

Universidad	Tecnológica de Pereira
Programa Académico	Ingeniería Industrial
Nombre del Semillero	Semillero grupo GAOPE
Nombre del Grupo de Investigación (si aplica)	Grupo de aplicaciones en optimización y procesos estocásticos
Línea de Investigación (si aplica)	Transporte
Nombre del Tutor del Semillero	Eliana Mirledy Toro Ocampo
Email Tutor	elianam@utp.edu.co
Título del Proyecto	Problema del agente viajero, estado del arte y solución a través de heurísticas
Autores del Proyecto	Jhon Alexander Ladino Muñoz, Carlos Armando Rodríguez López, Angiee Rivera Acevedo, Daniela Ospina Toro
Ponente (1)	Jhon Alexander Ladino Muñoz
Documento de Identidad	1088296073
Email	Jhon.ladino@utp.edu.co
Ponente (2)	Carlos Armando Rodríguez López
Documento de Identidad	1088285029
Email	caarrodriguez@utp.edu.co
Teléfonos de Contacto	3105487424-3136640639
Nivel de formación de los estudiantes ponentes (Semestre)	Octavo semestre de Ingeniería Industrial
Área de la investigación (seleccionar una- Marque con una x)	• Ciencias Naturales
	• Ingenierías y Tecnologías (x)
	• Ciencias Médicas y de la Salud.
	• Ciencias Agrícolas
	• Ciencias Sociales
	• Humanidades
	• Artes, arquitectura y diseño

Problema del agente viajero, estado del arte y solución a través de heurísticas

¹Jhon Alexander Ladino Muñoz Carlos Armando Rodriguez López
Angiee Rivera Acevedo Daniela Ospina Toro
Universidad Tecnológica de Pereira

Resumen

El problema del agente viajero es un problema clásico de la investigación de operaciones. Su solución siempre se ha considerado como una gran desafío tanto por su cantidad de aplicaciones como por su complejidad computacional. El objetivo de este documento es presentar una revisión del estado del arte del problema, que permita mapear su importancia a nivel académico y empresarial. Además presentar el software Concorde de uso libre, a partir del cual es posible resolver el problema a través de técnicas exactas y aproximadas con tiempos computacionales muy eficientes considerando el tamaño de los casos de prueba analizados.

Palabras clave: problema del agente viajero, estado del arte, heurísticas constructivas, heurísticas de mejoramiento.

Introducción

El problema del agente viajero, *Travel Salesman Problem* (TSP) es uno de los problemas clásicos de optimización, donde se considera un agente viajero que debe visitar un conjunto de ciudades considerando: *i*) Iniciar y terminar su recorrido en una ciudad; *ii*) Cada ciudad debe visitarse una única vez; *iii*) Se deben visitar todas las ciudades; *iv*) Encontrar una secuencia de visitas óptima que garantice un tour para el cual la distancia total recorrida sea mínima. Se han documentado un sin número de aplicaciones en diferentes tipos de organizaciones tales como: *i*) Planificación del transporte: Planificación de gruas [1], Planificación de rutas escolares[2] Planificación de camiones de entrega[3]. *ii*) Planificación de vehículos aéreos no tripulados [4]. *iii*) Planificación de la producción: Programación de impresión [5], planificación del laminado en caliente [6], Programación de anuncios pre-print [7]. *iv*) Sistemas de satélite, diseño de sistemas de topografía por satélite [8].

1. Planteamiento del problema

En el intento de resolver el problema del agente viajero, muchos investigadores han dedicado sus estudios a plantear un modelo matemático que lo describa totalmente y a partir de él encontrar estrategias de solución eficientes para casos de diferente tamaño. En el análisis de los modelos matemáticos se ha determinado que el problema es NP-duro, lo

¹Ingeniería Industrial – VIII semestre – Estudiante – Jhon.ladino@utp.edu.co,

²Ingeniería Industrial – VIII semestre – Estudiante – caarrodriguez@utp.edu.co,

³Ingeniería Industrial – IX semestre – Estudiante – angie.rivera@utp.edu.co,

⁴Ingeniería Industrial – Docente- dospina@utp.edu.co.

que indica que no es posible determinar una estrategia exacta en tiempo polinomial que permita resolver el en un tiempo computacional aceptable. El espacio de soluciones de este problema es $n!$, donde n es el número de nodos a visitar, por tanto este crece exponencialmente en relación al tamaño del problema. Las heurísticas se presentan como una alternativa interesante en problemas de este tipo debido a que a partir de este tipo de estrategias se encuentran soluciones de muy buena calidad en tiempos computacionales competitivos. En este documento se presentan cuatro heurísticas disponibles en el software concorde, las cuales se aplicaran a diferentes tamaños de problema para evaluar su eficiencia en cuanto a tiempos computacionales y calidad de las respuestas. Además se presentará la forma de interactuar con el software a fin de poder resolver casos generados por el usuario.

Justificación

Como se ha comentado se han documentado gran cantidad de aplicaciones en la solución del problema. El interés de la investigación ha sido la solución del problema de ruteo de vehículos aplicado a la cadena de suministro donde gran cantidad de sus tareas incluye transporte de insumos, productos en proceso y terminados así como atención de diferentes servicios. El costo del transporte afecta directamente el valor de los productos y servicios finales que recibe una comunidad así que resolver el problema de forma eficiente impacta directamente la sostenibilidad económica de las organizaciones. Si se conoce claramente el planteamiento del modelo matemático del problema así como sus estrategias de solución es posible llegar a planteamientos multiobjetivo donde además de minimizar los costos de operación es posible minimizar las emisiones contaminantes en la ejecución de las rutas si se reducen los consumos de combustible.

Objetivos

General

Realizar una revisión del estado del arte del problema del TSP donde se identifique relevancia, aplicaciones, modelos matemáticos, estrategias de solución para instancias de la literatura especializada y para casos que se planteen.

Específicos

i) Identificar las variables de solución que describen el problema, las restricciones que deben cumplirse para obtener una solución factible, con el fin de evaluar la complejidad matemática del mismo. ii) Resolver el TSP mediante técnicas exactas y heurísticas para validar la efectividad de cada una de ellas. iii) Explorar la solución del TSP mediante software que tenga módulos exactos y aproximados a fin de comparar las respuestas obtenidas mediante diferentes heurísticas.

2. Referente teórico

La primera solución reportada para resolver TSP fue en 1954, cuando George Dantzig, Ray Fulkerson, y Selmer Johnson publicaron la descripción de un método de solución del PAV (Problema del Agente Viajero) titulado “*Solutions of a large scale traveling salesman problem*” (Soluciones de gran escala para el problema del agente viajero) para resolver una

instancia de 49 ciudades donde un agente viajero desea visitar un conjunto de ciudades, asignándoles un costo por visitar ciudades contiguas (distancia de traslado entre dos ciudades) [9]. El TSP ocupa un papel crucial en la investigación de operaciones, tanto a nivel teórico como aplicado, siendo dicho problema reconocido por su complejidad de resolución, sencilla formulación y prueba de validación inicial de nuevos avances algorítmicos en la optimización combinatoria [10, 11].

Probar la efectividad de los métodos emergentes en el TSP posibilita luego trascender hacia otros problemas, como ocurrió con la relajación lagrangiana, con el branch and bound [12] y con el modelo inicial de Colonia de Hormigas [13]. Desde 1954 se ha incrementado la efectividad de los métodos desarrollados para resolver problemas TSP cada vez de mayor tamaño. Por ejemplo, Dantzig, Fulkerson y Johnson [14] resolvieron para 49 ciudades, Held y Karp [15] para 64 ciudades, Padberg y Rinaldi [16] aportaron respuesta a una instancia de 2.392 ciudades y en el 2004 ya se conocía respuesta para 24.978 ciudades [4]. El TSP de mayor tamaño resuelto hasta la fecha ha sido para una instancia de 85.900 ciudades, denominada pla 85.900 en el dominio público TSPLIB. En dicha librería también se dispone de problemas de mayores tamaños, con la mejor solución encontrada. Los principales métodos para aportar respuestas al TSP pueden dividirse en: algoritmos heurísticos, de enumeración implícita y metaheurísticas. Entre las heurísticas clásicas para el TSP están las de construcción son: Vecino más Cercano, de inserción (más lejana, más cercana); las de mejoramiento son otra categoría: k-opt, LK y LKH; respecto a métodos de enumeración implícita, están los algoritmos tipo Branch and bound. Entre las metaheurísticas de los estudios más posicionados para el TSP están: Algoritmos Genéticos, Sistema Colonia de Hormigas, Búsqueda Dispersa, Búsqueda de Entorno Variable y Recocido Simulado. Con base en el estudio de Savla, Frazzoli y Bullo [17] donde se menciona el término “algoritmo aproximado de factor constante”, se destaca que algunos estudios no distinguen entre “algoritmos aproximados” y los “algoritmos heurísticos”, mientras que otros sí lo hacen. Partiendo de los que diferencian dicha agrupación, se menciona que una heurística, a pesar de ser eficiente desde el punto de vista práctico, no tiene una medida teórica de la calidad del algoritmo ni del tiempo de ejecución, en tanto que los aproximados llegan, en un tiempo polinomial, a una respuesta cercana a la óptima, pero sabiendo qué tan lejos de ella se encuentran [18] Entre los que no diferencian sobre la mencionada clasificación, están Ríos y Bard [19], Vélez y Montoya [20] y Quevedo y Ríos [21]. A medida que se avanza en el desarrollo de nuevos métodos de resolución para el TSP y, a su vez, se resuelven instancias cada vez más complejas, el probar la efectividad de dichos métodos a través de instancias de grandes tamaños no parece ser un requisito para los trabajos futuros, puesto que en los estudios analizados se encuentran experimentos válidos empleando instancias de tamaño máximo, no necesariamente crecientes a través del tiempo: 1.577 ciudades [13], 48 ciudades [22], 51 ciudades [23], 100 ciudades [24] y 1.904.711 ciudades [25]. Respecto a dichas pruebas computacionales, se destaca el aporte del TSPLIB [26], prevaleciendo como una fuente de información de amplio uso en los estudios mejor posicionados, vital para realizar comparaciones entre los métodos existentes,

facilitar validaciones de las nuevas propuestas y posibilitar el reconocimiento de las contribuciones más destacadas. En vista de los avances en el tema, el TSP aún prevalece como un importante campo de conocimiento, puesto que además del caso original, que, como se ha evidenciado, aún constituye un atractivo para la comunidad académica, se destacan también en los estudios mejor posicionados seis de sus variantes: m-TSP, PTSP, TSPPDL, TSPTW, GTSP y DTSP. Es así que desarrollar nuevos métodos para aportar respuestas satisfactorias al TSP contribuye no solo al progreso académico, sino también a un significativo avance en el desempeño de las organizaciones industriales y de servicios, cuyo modo de operación se relaciona directamente o en parte con la estructura de este problema [27, 28].

Las variables que han sido empleadas por la mayoría de los investigadores que dan solución a TSP son: *i*) Tiempo de recorrido entre ciudades: horas, minutos, días, semanas, etc.

ii) Distancia de recorrido entre ciudades: metros, kilómetros, millas, milímetros, etc.

iii) Costo de traslado: dinero, desgaste de las piezas, gasto de energía, etc.

Las variables que se pueden adoptar dependen de cada problema, por ejemplo: *i*) Circuitos electrónicos: cantidad de soldadura utilizada, menor espacio entre los puntos de soldadura de los circuitos, evitar el cruce entre las líneas de soldadura, tiempo de fabricación, distribución de los circuitos, entre otras. *ii*) Control de semáforos: Número de semáforos (nodos), tiempo de traslado entre semáforos, cantidad de autos que pasan por un punto, entre otras variables.

iii) Previsión del tránsito terrestre: puntos en una ciudad, cantidad de vehículos, tiempo de traslado, tipos de vehículos, horas pico, correlación entre variables, regresión lineal, etc.

iv) Entrega de productos: Peso de las entregas, número de entregas, nodos (domicilios) a visitar, recorridos, tiempos de traslado, tipo de vehículo, etc. *v*) Estaciones de trabajo: secuencia de actividades, lugar de las herramientas (nodos), Tipo de herramientas, tiempo de uso, etc. *vi*) Edificación: puntos de edificación (construcciones), distancia entre las construcciones y los insumos, vehículos (grúas, camiones de volteo, etc.), cantidad de combustible que emplean, etc.

3. Metodología

Este documento se presenta como un trabajo de investigación teórica que consta de los siguientes ítems *i*) Revisión del estado del arte donde se identifica relevancia del problema basado en las aplicaciones en diferentes escenarios, que se ha incluido en el referente teórico. *ii*) Definición de un modelo matemático que describa el TSP. *iii*) Definición de heurísticas que permitan la solución del TSP. *iv*) Identificación de un software donde sea posible resolver casos de la literatura especializada para comparar las respuestas obtenidas mediante las diferentes metodologías. 5) Evaluación de resultados

3.1 Modelamiento Matemático

El TSP es definido sobre un grafo $G = (V, A)$ donde V es un conjunto de nodos (vértices) y A es un conjunto de arcos (bordes). Sea $C = (C_{ij})$ un costo (distancia) matriz asociada con A . la matriz C se dice que es simétrica cuando $C_{ij} = (C_{ji}), \forall (i, j) \in A$ y asimétrica en caso

contrario. Si $c_{ij} + c_{jk} \geq c_{ik}, \forall i, j, k \in V, C$ se dice que satisface el triángulo de desigualdad. Varias formulaciones de programación entera han sido propuestas anteriormente en la literatura, entre las cuales están la formulación basada en la asignación, formulación basada en árboles y la formulación basada en el flujo de los 3 índices, las cuales se presentan a continuación:

Formulación de programación entera basada en la asignación

El TSP suele formularse usando una formulación de programación lineal entera de doble índice. Donde se define c_{ij} como el costo de transitar el arco i, j y la variable binaria x_{ij} como que se activa cuando se recorre el arco.

$$\text{Minimizar } \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (1)$$

s.a

$$\sum_{i \in A} x_{ij} = 1 \quad j \in A (i \neq j) \quad (2)$$

$$\sum_{j \in A} x_{ij} = 1 \quad i \in A (i \neq j) \quad (3)$$

Restricciones de Sub-tour

$$x_{ij} \in [1], \forall (i, j) \quad (4)$$

Donde (1) representa la minimización de los costos de los arcos activos. Las restricciones (2) y (3) indican que se sale y se ingresa una única vez al nodo. Todo los nodos deben ser de grado 2. (4) representa la naturaleza de las variables. Las restricciones se denominan restricciones de eliminación de subtours (SECs).

Estas restricciones pueden ser mostradas como:

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, \forall S \subseteq V \setminus \{1\}, S \neq \emptyset \quad (6)$$

O como alternativa la siguiente forma:

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \geq 1, \forall S \subseteq V \setminus \{1\}, S \neq \emptyset \quad (7)$$

Las restricciones (6) y (7) imponen los requerimientos de conectividad, es decir, previene la formación de subtours de cardinalidad S no incluyendo el depósito. Desafortunadamente, ambas familias de este tipo de restricciones incrementa exponencialmente si incrementa el número de nodos, por lo tanto, no es práctico resolver el problema por relajación de la programación lineal. Superado este problema por mediante la introducción de variables continuas adicionales, es decir, nodos potenciales, resultando un numero polinómico de SECs. Estos SECs son dados como (denotados como MTZ-SECs):

$$u_i - u_j + px_{ij} \leq p - 1, \text{ para } 2 \leq i \neq j \leq n. \quad (8)$$

En (8) p denota el número máximo de nodos que pueden ser visitados por un vendedor. El nodo potencial de cada nodo indica el orden del nodo correspondiente en el tour [4].

3.2 Heurísticas para resolver el TSP

Las técnicas heurísticas para resolver el TSP pueden ser constructivas como la vecino más cercano, la golosa ó Borůvka, que entregan soluciones en tiempos eficientes. También puede ser a través de heurísticas de mejoramiento con el Lin_ Kernighan, que requieren de una solución de inicio y cuyos resultados son de mejor calidad.

3.2.1 Algoritmo del vecino más cercano, *Nearest Neighbor Algorithm* (n^2) [29]

1. Seleccionar una ciudad aleatoria.
2. Encontrar la ciudad más cercana sin visitar e ir allí.
3. ¿Queda alguna ciudad sin visitar? Entonces repetir el paso 2.
4. Regresar a la primera ciudad.

3.2.2 Heurística Golosa, *Greedy Algorithm* ($n^2 \log_2(n)$) [30]

1. Ordenar todas las aristas.
2. Seleccionar la arista más corta y agregarla al tour si no viola ninguna de las restricciones anteriores (si la ciudad no es de grado 2).
3. ¿El tour contiene N aristas? Si no, repetir el paso 2.

3.2.3 Borůvka Algorithm $O(\log(n))$ [31]

1. Se calcula el vecino más cercano para cada una de las ciudades.
2. En lugar de poner una lista de prioridad, sólo se organizan las aristas en orden ascendente.
3. Se recorre la lista y se agregan los nodos.
4. Si después de recorrer la lista todavía no se tiene un tour, se repite el proceso con las ciudades que todavía no tienen grado 2 y con vecinos elegibles.
5. Se repite el paso 4 hasta formar un tour.

3.2.4 Algoritmo de Lin_ Kernighan [32]

Esta heurística es considerada como uno de los métodos más efectivos para generar soluciones de buena calidad para el problema del agente viajero. Esta heurística es una variante de la búsqueda local en profundidad, la cual comienza con un tour T factible y a partir de este se explora un vecindario de tours T_1, T_2, \dots, T_n , mediante el cambio de k arcos de T . Los k cambios son necesarios para decrementar la longitud del tour. El proceso termina cuando no es posible encontrar mejoras y se dice que el tour es $k - \text{opt}$, Lin ha estudiado los casos con $k = 2$ y $k = 3$, encontrando soluciones para el TSP de muy buena calidad y en tiempos de computo muy cortos. Con el fin de encontrar soluciones óptimas o cercanas al óptimo se recomienda iniciar el proceso de cambios de aristas desde diferentes puntos de partida. Adicional a esto, Lin y Kernighan en [32], han realizado una mejora a este algoritmo, la cual consiste en una búsqueda local en ancho 3 – Opt seguida de una búsqueda local en profundidad.

4. Resultados

Con el fin de analizar la eficiencia de las diferentes heurísticas descritas se utiliza el software Concorde [33]. El software Concorde se ejecutó en un computador Lenovo Z470 con un procesador Intel Core i5-2410M de 2.3 GHz, memoria RAM de 4 GB. Los casos analizados van desde 48 hasta 85900 nodos.

Tabla 1.

	Instancia	Vecino más cercano	Greedy	Boruvka	LK - Greedy	LK- Borukva	BKS
1	att48	40738	40159	40922	33522	33522	10628
2	a280	3270	3222	3104	2640	2651	2579
3	u574	48162	44846	44627	37218	37040	36905
4	dsj1000	24138334	21705718	22506329	18729206	18759481	18659688
5	rl5915	687859	638722	640142	583017	590888	565530
6	pla7397	30116992	27135605	27518657	23706860	23727567	23260728
7	usa13509	25456382	23225412	23672435	20048586	20042317	19982859
8	d18512	734395	737483	739505	647414	647376	645238
9	pla33810	82286372	76707590	77635146	67396059	67721965	66048945
10	pla85900	1,76E+08	163352095	166344316	144386082	144166651	142382641

Resultados obtenidos en el software Concorde

Fuente: Elaboración propia

Según se observa en la tabla 1, las respuestas han variado de acuerdo a la heurística en la que se resuelve el problema de TSP. Se debe recordar que para aplicar la heurística Lin_Kernighan es necesario tener un tour inicial, el software tiene la opción de elegir la estrategia para generar la solución de inicio, en las columnas 6 y 7 corresponden a los resultados obtenidos cuando se utiliza la heurística Greedy ó con la heurística Borukva, los mejores resultados se obtienen cuando se inicia con un tour obtenido por la heurística Greedy en el 40% de los casos, mientras que el 60% restante tiene mejor desempeño cuando se inicia con Borukva.

De forma exacta se resuelven instancias hasta de 1000 clientes en la opción de metodología exacta del software Concorde. Para instancias de mayor tamaño sólo es posible encontrar respuestas mediante las estrategias heurísticas.

La heurística Lin_Kernighan es muy eficiente tanto que para los casos de 48 a 1000 clientes encuentra la misma solución que se obtiene con el método exacto.

5. Conclusiones

Se hizo una revisión del estado del arte del TSP donde se muestra el interés académico y empresarial que ha recibido el planteamiento y la solución del mismo.

El software Concorde, es una herramienta gratuita de fácil aplicación donde se puede validar la efectividad de las diferentes heurísticas y además es posible crear instancias propias usando la aplicación de google maps.

La heurística de LK es muy eficiente y en muchas instancias entrega la solución óptima obtenida a través de un algoritmo exacto. El LK es una solución de buena calidad que puede considerarse como solución de inicio para problemas de ruteo de vehículos.

La heurística LK puede ser usada para la implementación de técnicas matheurísticas donde es posible resolver una parte del problema de forma exacta y otra de forma aproximada, debido a que está heurística entrega la solución óptima para instancias de TSP de mediano porte.

Impactos

Se presentó una revisión del estado del arte del TSP donde se identifica que es posible aplicar la definición del problema a muchas otras situaciones diferentes al ruteo de vehículos tales como secuenciamiento de tareas, control de semáforos, ubicación de edificios etc. El software Concorde es sencillo de usar y puede adaptarse a empresas de cualquier tamaño.

Bibliografía

- [1] K. H. Kim and Y.-M. Park, "A crane scheduling method for port container terminals," *European Journal of operational research*, vol. 156, pp. 752-768, 2004.
- [2] R. Angel, W. Caudle, R. Noonan, and A. Whinston, "Computer-assisted school bus scheduling," *Management Science*, vol. 18, pp. B-279-B-288, 1972.
- [3] X. Wang and A. C. Regan, "Local truckload pickup and delivery with hard time window constraints," *Transportation Research Part B: Methodological*, vol. 36, pp. 97-112, 2002.
- [4] D. L. Applegate, R. E. Bixby, V. Chvatal, and W. J. Cook, *The traveling salesman problem: a computational study*: Princeton university press, 2011.
- [5] S. Gorenstein, "Printing press scheduling for multi-edition periodicals," *Management Science*, vol. 16, pp. B-373-B-383, 1970.
- [6] L. Tang, J. Liu, A. Rong, and Z. Yang, "A multiple traveling salesman problem model for hot rolling scheduling in Shanghai Baoshan Iron & Steel Complex," *European Journal of Operational Research*, vol. 124, pp. 267-282, 2000.
- [7] A. E. Carter and C. T. Ragsdale, "Scheduling pre-printed newspaper advertising inserts using genetic algorithms," *Omega*, vol. 30, pp. 415-421, 2002.
- [8] H. A. Saleh and R. Chelouah, "The design of the global navigation satellite system surveying networks using genetic algorithms," *Engineering Applications of Artificial Intelligence*, vol. 17, pp. 111-122, 2004.
- [9] A. F. Penna, "Problema del agente viajero," *XIKUA Boletín Científico de la Escuela Superior de Tlahuelilpan*, vol. 2, 2014.
- [10] G. Laporte, "The traveling salesman problem: An overview of exact and approximate algorithms," *European Journal of Operational Research*, vol. 59, pp. 231-247, 1992.
- [11] B. Angeniol, G. D. L. C. Vaubois, and J.-Y. Le Texier, "Self-organizing feature maps and the travelling salesman problem," *Neural Networks*, vol. 1, pp. 289-293, 1988.
- [12] R. Álvarez, A. Corberán, and J. Tamarit, "La combinatoria poliédrica y el problema del viajante," *Aplicación al caso de ciento tres ciudades Españolas. Qüestió*, vol. 9, pp. 199-213, 1985.
- [13] M. Dorigo and L. M. Gambardella, "Ant-Q: a reinforcement learning approach to combinatorial optimization," Tech. Rep. IRIDIA/951995.
- [14] G. Dantzig, R. Fulkerson, and S. Johnson, "Solution of a large-scale traveling-salesman problem," *Journal of the operations research society of America*, vol. 2, pp. 393-410, 1954.
- [15] M. Held and R. M. Karp, "The traveling-salesman problem and minimum spanning trees: Part II," *Mathematical programming*, vol. 1, pp. 6-25, 1971.
- [16] M. Padberg and G. Rinaldi, "Optimization of a 532-city symmetric traveling salesman problem by branch and cut," *Operations Research Letters*, vol. 6, pp. 1-7, 1987.
- [17] K. Savla, E. Frazzoli, and F. Bullo, "Traveling salesperson problems for the Dubins vehicle," *IEEE Transactions on Automatic Control*, vol. 53, pp. 1378-1391, 2008.
- [18] J. R. M. Torres, R. Montoya, C. D. P. Arboleda, D. Paternina, and Y. Frein, "Minimización del tiempo total de flujo de tareas en una sola máquina," *Ingeniería y Desarrollo*, 2002.
- [19] R. Ríos and J. Bard, "Heurísticas para secuenciamiento de tareas en líneas de flujo," *línea*< <http://yalma.fime.uanl.mx/~roger/work/Papers/article/article-ciencia-2000.pdf>>[consulta: 2 de Junio 2010], 2000.

- [20] M. C. Vélez and J. A. Montoya, "Metaheurísticos: una alternativa para la solución de problemas combinatorios en administración de operaciones," *Revista Eia*, pp. 99-115, 2007.
- [21] D. QUEVEDO and R. RÍOS, "Uso de búsqueda Tabú en la solución del problema de asignación cuadrática," *Ingenierías*, vol. 8, pp. 40-48, 2010.
- [22] P. Larranaga, C. M. Kuijpers, R. H. Murga, I. Inza, and S. Dizdarevic, "Genetic algorithms for the travelling salesman problem: A review of representations and operators," *Artificial Intelligence Review*, vol. 13, pp. 129-170, 1999.
- [23] H. Duan and X. Yu, "Hybrid ant colony optimization using memetic algorithm for traveling salesman problem," in *Approximate Dynamic Programming and Reinforcement Learning, 2007. ADPRL 2007. IEEE International Symposium on*, 2007, pp. 92-95.
- [24] F. Carrabs, J.-F. Cordeau, and G. Laporte, "Variable neighborhood search for the pickup and delivery traveling salesman problem with LIFO loading," *INFORMS Journal on Computing*, vol. 19, pp. 618-632, 2007.
- [25] H. D. Nguyen, I. Yoshihara, K. Yamamori, and M. Yasunaga, "A new three-level tree data structure for representing TSP tours in the Lin-Kernighan heuristic," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 90, pp. 2187-2193, 2007.
- [26] G. Reinelt, "TSPLIB—A traveling salesman problem library," *ORSA journal on computing*, vol. 3, pp. 376-384, 1991.
- [27] J. L. González Velarde and R. Z. Ríos Mercado, "Investigación de operaciones en acción: Aplicación del TSP en problemas de manufactura y logística," *Ingenierías*, vol. 2, pp. 18-23, 1999.
- [28] J. I. Pérez Rave and G. P. Jaramillo Álvarez, "Relevant literary space on travelling salesman problem (TSP): contents, classification, methods and fields of inspiration," *Production*, vol. 23, pp. 866-876, 2013.
- [29] B. Golden, L. Bodin, T. Doyle, and W. Stewart, Jr. *Approximate Traveling Salesman Algorithms Operations Research* 28:3-part-ii , 694-711 , 1980
- [30] Lau, V. K., & Kwok, Y. K. R. *Channel-adaptive technologies and cross-layer designs for wireless systems with multiple antennas: theory and applications (Vol. 85)*. John Wiley & Sons, 2006
- [31] Bazlamaçcı, Cüneyt F., and Khalil S. Hindi. "Minimum-weight spanning tree algorithms a survey and empirical study." *Computers & Operations Research* 28.8 pp. 767-785, 2001
- [32] Lin, S., & Kernighan, B. W.. An effective heuristic algorithm for the travel Salesman proble, *Operations Research* 21(2), pp 498—516, 1973.
- [33] Applegate, D., Bixby, R., Chvátal, V., & Cook, W. Concorde TSP solver [Computer Software]