Análisis comparativo de las aproximaciones heurísticas Ant-Q, recocido simulado y búsqueda tabú en la solución del problema del agente viajero

Jair J. de la Cruz H.*, Adriana Mendoza B.*
Astrid del Castillo Ch.**, Carlos D. Paternina A.***
Grupo de Investigación en Sistemas Inteligentes

Resumen

En los últimos años, una de las áreas de investigación que ha adquirido mayor importancia
ha sido la de las heurísticas basadas en varios fenómenos físicos, biológicos y sociales de la vida diaria. En este artículo se evalúan y comparan las características de desempeño de las heurísticas Ant - Q (una aproximación que se deriva del comportamiento de una comunidad de hormigas), Búsqueda Tabú (fundamentada en un fenómeno social de memoria dinámica) y Recocido Simulado (una metodología análoga al proceso metalúrgico de recocido) en la solución del problema estándar del agente viajero, objetivo para el cual se implementaron principios básicos y algoritmos computacionales de cada técnica y se resolvieron algunas instancias conocidas.
Palabras clave: Agentes, lista tabú, temperatura, control, optimización, parámetros.

Abstract

During the years, one of the most important research areas has been that of heuristics based upon various physical, biological and social daily life phenomena. In this article, the performance characteristics of some heuristics are assessed and compared to solve the standard problem of the traveler agent. These heuristics are as follows: Ant-O heuristics (an approach derived from the behavior of an ant community), Taboo Searching (based on a social phenomenon of dynamic memory) and Simulated Annealing (a methodology similar to the metallurgic process of annealing). To achieve the objective, basic principles and computer algorithmic for each technique were implemented and some known instances were solved.
Key words: Agents, taboo list, temperature, control, optimization, parameters

Fecha de recepción: 3 de septiembre de 2003
Fecha de aceptación: 18 de noviembre de 2003

*Ingeniero Industrial, Magister en Ingeniería Industrial, Universidad del Norte
**Ingeniero Industrial, Candidato a Magister en Ingeniería Industrial, Universidad del Norte
***Ingeniero Industrial, Universidad del Norte. Master of Industrial Engineering y Ph. D. en Industrial Engineering, University of South Florida (USA). *cpaternina@uninorte.edu.co*

1. INTRODUCCIÓN

Desde hace varias décadas se ha venido intensificado la formulación de una gran variedad de problemas de optimización combinatoria que requieren ser abordados con metodologías generales sin dejar de considerar explícitamente el universo de variables de decisión que cada situación involucra. Estos problemas no resultan ser tan difíciles de entender, pero sus esquemas de solución, a través de técnicas exactas, son apenas aplicables cuando el número de variables de decisión es relativamente pequeño, y en general gran parte de los problemas cotidianos no cumplen esta condición.

Ante dicha realidad, muchos esfuerzos en investigación se han concentrado en desarrollar nuevas técnicas y procedimientos heurísticos, los cuales han tenido un gran auge, sin duda debido a la necesidad de disponer de herramientas que permitan resolver problemas tales como los de optimización en la clasificación NP - duros. Cabe anotar que estas técnicas no necesariamente conducen a hallar la solución óptima de un problema en tiempo polinomial, es decir, con estas metodologías pueden obtenerse muy buenas soluciones, diferentes pero cercanas a la óptima, que contribuyan a satisfacer los requerimientos del problema.

Para resolver un problema de optimización es necesario tener conocimiento acerca de la formulación matemática del mismo, además de su aspecto conceptual, con el fin de estudiar la existencia de métodos (y sus procedimientos, incluso algorítmicos) que permitan encontrar buenas soluciones en tiempo polinomial y cómo han de ser implementados, si es posible. Un área de interés mediante el cual pueden analizarse la estructura y propiedades que encierran las formulaciones matemáticas de algunos problemas es la teoría de grafos, ya que ofrece un marco de referencia acertado para iniciar la construcción de un algoritmo conveniente a la situación específica que se va a abordar. Entre los problemas «intratables» que pueden ser estudiados mediante la teoría de grafos se encuentra el problema del agente viajero, cuya finalidad es determinar la trayectoria más corta que debe seguir un agente que debe visitar todas las ciudades o clientes de una red, exactamente una vez cada uno, y regresar al final de la travesía a la ciudad o cliente inicial. Como bien es conocido, el problema del agente viajero pertenece a la clase NP- duro, por lo que ampliamente se considera irresoluble en tiempo polinomial.

Este artículo tiene como propósito aplicar las metodologías heurísticas Ant-Q, Búsqueda Tabú y Recocido Simulado en la solución de instancias pequeñas del problema del agente viajero, y además implementar algoritmos computacionales de estas técnicas para resolver instancias con muchas variables de decisión, lo cual permitirá encontrar soluciones que puedan ser consideradas buenas, sino las óptimas, en comparación con las conocidas en bibliografía. Finalmente, analiza

comparativamente las características de desempeño de los algoritmos implementados, con base en los resultados obtenidos.

2. EL PROBLEMA DEL AGENTE VIAJERO

El problema del agente viajero es un problema estándar NP - duro, en el cual un vendedor tiene que visitar cada ciudad o cliente de un conjunto dado, exactamente una vez, partiendo de una ciudad o cliente origen y regresar a éste luego de haberlos visitado a todos. El agente vendedor debe escoger una trayectoria que minimice el costo total, que en general se define en función de la distancia recorrida.Matemáticamente, una instancia I = (G, C) del problema del agente viajero se define como sigue: Dado un grafo dirigido G = (V, E), donde | V | = N y V = {1, 2, ..., N }, y una matriz de costos C = {c_{ij}} de tamaño N x N, donde cada arco (i, j) tiene asociado un costo c_{ij}, encuentre una trayectoria tal que todos los arcos del grafo sean visitados exactamente una vez y el costo total sea mínimo; es decir, hallar una permutación F (i): V Æ V de todos los vértices, existentes en exactamente un ciclo, tal que la función

$$\sum_{i=1}^{N} c_{ij}\Phi(i)$$

sea minimizada.

En general, el grafo G se considera completamente conexo, pero si no lo es, la formulación matemática puede asignar valores ∞ (o un número muy grande) a los c_{ij} para todos los arcos (i, j) que no existan en el problema que se está modelizando.

3. ANT-Q, UNA APROXIMACIÓN DE APRENDIZAJE REFORZADO

La técnica Ant-Q es un aproximación inspirada en la observación real de cómo una colonia de hormigas puede hallar la ruta más corta entre una fuente alimenticia y su nido mediante un sistema de comunicación común. Mientras van desde el alimento a su nido y viceversa, las hormigas depositan en el suelo una feromona y forma en su recorrido una pista con la sustancia. Las hormigas pueden sentirla y, cuando toman su camino, tienden a escoger las rutas marcadas con fuertes concentraciones de la pista. La feromona ayuda a las hormigas a localizar su alimento (o el nido), y también puede ser utilizada por otras para localizar sus compañeras. Análogamente, una colonia de hormigas artificiales que cooperan entre sí puede ser utilizada para encontrar una buena solución en problemas de optimización como el del agente viajero. La cooperación es la clave de la técnica Ant-Q: un conjunto de hormigas que se comunican indirectamente por estigmergia. Estigmergia es el estímulo que recibe cada hormiga por sus logros, lo que en

Aprendizaje Reforzado se conoce como refuerzo en el aprendizaje. De allí que Ant-Q sea una aproximación basada en esta meta - heurística.

En lo que respecta al problema del agente viajero, las hormigas artificiales o agentes van de una ciudad o cliente a otro según un parámetro probabilístico, y conforme se retiran de cada uno de ellos, depositan en cada arco del recorrido una feromona o señal numérica que puede ser identificada por el resto de agentes. Algunos de los autores que han estudiado las analogías algorítmicas de las hormigas son M. Dorigo, L. Gambardella y A. Colorni.

3.1. ALGORITMO COMPUTACIONAL

A partir de esta sección se hará referencia a las ciudades o clientes como nodos, y a costos como distancias entre pares de nodos (ciudades o clientes), con la notación c_{ij}.

Sea AQ (r, s) un valor real positivo asociado con el arco (r, s) que indica qué tan favorable sería visitar el nodo s si el agente se encuentra en este momento en el nodo r. Sea HE (r, s) un valor heurístico asociado con el arco (r, s) que provee una evaluación heurística de qué arcos son mejores (usualmente HE (r, s) = 1 / c_{rs}). Sea k un agente cuya tarea es completar una trayectoria que visita todos los nodos y regresa a su punto de partida, y $J_k(r)$ una lista que contiene los nodos aún no visitados por el agente k. Sea q, $0 \leq q \leq 1$ un valor escogido aleatoriamente con probabilidad uniforme, y q_0, $0 \leq q_0 \leq 1$ un parámetro de selección. Un agente k situado en el nodo r irá al nodo s utilizando la siguiente regla de escogencia:

$$s = \arg \max_{u \in J_k(r)} \{ | AQ(r,u) |^{\delta} \cdot | HE(r,u) |^{\beta} \}, \text{ si } q \leq q_0,$$

de otra manera, s se selecciona aleatoriamente en función de los valores AQ (r, u) y HE (r, u), con $u \in J_k(r)$. δ y β indican la importancia relativa de los valores AQ (r, u) y HE (r, u) en la regla respectivamente.

Por otro lado, m agentes (hormigas artificiales) son utilizadas para dar forma a los valores de aprendizaje, AQ (r, s), de modo que éstos puedan favorecer en la regla de escogencia que se debe hallar buenas soluciones del problema. Estos valores se actualizan según la siguiente regla:

$$AQ(r,s) \leftarrow (1-\alpha) \cdot AQ(r,s) + \alpha \cdot [\Delta AQ(r,s) + \gamma \cdot \max_{z \in J_k(s)} AQ(s,z)]$$

Donde α y γ son los parámetros de aprendizaje y $\Delta AQ(r,s) = 0$, excepto después de que cada agente ha completado su trayectoria, en cuyo caso $\Delta AQ(r,s) = W / L'$,

siendo W un parámetro constante y L' la longitud de la trayectoria más corta, encontrada por alguno de los agentes. El término $\Delta AQ(r, s)$ se conoce como refuerzo descontado.

Con base en estas reglas básicas, el algoritmo Ant-Q implementado viene dado así:

Paso 1. Se inicializan los valores $AQ(r, s)$ para cada arco, $AQ(r, s) = AQ_0$; se ubican los agentes en los nodos puntos de partida y se inicializan las listas de nodos aún por visitar para cada uno de los m agentes. Se utilizó $AQ_0 = 1 / (N \cdot$ longitud media de los arcos en la instancia).

Paso 2. En este paso cada agente construye una trayectoria utilizando la regla de escogencia presentada arriba. A medida que cada agente k agrega un nuevo nodo a su trayecto actualiza también la lista $J_k(r)$ y un vector donde almacena el nodo visitado, $Ruta_k$. Dado que el término $\Delta AQ(r,s) = 0$, la regla de actualización de los valores $AQ(r, s)$ se reduce a

$$AQ(r, s) \leftarrow (1 - \alpha) \cdot AQ(r, s) + \alpha \cdot \gamma \cdot \max_{z \in J_k(s)} AQ(s, z)]$$

Paso 3. Una vez los agentes han completado su trayectoria y regresado al nodo de partida, se calculan las distancias recorridas y se encuentra la mínima, L'. Con base en L' se computa el refuerzo descontado $\Delta AQ(r, s)$. Los valores $AQ(r, s)$ de los arcos (r, s) que pertenecen a la secuencia de distancia L' se actualizan con $\Delta AQ(r, s)$.

En la primera iteración, la distancia mínima encontrada L* es igual a L' (como una solución inicial) y la secuencia de visita a los nodos se almacena en Ruta*. A partir de la segunda iteración, L* y Ruta* se actualizan sólo si $L' < L^*$.

Paso 4. Si se satisface la condición de terminación, usualmente en función de un número de iteraciones, el algoritmo se detiene y L* es la menor distancia hallada, donde Ruta* es la secuencia de nodos que recorre esa mínima distancia. De cualquier otro modo, regrese al paso 2.

Para ilustrar el funcionamiento del algoritmo se propone el siguiente problema pequeño de alistamiento de máquinas, para el cual debe encontrarse la secuencia de costo mínimo. Cada costo indica el tiempo requerido para alistar una máquina que va a procesar el trabajo r dado que inmediatamente antes se procesó el trabajo s.

Tabla 1
Costo de alistamiento de máquina por cambio del trabajo r al trabajo s

r / s	1	2	3	4
1	–	19	25	21
2	24	–	23	20
3	21	25	–	22
4	20	23	21	–

Paso 1. Se inicializan los valores AQ (r, s), AQ (r, s) = 0.000946. Se utilizarán 4 agentes ubicados inicialmente en los nodos 1, 2, 3 y 4 respectivamente. Se inicializan las listas de nodos aún por visitar para cada uno de los 4 agentes. Se escoge un número de iteraciones (por ejemplo, 50) como condición de terminación.

Paso 2. En este paso cada agente construye una trayectoria utilizando la regla de escogencia. Al inicio del algoritmo se asignan $\delta = 1$, $\beta = 2$ y $q_0 = 0.9$. Por ejemplo, para el agente 1 al elegir el segundo nodo que debe visitar, genera q = 0.392, luego la regla de escogencia permite determinar que el nodo 2 debe ser visitado ahora, por ser el arco (1, 2) el de mayor valor en la función s.

En cuanto a la actualización de valores AQ (r, s), por ejemplo, para el arco recorrido por el agente 1, $AQ(1,2) \leftarrow [(1-0.1)\cdot 0.000946 + 0.1 \cdot 0.3 \cdot 0.000946]$, donde los valores $\alpha = 0.1$ y $\gamma = 0.3$ fueron asignados al inicio del algoritmo. Los siguientes agentes a escoger nuevos nodos a visitar tendrán en cuenta que el valor de AQ (1, 2) ha sido actualizado.

Para el agente 2, q = 0.971, luego escoge aleatoriamente el siguiente nodo que vaya a visitar, por ejemplo, el 3. Para el agente 3, q = 0.487, luego la regla de escogencia permite determinar que el nodo 1 debe ser visitado ahora. Actualiza el valor AQ (3.1).

Este procedimiento es similar para los agentes restantes y se repite hasta que cada lista de nodos aún no visitados es agotada, es decir, $J_k(r) = \varnothing$ para todas las k. En este momento los agentes regresan a su punto de partida.

Paso 3. Se calcula la distancia recorrida por cada agente. En la siguiente tabla aparecen las trayectorias seguidas por cada agente y la distancia respectiva de cada secuencia.

Tabla 2

Agente	Secuencia	Costo
1	1, 2, 4, 3, 1	81*
2	2, 4, 1, 3, 2	90
3	3, 1, 4, 2, 3	86
4	4, 1, 2, 3, 4	84

Note que L' = 81. Como el algoritmo se encuentra en la primera iteración, L*= 81 y Ruta* = {1, 2, 4, 3, 1}. Se computa el refuerzo descontado con W = 10 como parámetro asignado al inicio del algoritmo, $\Delta AQ(r, s) = 10/81$, y se actualizan los valores AQ (r, s) de los arcos que pertenecen a la secuencia
{1, 2, 4, 3, 1}.

Paso 4. Si se satisface la condición de terminación, el algoritmo se detiene. Como sólo se ha ejecutado una primera iteración, el algoritmo vuelve al paso 2.

Note que al regresar al paso 2 cada agente recibe ahora la información que entre todos han reunido; en otros términos, en cada oportunidad que los agentes artificiales tienen de volver a encontrar una trayectoria mínima, ya cada uno de ellos habrá aprendido de la experiencia en el pasado, lo que en la analogía de las hormigas no es más que el rastro de feromona en el camino. Por otro lado, el parámetro q permite a las hormigas aventurarse a explorar nuevas rutas con poco o ningún conocimiento acerca de las soluciones que puedan construirse.

4. RECOCIDO SIMULADO, BASADO EN UN PROCESO METALÚRGICO

La heurística Recocido Simulado se basa en el proceso metalúrgico de recocido: Cuando un metal es calentado hasta una temperatura muy elevada, sus átomos pueden moverse a altas velocidades, y si éste es enfriado muy lentamente, los átomos se arreglan en estructuras o patrones de mínima entropía, suministrando al metal mucha más resistencia que antes. En la técnica, las soluciones construidas en las primeras iteraciones de los algoritmos describen grandes perturbaciones en las variables de decisión y en la función que se va a optimizar, y a medida que un parámetro de control va ajustándose lentamente (equivalente a la disminución de temperatura), las nuevas soluciones van siendo más selectivas, incluso admitiendo algunos empeoramientos, pero finalmente conformando una solución que puede considerarse como buena, teniendo en cuenta toda la información generada en los pasos anteriores y admitiendo sólo mejoramientos en las soluciones iniciales.

Cabe anotar que el Recocido Simulado ha sido probado con éxito en numerosos problemas de optimización, y ha demostrado gran habilidad para evitar quedar atrapado en óptimos locales. Esta metodología experimentó un gran auge en la década de los ochenta, desde 1983, año en que Kirlpatrick, Gelatt y Vecchi lo proponen como un procedimiento para obtener soluciones aproximadas a problemas de optimización como el del agente viajero.

4.1. ALGORITMO COMPUTACIONAL

El algoritmo de Recocido Simulado implementado tiene básicamente la configuración de un procedimiento de inserción. El parámetro de control es el número de nodos incluidos en las soluciones, iniciando con un conjunto reducido en que se van insertando estratégicamente y de modo muy lento los demás nodos hasta obtener soluciones completas de la instancia. La variable objetivo en este caso no es exactamente la distancia total recorrida por el agente, ya que en soluciones intermedias, donde no se han incluido todos los nodos, no es posible calcularla. Para este inconveniente se implementa el valor esperado de la distancia total recorrida como medida de desempeño que se va a optimizar, con la ventaja de que al poder generar soluciones completas, este valor esperado será igual al de la función objetivo.

Formalmente, sea T_i la temperatura inicial o número inicial de nodos, donde $T_i = |[kN]|$, con $0 < k < 1$. Entre mayor sea la temperatura inicial o menor sea el conjunto inicial de nodos, mayor será la calidad de la solución final. Sea T_f la temperatura final, $T_f = N$. Sea $f(x)$ el valor de la función objetivo para una solución específica x, y x^* la mejor secuencia hallada, con valor objetivo $f^*(x)$. Nótese que x^* es equivalente al vector Ruta* del algoritmo Ant-Q, y similar para $f^*(x)$ y L^*. Sea $E(x)$ el valor esperado de la función objetivo para una solución específica x.

El conjunto inicial de nodos, de tamaño T_i, se selecciona mediante un procedimiento miope: Se escogen los T_i primeros nodos distintos en un vector ordenado de todos los arcos de mayor a menor distancia. De este modo, los nodos de la solución inicial tienden a generar un valor esperado de distancia muy alto (de poca calidad) y que puede ser mejorado de manera significativa con las inserciones posteriores, similar a lo que sucede en las metodologías de búsqueda local mediante descenso máximo.

En cuanto a las inserciones, existe una lista J de nodos aún no incluidos en la solución (como la lista implementada para cada agente en Ant- Q) de la que se extrae una lista I de candidatos que van a ser insertados. Para que un nodo de la lista J sea incluido en la lista I debe cumplirse que para los nodos extremos de algún arco en la secuencia de la solución corriente la suma de las distancias de los

dos arcos que se forman al insertar tal nodo en medio de los extremos sea mínima. Usualmente, los arcos que se toman de la secuencia actual para llevar a cabo este procedimiento son aquellos que se consideran de mayor distancia dentro de la solución, y en este caso se tienen en cuenta los arcos cuya distancia se encuentra sobre el promedio de los arcos de la secuencia. Finalmente, se selecciona un nodo aleatoriamente de la lista, a partir de una distribución discreta basada en las distancias tentativas de inserción.

Con base en estos procedimientos, el algoritmo de Recocido Simulado viene dado así, con la notación especificada:

Paso 1. Se asigna un número determinado de nodos a la temperatura inicial, T_i. Se ordenan los arcos de mayor a menor distancia y se escogen los T_i primeros nodos de la lista. Se construye una solución inicial, x_0, con los nodos seleccionados. Esta solución inicial se asigna a x, $x \leftarrow x_0$, y el valor esperado de la distancia total recorrida se asigna a f (x), $f(x) \leftarrow E(x_0)$. Además, se inicializan $x^* \leftarrow x_0$ y $f^*(x) \leftarrow E(x_0)$.

Paso 2. Se asigna $T_i \leftarrow T_i + 1$ y se selecciona nuevo nodo para insertar con la política que se implementó. El nodo escogido se inserta para obtener la nueva solución x. Ahora se aplica un procedimiento de intercambios, 2-Opt, con la finalidad de reorganizar la secuencia y de tal modo que el valor esperado de la distancia total sea mínimo. Se actualiza x con la secuencia resultante y además f (x), $f(x) \leftarrow E(x)$.

Paso 3. Si se satisface la condición de terminación, $T_f = N$, el algoritmo se detiene y se calcula la distancia de la secuencia, que viene siendo el mismo valor esperado. Esta secuencia será la mejor encontrada, x*, y así mismo la distancia recorrida, f*(x). De cualquier otro modo, regrese al paso 2.

Para ilustrar el funcionamiento del algoritmo, se aplica para el mismo problema de la tabla 1.

Paso 1. Se asigna $T_i = 2$. Los T_i primeros nodos en la lista de arcos ordenados son los extremos del mayor arco (3, 2), de longitud 25. La solución inicial x_0 viene dada por la secuencia {3, 2, 3}, cuya distancia total es 25 + 23 = 48. El valor esperado de esta secuencia viene dado por $E(x_0) = 48 \cdot 4/2 = 96$. Se inicializan x*y f *(x).

Paso 2. Se asigna $T_i \leftarrow 3$ y se selecciona nuevo nodo a insertar, así: el promedio de los arcos de la secuencia actual es 24, luego el único arco base es el (3, 2) de longitud 25. La lista J tiene como nodos aún por visitar el 1 y el 4. Si se insertara el nodo 1 en medio de los extremos del arco base, 3 y 2, la distancia total

tentativa del par de arcos que se forman es 21 + 19 = 40; si se insertara el nodo 4, esta distancia sería 22 + 23 = 45. La lista de candidatos contiene sólo al nodo 1, por lo tanto se escoge el nodo 1, y la nueva secuencia es {3, 1, 2, 3}. Esta solución tiene distancia igual a 40 + 23 = 63 y su valor esperado viene dado por E (x) = 63 · 4 / 3 = 84.

La siguiente tabla muestra los intercambios posteriores en busca de una distancia mínima en la solución intermedia:

Tabla 3

Secuencia intercambiada	Distancia total	Valor esperado
1, 3, 2, 1	74	98.67
3, 2, 1, 3	74	98.67
2, 1, 3, 2	74	98.67

Nótese que las distancias totales y esperadas son iguales en todos los casos, ya que las secuencias generadas con los intercambios corresponden a una misma secuencia. En efecto, esto sucede porque la solución sólo tiene 3 nodos, pero en problemas de mayor tamaño se podrían generar soluciones con distintos valores. Finalmente, se actualiza x con la secuencia resultante, {3, 1, 2, 3}, y además f (x), f (x) ← 84.

Paso 3. Como $T_i \neq T_f$, el algoritmo regresa al paso 2 para insertar el último nodo en la lista J, el 4.

Paso 2. Se asigna $T_i \leftarrow 4$. El promedio de los arcos de la secuencia actual es 21, luego existe un sólo arco base, el (2, 3), cuya longitud es 23. Se inserta el nodo 4 para generar la solución {3, 1, 2, 4, 3}. Esta solución tiene distancia igual a 40 + 20 + 21 = 81 y su valor esperado viene dado por E (x) = 81 · 4 / 4 = 81.

La siguiente tabla muestra los intercambios posteriores:

Tabla 4

Secuencia intercambiada	Distancia total	Valor esperado
1, 3, 2, 4, 1	90	90
3, 2, 1, 4, 3	91	91
3, 4, 2, 1, 3	94	94
2, 1, 3, 4, 2	94	94
3, 1, 4, 2, 3	88	88
4, 1, 2, 3, 4	84	84

Se actualiza x con la secuencia resultante {3, 1, 2, 4, 3}, y f (x) ← 81.

Paso 3. Como $T_i = T_f = 4$, el algoritmo termina, y la mejor solución encontrada x^* es la secuencia {3, 1, 2, 4, 3}, de distancia $f^*(x) = 81$.

Note que quizá la mayor contribución del algoritmo a la solución final es que a medida que la temperatura disminuye lentamente, en la secuencia van quedando fijos los mejores elementos, aunque sólo sea una solución parcial, lo cual coincide con el proceso metalúrgico, ya que la secuencia última va adquiriendo la mejor estructura posible hasta solidificar, y en cada transición aumenta la oportunidad de encontrar mejores ordenamientos gracias a que las etapas de intercambio son cada vez más extensas.

5. BÚSQUEDA TABÚ ANALOGÍA CON UN FENÓMENO SOCIAL

En la actualidad, la palabra «tabú» se utiliza para denotar prohibiciones sociales concebidas a partir de costumbres y que son a la vez una especie de medidas protectoras contra algo que constituye un riesgo. Por esta razón, la Búsqueda Tabú es una metodología que involucra medidas o estrategias que conducen un proceso de búsqueda de tal manera que el espacio de soluciones a un problema sea explorado evitando la optimalidad local, lo cual precisa un riesgo. No obstante, la relación más importante entre Búsqueda Tabú y el sentido original de la palabra es que, como realmente sucede, los tabúes son transmitidos de una generación a otra a través de las costumbres, y éstas no son más que un tipo de memoria social dinámica o memoria adaptativa. En la metodología, esta memoria permite a la búsqueda abordar el espacio de soluciones efectivamente, apoyada en la premisa de que una mala escogencia estratégica puede contener más información que una buena selección al azar. Este último aspecto se conoce como exploración sensitiva.

En adición, aparte de algunas otras propiedades de memoria, la Búsqueda Tabú intensifica la búsqueda en aquellas regiones del espacio de soluciones que halla atractivas, en términos de calidad, explorando detenidamente las vecindades de las mejores soluciones encontradas; y en contraste, induce al procedimiento de búsqueda a explorar regiones no visitadas, como una estrategia de diversificación. Los autores que han desarrollado ampliamente la teoría y práctica de la Búsqueda Tabú son F. Glover y M. Laguna.

Para el problema del agente viajero, como para otros que pueden abordarse con esta técnica, es necesario obtener primero una solución inicial con algún procedimiento convencional, y es éste el punto de partida para crear algunas vecindades en busca de mejores soluciones y con objeto de conocer el espacio de soluciones. Una vecindad puede generarse a partir de una solución mediante un intercambio, como en el algoritmo que se presenta a continuación. En lo que se refiere a la memoria adaptativa, la metodología requiere además implementar

una lista que almacene información acerca de las soluciones evaluadas y cuya longitud venga dada por un parámetro.

5.1. ALGORITMO COMPUTACIONAL

Sea f (x) la función que se va a minimizar sujeta a $x \in X$, donde X es el conjunto de soluciones que satisfacen las restricciones formuladas para el problema del agente viajero y cada vector x contiene las variables de decisión. Cada $x \in X$ tiene asociada una vecindad $N(x) \subset X$, y cada solución $x' \in N(x)$ se construye a partir de x mediante un movimiento. Sea m (x) una función generadora de vecindades N (x). Considere una secuencia finita $\{x_t\}$, $1 \leq t \leq k$, $x_{t+1} \in N(x_t)$, luego un conjunto ordenado $\varphi = \{(i_k, j_k), (i_{k-1}, j_{k-1}), \ldots, (i_{k-l+1}, j_{k-l+1})\}$ es una lista tabú (de longitud l) si los vectores x_t y x_{t+1} difieren en las coordenadas (i_t, j_t). Finalmente, sea $N(x_t, \varphi)$ el conjunto de soluciones $x' \in N(x_t)$ que no pertenecen a la lista φ, donde $N(x_t, \varphi)$ puede ser vacío dado un conjunto $N(x_t)$ no vacío. Con base en esta notación se implementa el siguiente algoritmo de Búsqueda Tabú:

Paso 1. Se construye una solución inicial $x_0 \in X$, mediante un algoritmo de inserción. Asigna a la lista tabú $\varphi \leftarrow \emptyset$. Un contador $t \leftarrow 0$, y $x \leftarrow x_0$, $f^*(x) \leftarrow f(x_0)$.

Paso 2. Genera una vecindad $N(x_t, \varphi)$ a partir de x_t con $m(x_t)$. $m(x_t)$ consiste en hacer intercambios entre pares de nodos de la secuencia x, similar a como se mostró en Recocido Simulado.

Paso 3. Se encuentra una $x_{t+1} \in N(x_t, \varphi)$ de tal modo que $f(x_{t+1})$ sea igual al menor de los $f(x_{t+1})$, $x_{t+1} \in N(x_t, \varphi)$.

Si existe un $x_{t+1} \in N(x_t)$, $x_{t+1} \in \varphi$, tal que $f(x_{t+1})$ es menor que $f^*(x)$, $x \in X$, entonces haga $x \leftarrow x_{t+1}$, $f^*(x) \leftarrow f(x_{t+1})$. Esta regla se conoce como criterio de aspiración.

Paso 4. Actualice la lista tabú φ y haga $t \leftarrow t + 1$. Si la condición de terminación se satisface, detenga el algoritmo; de cualquier otro modo, regrese al paso 2. Los movimientos que ingresan a la lista tabú son los inversos de los generados, es decir, $m^{-1}(x_{t+1})$.

Para ilustrar el funcionamiento del algoritmo, se aplica para el mismo problema de la tabla 1.

Paso 1. Se construye una solución inicial $x_0 \in X$, mediante un algoritmo de inserción, por ejemplo {1, 2, 3, 4, 1}. La lista tabú $\varphi \leftarrow \emptyset$ tiene longitud l. El contador $t \leftarrow 0$, y $x \leftarrow x_0$, $f^*(x) \leftarrow f(x_0) = 85$. Se escoge como criterio de terminación un número de iteraciones, t_f.

Paso 2. Genera una vecindad $N(x_t, \varphi)$ a partir de x_t con $m(x_t)$, así:

Tabla 5

Vecindad	Costo
2, **1**, 3, 4, 2	94
3, 2, **1**, 4, 3	91
4, 2, 3, **1**, 4	88
1, **3**, **2**, 4, 1	90
1, **4**, 3, **2**, 1	91
1, 2, **4**, **3**, 1	**81**

Paso 3. El menor $f(x_{t+1})$, $x_{t+1} \in N(x_t, \varphi)$, es 81, donde $x_{t+1} = \{1, 2, 4, 3, 1\}$. Como se satisface el criterio de aspiración, se actualiza $x \leftarrow \{1, 2, 4, 3, 1\}$ y $f^*(x) \leftarrow 81$.

Paso 4. Se actualiza la lista tabú, $\varphi \leftarrow \varphi + (3, 4)$ y $t \leftarrow 1$. Como la condición de terminación no se satisface, el algoritmo regresa al paso 2. Nótese que el movimiento ingresado a la lista tabú es el inverso a (4, 3), que caracteriza la solución que pasa a la siguiente iteración.

Paso 2. Genera una nueva vecindad $N(x_t, \varphi)$ a partir de x_t con $m(x_t)$. El movimiento (3,4) se encuentra bloqueado actualmente por la lista tabú.

Paso 3. El menor $f(x_{t+1})$, $x_{t+1} \in N(x_t, \varphi)$ es 88, donde $x_{t+1} = \{1, 4, 2, 3, 1\}$.

En el paso 4 la secuencia {1, 4, 2, 3, 1} regresa al paso 2, y así sucesivamente hasta que se cumple el criterio de terminación. En este momento, x es la mejor secuencia encontrada y tiene valor objetivo $f^*(x)$.

Nótese que al bloquear algunos movimientos por ser clasificados como tabú, la exploración se reduce aun más que en un procedimiento corriente de búsqueda local, además de admitir soluciones estratégicamente no tan buenas. No obstante, una manera de garantizar que en la corrida del algoritmo se encontrarán soluciones muy buenas depende en parte de la calidad de la función generadora de las vecindades, ya que ésta debe poder mejorar los atributos de desempeño de las soluciones que van siendo aceptadas por la estrategia sensitiva.

En lo que respecta a *la lista tabú*, un parámetro de interés es su longitud. La permanencia de cierto movimiento dentro de la lista está expresada en términos de un determinado número de iteraciones l. Si la longitud l ha sido alcanzada, el último elemento de la lista, que es el más antiguo, se remueve. Si este movimiento retirado aparece de nuevo exactamente en la misma solución por la cual fue incluido en la lista, es casi seguro que el algoritmo empiece a ciclar. Por

esta razón, la memoria adaptativa debe tener en cuenta también los aspectos de frecuencia de ingreso de un movimiento a la lista, así como qué tan reciente sucedió dicho ingreso, así como su remoción.

6. RESULTADOS COMPUTACIONALES EN INSTANCIAS CONOCIDAS

En la siguiente tabla se presentan los resultados obtenidos con cada una de las metodologías heurísticas analizadas, resolviendo tres instancias distintas y conocidas del problema, ftv33, ry48p y kro124p, cuyas soluciones óptimas han sido previamente obtenidas mediante la técnicas de Ramificación y Acotamiento (Branch and Bound) o con búsqueda exhaustiva, y aparecen registradas en fuentes bibliográficas. Las meta- heurísticas aparecen identificadas como AQ, para Ant-Q, SA, para Recocido Simulado y TS para Búsqueda Tabú. El tamaño de muestra, o número de veces que fue ejecutado cada algoritmo con cada instancia, es 10. Para cada metodología aplicada se especifican también las mejores soluciones obtenidas, la media muestral, desviación estándar y la duración media de cada algoritmo computacional (en segundos), con un procesador de 800 MHz y 128 MB de memoria RAM.

Tabla 6

Nombre del problema	ftv33			ry48p			kro124p		
METODOLOGÍA APLICADA	AQ	SA	TS	AQ	SA	TS	AQ	SA	TS
Número de clientes	34	34	34	48	48	48	100	100	100
Mejor conocida	1286	1286	1286	14422	14422	14422	36230	36230	36230
Mejor encontrada	1296	1318	1298	14511	14709	14608	37296	38437	37415
Error relativo	0,78%	2,49%	0,93%	0,62%	1,99%	1,29%	2,94%	6,09%	3,27%
Media de la muestra	1312,1	1344,7	1310	14710,5	15272,3	14708,2	38967,4	39974,2	38791,8
Desviación estándar	10,92	26,09	9,94	193,73	331,83	103,13	1163,53	765,44	909,91
Duración algoritmo	88,3	15,8	26,3	195,1	21,7	46,2	1134,6	263,1	243,6

Para el algoritmo implementado con Ant-Q se emplearon en cada caso N agentes, y α=0.1, β = 2, γ = 0.3, q_0 = 90, con 350 iteraciones; para el algoritmo de Recocido Simulado se escogió como temperatura inicial t_i = 3 clientes y 350 iteraciones en el procedimiento de intercambios interno a cada nivel de temperatura; en el algoritmo de Búsqueda Tabú se tomó como solución inicial una construida mediante inserciones, la longitud de la lista tabú l = 75 movimientos y se ejecutaron 350 iteraciones. Todas las técnicas fueron mejoradas adicionalmente con 75 iteraciones de intercambios 2 - Opt. Los parámetros que han sido escogidos para cada una de las metodologías son los sugeridos en la bibliografía.

De los resultados obtenidos puede observarse que la solución encontrada por los algoritmos implementados no coincide en ninguno de los casos con la óptima conocida, a pesar de que algunos autores sí han logrado tales valores. Esto se

justifica porque estas aproximaciones computacionales, aunque son completas, es decir, contienen todas las pautas sugeridas en la bibliografía, no tienen en cuenta algunas características especiales como las de memoria en el caso de Búsqueda Tabú, y que podrían ser objeto de posteriores esfuerzos investigativos y de desarrollo. Sin embargo, las mejores soluciones halladas se encuentran todas por debajo o muy cercanas del 5% del valor óptimo conocido. En el caso de Ant-Q, se obtuvo una diferencia relativa tan solo del 0.62% en el mejor de los casos (instancia ry48p) y de 2.94% en el peor (instancia kro124p), aunque en el promedio la Búsqueda Tabú encontró mejores soluciones. La técnica de Recocido Simulado, tal como se implementó, encontró algunas buenas soluciones, pero en general podría ser mejorada, por ejemplo, aplicando una estrategia de múltiples agentes cuyos puntos de partida sean distintos conjuntos de nodos a la misma temperatura.

En lo referente a dispersión de las soluciones, el algoritmo Ant-Q y de Recocido Simulado son los que entregan soluciones con mayor desviación estándar, sobre todo en los problemas más grandes. Acerca del tiempo computacional, note que la Búsqueda Tabú en general es bastante rápida para la calidad de soluciones que encuentra; en cambio, Recocido Simulado suele ser rápido pero con soluciones no tan buenas como las encontradas por las otras dos técnicas. El algoritmo Ant-Q es relativamente lento, quizá porque la estrategia de múltiples agentes tiene tamaño N. M. Dorigo y L. Gambardella sugieren utilizar 10 agentes, pero el algoritmo se ejecuta a 5.000 o más iteraciones. Al respecto, en pruebas preliminares con un número mayor de iteraciones para cada uno de los tres algoritmos implementados, se hizo notable que después de 600 iteraciones no ocurre un mejoramiento significativo en el valor objetivo.

Finalmente, y como ventaja para las tres aproximaciones presentadas, cabe mencionar que existe un amplio número de problemas sobre los cuales pueden aplicarse y adaptarse de modo relativamente sencillo; y en estas implementaciones que fueron puestas en marcha, con muy poca experiencia, se pudo observar que pueden obtenerse en general soluciones muy buenas sin requerir grandes esfuerzos computacionales ni algorítmicos: sólo con tener en cuenta los principios generales e ideas básicas de cada teoría, y como pudo verse además en el ejemplo resuelto con las tres técnicas, el funcionamiento de los algoritmos resulta poco complicado de entender.

CONCLUSIONES

Como conclusión puede afirmarse que las estrategias implementadas de acuerdo con cada filosofía descrita permiten determinar muy buenas soluciones a las distintas instancias en que se utilizó. Para la primera, Ant-Q, la mayor desventaja es quizás el tiempo que requiere para converger a una solución de alta calidad,

pero en contraste con las otras dos técnicas, que son más rápidas, ofrece mejores soluciones. La ventaja de Recocido Simulado obedece más a la facilidad para implementarlo, aunque puede llegar a ser tan complejo como se desee con el fin de obtener soluciones más convincentes. La Búsqueda Tabú, por su parte, produce excelentes resultados en promedio y en un tiempo menor, sin embargo, implantar en los algoritmos todas las características de memoria que implica esta metodología puede no ser tan sencillo.

Como pudo observarse también en los resultados obtenidos, las tres técnicas implementadas permiten encontrar una buena solución a los problemas propuestos, pero ninguna garantizó optimalidad. No obstante, en la cotidianidad, poseer herramientas de este tipo es de enorme utilidad ante la necesidad de poder resolver un problema tan pronto como éste se presente, es decir, como los requerimientos de soluciones son prácticamente inmediatos, es preferible sacrificar una mejor solución que pueda encontrarse en un tiempo muy extenso si puede obtenerse en este instante una muy buena alternativa. Lo mismo se puede aplicar desde el punto de vista del costo de evaluar todas las posibles soluciones. En este sentido, las técnicas aquí estudiadas pueden ser utilizadas para agilizar muchas de las decisiones del día a día.

El tercer aspecto que cabe anotar es que la tecnología moderna, especialmente en el área informática, ha permitido mejoras considerables en la planeación, diseño y control de la producción, ya que son herramientas de apoyo a las técnicas heurísticas y metaheurísticas que ayudan a solucionar problemas tan complejos como la programación de las operaciones en una línea de producción o el enrutamiento de una flota de vehículos cuya capacidad es restringida, y proporcionan soluciones muy buenas y garantizan a los usuarios tomar decisiones acertadas sobre tales situaciones, así como cierta flexibilidad para adaptarlas a distintos escenarios o sistemas sometidos a entornos altamente cambiantes.

Referencias

ASKIN, R. & STANRIDGE, Ch., *Modeling and Analysis of Manufacturing Systems*. Estados Unidos: John Wiley & Sons, 1993.

COLORNI, A., TRUBIAN M., RIGHINI, G. & MAFFIOLI, V., Heuristics From Nature For Hard Combinatorial Optimization Problems. *International Transactions in Operational Research*, 3, 1, 1993.

DÍAZ, A., GLOVER, F. GHAZIRI, A. M., GONZÁLEZ, J. L., LAGUNA, M. MOSCATO P. & TSENG, F. T., *Optimización heurística y redes neuronales en dirección de operaciones e ingeniería*. España: Paraninfo, 1996.

DÓRIGO, M., DI CARO, G. & GAMBARDELLA, L., Ant Algorithms for Discrete Optimization. *Artificial Life*, 5, 3, 1999.

DÓRIGO, M.& GAMBARDELLA, L., A study of some properties of Ant-Q. *Proceedings of 4th. International Conference on Parallel Problem solving from Nature*. Berlín, 1996.

GAMBARDELLA, L. & DÓRIGO M., Ant Colonies for the Traveling Salesman Problem. *BioSystems*, 1997.

——— Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation*, 1, 1, 1997.

GAMBARDELLA, L. & DÓRIGO, M., Ant-Q: A Reinforcement Learning Approach to the Traveling Salesman Problem. *Proceedings of ML – 95, 20th. International Conference on Machine Learning*. Morgan Kaufmann, 1995.

GLOVER, F. & LAGUNA M., *Tabú Search*. Boston: Kluwer Academic Publishers, 1997.

GUTIÉRREZ, M. *et al.*, Optimización con Recocido Simulado para el problema de conjunto Independiente. URL: http://www.azc.uam.mx/publicaciones/enlinea2/3-2.htm

HERTZ, A.,TAILLARD, E. & DE WERRA, D., *A Tutorial On Tabu Search*. Département de Mathématiques, Université de Montreal. Canada, 1997.

JHONSON, D. & MCGEOCH, L., *Experimental analysis of the heuristics for the Symmetric Traveling Salesman Problem*, 2002.

——— *The Traveling Salesman Problem: A case study in Local Optimization*, 1995.

KIRKPATRICK, S. GELATT, S. & VECCHI, M., Optimization by Simulated Annealing. *Science 220*, 4598, 1983.

MARTI, R., Templado Simulado. URL: http://www.uv.es/~rmarti/templar.html

NILSSON, C. *Heuristics for the Traveling Salesman Problem*. Linköping University, 2003.

PINEDO, M. & CHAO, X., *Operations Scheduling with Applications in Manufacturing and Services*. Estados Unidos: Irwin / McGraw-Hill, 1999.

RÍOS, D., RIOS, S. & MARTÍN, J., *Simulación: Métodos y aplicaciones*. Madrid: Alfaomega, 2000.

RÍOS, R. & GONZÁLEZ, J., Investigación de operaciones en acción: Heurísticas para la solución del Problema del Agente Viajero. *Ingenierías, 3*, 9, 2000.

SINGH, A.,RANGASWAMY, B. & MEERAN, S., Job - *Shop Neighborhoods and Move Evaluation Strategies*, 1998.

SIPPER, D. & BULFIN , R., *Planeación y Control de la Producción*. Estados Unidos: McGraw-Hill, 1998.

SUTTON, R. & BARTO, A., *Reinforcement Learning: An Introduction*. John Wiley & Sons, 1998.