CO$ 
  While (Hormiga  $k$  no haya completado su
  solución)
    Seleccionar el próximo nodo  $s$  mediante ecuación
    (7) or (8)
    Actualizar el rastro de feromona  $\tau_{rs}$  (Ecuación 10)
    Actualizar recorrido:  $\psi^k \leftarrow \psi^k + \langle s \rangle$ 
    Actualizar distancia recorrida:  $L^k \leftarrow L^k + d_{rs}$ 
  EndWhile
EndFor

```

El mecanismo de repulsión o competencia, entre hormigas de diferentes tipos, se deriva del término

$[\phi_a(r, s)]^{-\gamma}$ en las expresiones (7) y (8). La hormiga tipo que pertenece a la colonia a (de todos los tipos que incluyen el conjunto CO), tiene menos probabilidad de incorporar el nodo s a camino si por ese arco (r, s) pasaron ya hormigas de otro(s) tipo(s). Es decir, si otros tipos de hormigas recorrieron ese arco, el término $\phi_a(r, s)$ (promedio de la feromona usando los valores de las matrices excluyendo el la hormiga tipo a) se hace mayor y por ende habrá menos probabilidad de que la hormiga adicione el nodo s al camino construido.

IV. RESULTADOS EXPERIMENTALES

El desempeño del M-ACS y su comparación con otros resultados tomados de la literatura se presentan en esta sección. Para la implementación y ejecución del algoritmo fue empleado un portátil (*laptop*) con las características siguientes:

- Procesador Intel Core i5 con 2.6GHz.
- Memoria *Ram* de 4GB.
- Implementación de los pseudocódigos en C#.

Para medir la eficacia del M-ACS fueron utilizados, como resultados de referencia, los valores reportados en [24]. La comparación tuvo como base los problemas de clásicos de TSP de gran escala (ver en [21]), específicamente con una cantidad de nodos superior a los 200.

En la ejecución del algoritmo M-ACS se emplearon la configuración de parámetros como sigue: $q_0 = 0.8$, $\alpha = \beta = \gamma = 1$, $\rho = 0.1$, solamente 10 hormigas para cada tipo de colonia. Como resultado de un estudio estadístico previo de Análisis de Varianza (*Analysis of Variance*, ANOVA) se establecieron 3 tipos colonias para la solución de las instancias, así como realizar el intercambio de feromona cada 10 iteraciones (10% de un total de 100 iteraciones).

Los resultados computacionales obtenidos se aprecian en la Tabla II. Donde la primera columna indica el problema TSP con su respectivo código tomado de *OR-Library*. Luego aparece la dimensión o cantidad de nodos que presenta cada instancia en la columna 2. Las columnas 3, 4 y 5 representan, respectivamente, el mejor valor obtenido (según la literatura especializada) (BK) en la función objetivo para la instancia, el mejor valor obtenido por [24] con el llamado Algoritmo de Hormigas en Dos Pasos (*Two Step Ant Colony System*, TS-ACS) y la eficacia del M-ACS. Las últimas dos columnas están dedicada a presentar las diferencias (Δ) entre la eficacia de los enfoques. El símbolo “+” indica una diferencia en el sentido favorable de la función objetivo (mínimo para el TSP), mientras que símbolo “-” indica lo contrario.

TABLA II
RESULTADOS DE EFICACIA PARA LAS INSTANCIAS DEL TSP

Problema	BK	TS-ACS	M-ACS	(Δ) BK vs M-ACS	(Δ) TS-ACS vs M-ACS
kroA200	29368	32775,2	31982,1	-2614,1	+793,1
tsp225	3919	4032,44	4037	-118	-4,56

a280	2579	2948,44	2931,61	-352,61	+16,83
lin318	42029	47365,16	45516,8	-3487,8	+1848,36
pcb442	50778	58190,92	58196,12	-7418,12	-5,2
rat575	6773	7034,43	7031,36	-258,36	+3,07
rat783	8806	10033,16	10012,11	-1206,11	+21,05

Los resultados en la tabla anterior describen que el desempeño del M-ACS se comporta bajo las características siguientes:

- En ningún problema se supera la mejor solución obtenida (negativos en toda la columna BK vs M-ACS). No obstante, para algunos problemas, la diferencia es mucho menor respecto al TS-ACS.
- En 4 instancias de las 7 comparaciones el M-ACS presenta mayor eficacia que el enfoque TS-ACS.
- Para los casos en que el algoritmo propuesto es superado, la mejoría puede considerarse como poco significativa.

Como complemento a los resultados descriptivos anteriores se presenta la Tabla III. El nivel de significación (p-value), aplicando la prueba de Wilcoxon, permite inferir que no existen diferencias significativas entre los enfoques basados en colonias de hormigas (TS-ACS vs M-ACS). Este valor proporciona suficiente evidencia para asegurar que el enfoque algorítmico propuesto resulta competitivo frente a otros algoritmos de la misma familia.

TABLA III
RESULTADOS DE LA PRUEBA DE WILCOXON

Comparación	p-value	Hipótesis
TS-ACS vs M-ACS	0.204	aceptada

V. CONCLUSIONES

En la presente investigación, se presenta el M-ACS como una variante de solución bioinspirada. El enfoque algorítmico introduce dos mecanismos para la estrategia de búsqueda: cooperación, la cual posibilita que hormigas del mismo tipo refuercen sus soluciones, y la competencia, donde aparece la repulsión entre hormigas de diferentes tipos introduciendo diversidad en las soluciones. El algoritmo propuesto es aplicado al TSP, problema de optimización de gran relevancia académica y práctica. Los estudios experimentales, descriptivos e inferenciales, permitieron comprobar que el M-ACS es un algoritmo competitivo en términos de eficacia. Un aspecto interesante se evidencia en la mejoría de la eficacia que logra el M-ACS, tomando como valor de referencia, las mejores soluciones encontradas para instancias del TSP en la literatura especializada.

Diversas tareas de investigación pudieran ser desarrolladas en el futuro. La combinación del M-ACS y algoritmos de búsqueda local resulta la acción de mayor interés para futuras contribuciones.

REFERENCIAS

- [1] H. Hoos and T. Stützle, "On the empirical scaling of run-time for finding optimal solutions to the travelling salesman problem", *European Journal of Operational Research*, vol. 238, pp. 87-94, Oct. 2014.
- [2] K. L. Hoffman, M. Padberg and G. Rinaldi, "Traveling Salesman Problem", *Encyclopedia of Operations Research and Management Science*, pp. 1573-1578. 2013.
- [3] K. Wakabayashi, A. Watanabe, J. Toyotani, K. Suzuki, K. Murata and S. Sala-ngam, "A Study on the Optimum Location of the Central Post Office in Bangkok: Applying the Travelling Salesman Problem", *Logistics Operations, Supply Chain Management and Sustainability EcoProduction*, pp. 525-537. 2014.
- [4] A. Baltz, M. El Ouali, G. Jäder, V. Sauerland and A. Srivastav, "Exact and heuristic algorithms for the Travelling Salesman Problem with Multiple Time Windows and Hotel Selection", *International Journal of the Operational Research Society*, vol. 65, pp. 231-245, April. 2014.
- [5] I. Mattik, P. Amorim and H.-O. Günther, "Hierarchical scheduling of continuous casters and hot strip mills in the steel industry: a block planning application", *International Journal of Production Research*, vol. 52, pp. 2576-2591, June. 2014
- [6] D. Gyulai, B. Kádár, A. Kovács and L. Monostori, "Capacity management for assembly systems with dedicated and reconfigurable resources", *CIRP Annals - Manufacturing Technology*, vol. 63, pp. 457-460, January. 2014.
- [7] M. Allahviranloo, J. Y. J. Chow, and W. W. Recker, "Selective vehicle routing problems under uncertainty without recourse", *Transportation Research Part E: Logistics and Transportation Review*, vol. 62, pp. 68-88, February. 2014.
- [8] D. J. Papageorgiou, G. L. Nemhauser, J. Sokol, M.-S. Cheon and A. B. Keha, "MIRPLib – A library of maritime inventory routing problem instances: Survey, core model, and benchmark results", *European Journal of Operational Research*, vol. 235, pp. 350-366, June. 2014.
- [9] J. Michallet, C. Prins, L. Amodeo, F. Yalaoui and G. Vitry, "Multi-start iterated local search for the periodic vehicle routing problem with time windows and time spread constraints on services", *Computers & Operations Research*, vol. 41, pp. 196-207, January. 2014.
- [10] G. Laporte, "A concise guide to the Traveling Salesman Problem", *Journal of the Operational Research Society*, vol. 61, pp. 35-40, March. 2010.
- [11] G. K. D. Saharidis, "Review of Solution Approaches for the Symmetric Traveling Salesman Problem", *Information System and Supply Chain Management*, vol. 7, pp. 1-15, January. 2014.
- [12] A. Fischer, F. Fischer, G. Jäger, J. Keilwagen, P. Molitor and I. Grosse, "Exact algorithms and heuristics for the Quadratic Traveling Salesman Problem with an application in bioinformatics", *Discrete Applied Mathematics*, vol. 166, pp. 97-114, March. 2014.
- [13] G. Pesant, M. Gendreau, J.-Y. Potvin and J.-M. Rousseau, "An Exact Constraint Logic Programming Algorithm for the Traveling Salesman Problem with Time Windows", *Transportation Science*, vol. 32, pp. 12-29, February. 1998.
- [14] M. A. Alba Martínez, J.-F. Cordeau, M. Dell'Amico and M. Iori, "A Branch-and-Cut Algorithm for the Double Traveling Salesman Problem with Multiple Stacks", *INFORMS Journal on Computing*, vol. 25, pp. 41-55, Winter.2013.
- [15] D. Karapetyan and G. Gutin, "Lin-Kernighan heuristic adaptations for the generalized traveling salesman problem", *European Journal of Operational Research*, vol. 208, pp. 221-232, February. 2011.
- [16] C. Changdar, G. S. Mahapatra and R. K. Pal, "An efficient genetic algorithm for multi-objective solid travelling salesman problem under fuzziness", *Swarm and Evolutionary Computation*, vol. 15, pp. 27-37, April. 2014.
- [17] B. Vallade and T. Nakashima, "Improving Particle Swarm Optimization Algorithm and Its Application to Physical Travelling Salesman Problems with a Dynamic Search Space", *Applied Computing and Information Technology*, vol. 553, pp. 105-119, May. 2014.
- [18] G. Dong, W. W. Guo and K. Tickle, "Solving the traveling salesman problem using cooperative genetic ant systems", *Expert Systems with Applications*, vol. 39, pp. 5006-5011, April. 2012.

- [19] M. Mavrovouniotis and S. Yang, "Ant colony optimization with immigrants schemes for the dynamic travelling salesman problem with traffic factors", *Applied Soft Computing*, vol. 13, pp. 4023-4037, October. 2013.
- [20] Y. J. Costa Salas, R. Abreu Ledón, N. I. Coello Machado and A. Nowé, "Sistema multi-tipos de colonias de hormigas para la solución de múltiples agentes vendedores". *Revista Técnica de la Facultad de Ingeniería Universidad del Zulia*, vol. 35, pp. 311-320, Diciembre. 2013.
- [21] The symmetric and asymmetric TSPLIB, Traveling Salesman Problem Instances: <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>.
- [22] M. Dorigo and L.M. Gambardella, "Ant colony system: A cooperative learning approach for the traveling salesman problem", *IEEE Transactions on Evolutionary Computation*, vol. 1, pp. 53-66. 1997.
- [23] A. Nowé, K. Verbeek and P. Vrancx, "Multi-type Ant Colony: The Edge Disjoint Paths Problem". *Paper presented at the Ants 2004*. Brussels, Belgium. 2004.
- [24] A. Puris, R. Bello and F. Herrera, "Analysis of the efficacy of a Two-Stage methodology for ant colony optimization: Case of study with TSP and QAP", *Expert Systems with Applications*, vol. 37, pp. 5443-5453, July. 2010.