



Revista de Matemática: Teoría y

Aplicaciones

ISSN: 1409-2433

mta.cimpa@ucr.ac.cr

Universidad de Costa Rica

Costa Rica

LÓPEZ, ERASMO; SALAS, ÓSCAR; MURILLO, ÁLEX

EL PROBLEMA DEL AGENTE VIAJERO: UN ALGORITMO DETERMINÍSTICO USANDO
BÚSQUEDA TABÚ

Revista de Matemática: Teoría y Aplicaciones, vol. 21, núm. 1, enero, 2014, pp. 127-144

Universidad de Costa Rica

San José, Costa Rica

Disponible en: <http://www.redalyc.org/articulo.oa?id=45331281008>

- ▶ Cómo citar el artículo
- ▶ Número completo
- ▶ Más información del artículo
- ▶ Página de la revista en redalyc.org



Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

EL PROBLEMA DEL AGENTE VIAJERO:
UN ALGORITMO DETERMINÍSTICO
USANDO BÚSQUEDA TABÚ

THE TRAVELING SALESMAN PROBLEM:
A DETERMINISTIC ALGORITHM
USING TABU SEARCH

ERASMO LÓPEZ* ÓSCAR SALAS† ÁLEX MURILLO‡

*Received: 28/Feb/2012; Revised: 31/May/2013;
Accepted: 1/Nov/2013*

*Colegio Técnico Profesional de Upala (CTPU), Upala, Costa Rica.
E-Mail: erlopez1@costarricense.cr
†Departamento de Matemática, Universidad Nacional, Heredia, Costa Rica.
E-Mail: oscar.salas@unipv.it
‡Sede del Atlántico, Universidad de Costa Rica, Turrialba, Costa Rica.
E-Mail: alex.murillo@ucr.ac.cr

Resumen

Se implementa un algoritmo correspondiente al método Búsqueda Tabú, llamado EraDeterminístico, experimentando con el algoritmo básico que explora el espacio de búsqueda e incorporando la diversificación como estrategia para explorar nuevas regiones. El algoritmo se desarrolla en el ambiente de programación Visual Basic 6.0 y la implementación se orienta a encontrar soluciones cercanas a la óptima del problema NP–Completo del Agente Viajero Simétrico (AVS). Para probar la funcionalidad, el modelo se compara con algunas instancias de la Travel Salesman Problem Library (TSPLIB), algunas instancias aleatorias y se aplica a tres situaciones reales. Finalmente, se presenta una sección donde se exponen recomendaciones y conclusiones, las cuales orientan posibles desarrollos futuros de ésta teoría y donde se pone en evidencia las bondades y la eficiencia de la implementación.

Palabras clave: búsqueda tabú; algoritmo determinístico; matriz de frecuencias; diversificación; problema del agente viajero.

Abstract

We implement an algorithm corresponding to the Taboo Search method, called EraDeterministic, experimenting with the basic algorithm that explores the search space and incorporating the diversification as strategy to explore new regions. The algorithm is developed in the programming environment Visual Basic 6.0 and the implementation is aimed at finding close solutions to the optimum of the problem NP–complete of the Symmetric Traveling Salesman (STS). To test the functionality, the model is compared with some instances of the Travel Salesman Problem Library (TSPLIB), some random instances and applied to three real-life situations. Finally, we present a section with comments and conclusions, that guide us on possible future developments that demonstrate the benefits and the efficiency of the implementation.

Keywords: tabu search; deterministic algorithm; frequencies matrix; diversification; permutation; traveling salesman problem.

Mathematics Subject Classification: 65N55.

1 Introducción

Los algoritmos determinísticos son una familia de algoritmos cuya meta es precisamente dar soluciones aproximadas a problemas generales de tipo NP, sin necesidad de recorrer todo el espacio de búsqueda. Las heurísticas clásicas realizan una exploración limitada sobre el espacio de búsqueda y normalmente las

soluciones producidas son buenas en poco tiempo. Su implementación es sencilla y son fácilmente adaptables a problemas del mundo real. Existen dos tipos de heurísticas: constructivas y de mejoramiento o de búsqueda local (Soto, 2008) [9]. El algoritmo implementado se ubica en las segundas, ya que pertenece a la familia de algoritmos cuya meta es precisamente dar soluciones aproximadas a problemas generales de tipo NP (por su acrónimo en inglés *nondeterministic polynomial time*), sin necesidad de recorrer todo el espacio de búsqueda.

Principalmente, los algoritmos de búsqueda tabú a menudo usan la experiencia ganada en búsquedas previas (memoria) para guiar nuevas búsquedas, con el fin de encontrar soluciones cercanas a las óptimas y tienen la opción de incorporar mecanismos para evitar caer en áreas limitadas del espacio de búsqueda. Para evitar caer en estas áreas, balancean dinámicamente la exploración de la experiencia almacenada en la matriz histórica de los mejores o peores movimientos y la exploración del espacio de búsqueda, llamada diversificación. Este balance es necesario, por un lado, para rápidamente identificar la región en el espacio de búsqueda donde las soluciones son buenas, por otra parte, para no perder mucho tiempo buscando dentro de las regiones que ya han sido exploradas o que al parecer no tienen buenas soluciones. Considerando que en las últimas décadas, el desarrollo de métodos no exactos para el diseño de rutas se han focalizado en los algoritmos metaheurísticos: Recocido Simulado, Búsqueda Tabú, Algoritmos Genéticos y Colonia de Hormigas. Este trabajo se centra en un algoritmo de búsqueda tabú y solo por mencionar algunos de los trabajos relacionados con estos algoritmos aplicados al problema de planificación de rutas tenemos a los citados por Soto (2008) [9].

2 Problema del agente viajero

Los problemas más conocidos en el ámbito de la planeación de rutas son: el Problema del Agente Viajero Simétrico (PAVS) y el problema de planeación de rutas vehiculares, para los cuales se han desarrollado diversos algoritmos que buscan una solución óptima con el menor costo computacional.

El problema del agente viajero o TSP por sus siglas en inglés (*Travelling Salesmen Problem*) es uno de los problemas más famosos y complejos de las ciencias computacionales y ha sido abordado por varias ramas de la ingeniería y por distintas razones, su principal aplicación es la de rutear desde distintas perspectivas, ya sea un proceso que lleva una secuencia específica o una distribución de carácter logístico en la que intervienen elementos del transporte, buscando la mejor ruta posible con criterios de economía en distancia o en costo. Proveer soluciones contribuye a mejorar tareas y procesos en distintos ámbitos,

científicos e industriales, proponiendo alternativas para el mejor uso de los recursos. Disciplinas que abordan este tema son la investigación de operaciones y la ciencias informáticas como algoritmia y teoría de grafos (López, 2008) [6].

De acuerdo con De los Cobos et al. (2010) [3], se define el problema del agente viajero, dado un entero $n > 0$ y las distancias entre cada par de las n ciudades, estas distancias se da por medio de la matriz (d_{ij}) de dimensión $n \times n$, donde d_{ij} es un entero mayor o igual a cero. Un recorrido es una trayectoria que visita todas las ciudades exactamente una vez. El problema consiste en encontrar un recorrido con longitud total mínima.

Una permutación π cíclica representa un recorrido, si se interpreta $\pi(j)$ como la ciudad después de la ciudad j , $j = 1, 2, \dots, n$. Así, el costo del recorrido es:

$$\sum_{j=1}^n d_{j\pi(j)}.$$

3 El algoritmo determinístico

3.1 La técnica búsqueda tabú

La Búsqueda Tabú (BT) es un método metaheurístico que puede utilizarse para resolver problemas de optimización combinatoria, tales como el problema del vendedor viajero (TSP). La búsqueda tabú utiliza un procedimiento de búsqueda local o por vecindades para moverse iterativamente desde una solución x hacia una solución x^* en la vecindad de x , hasta satisfacer algún criterio de parada. Para poder explorar regiones del espacio de búsqueda que serían dejadas de lado por el procedimiento de búsqueda local, la búsqueda tabú modifica la estructura de vecinos para cada solución a medida que la búsqueda progresá. Las soluciones admitidas para $N^*(x)$, el nuevo vecindario, son determinadas mediante el uso de estructuras de memoria. La búsqueda entonces progresá moviéndose iterativamente de una solución x hacia una solución x^* en $N^*(x)$ (Glover, 2003) [5].

Quizás la estructura de memoria más importante usada para determinar las soluciones permitidas a un $N^*(x)$, sea la lista tabú. En su forma más simple, una lista tabú es una memoria de corto plazo que contiene las soluciones que fueron visitadas en el pasado reciente (menos de n iteraciones atrás, donde n es el número de soluciones previas que van a ser almacenadas). La búsqueda tabú excluye las soluciones en la lista tabú de $N^*(x)$. Una variación de la lista tabú prohíbe soluciones que tienen ciertos atributos (i.e., soluciones al problema del viajante de comercio (TSP) que incluyen aristas no deseadas) o prevenir ciertos

movimientos (i.e., un arco que fue agregado a un recorrido del TSP no puede ser eliminado en los siguientes n movimientos). Los atributos seleccionados de las soluciones recientemente visitadas son denominados “tabú-activos”. Las posibles soluciones que contengan elementos tabú-activos son “tabú”.

Para mantener el estado de los atributos del movimiento que componen restricciones tabú, y para determinar cuándo son aplicables estas restricciones, Glover y Melián (2003) [5] señalan varias clases básicas de funciones de memoria; funciones de memoria basadas en lo reciente que se especifican mediante los vectores $ComienzoTabu(e)$ y $FinTabu(e)$, donde e varía sobre atributos relevantes a una aplicación particular. Estos vectores identifican, respectivamente, las iteraciones de comienzo y finalización del período tabú para el atributo e , acotando así el período durante el cual e es tabú activo y la memoria basada en la frecuencia que proporciona un tipo de información que complementa la información proporcionada por la memoria basada en lo reciente, ampliando la base para seleccionar movimientos preferidos. Como lo reciente, la frecuencia a menudo está ponderada o descompuesta en subclases teniendo en cuenta las dimensiones de calidad de la solución e influencia del movimiento.

Las listas tabú que contienen atributos que pueden ser más efectivas para algunos dominios, pese a que se presentan nuevos problemas. Cuando sólo un atributo es marcado como tabú, que por lo general ocurre, cuando más de una solución es marcada como tabú. Algunas de estas soluciones, que ahora deben ser evitadas, podrían ser de excelente calidad y no serían visitadas. Para mitigar este problema, se introducen los criterios de aspiración: éstos pueden modificar el estado de tabú de una solución, incluyendo alguna solución antes excluida en el conjunto de soluciones permitidas. Un criterio de aspiración muy utilizado en la literatura, es admitir soluciones que son mejores que la mejor solución conocida al momento (Barros, 2006) [2] y este será el utilizado en la implementación que aquí se considera.

3.2 Algoritmo propuesto

El algoritmo propuesto utiliza la memoria a corto y largo plazo, ya que hace uso de la matriz de frecuencias que expone De los Cobos (2010) [3], que permite la exploración de nuevas regiones liberando de esta manera los movimientos señalados como tabú.

La matriz de frecuencias se construirá al inicio del procedimiento y llevará la historia de los movimientos admisibles y también será la responsable de definir el criterio de la búsqueda en otras regiones. Sin embargo, es importante aclarar que el método generará un conjunto de soluciones y se escogerá aquel cuya función de costo sea mejor. Además, para empezar las iteraciones es necesario

escoger una solución inicial mediante la permutación natural y a partir de ésta se encontrarán las posibles vecindades.

Cada vecindad será explorada con detalle y se establecerá sobre sus elementos las restricciones necesarias para reducir el número de posibilidades en cada iteración. El método de restringir será semejante al expuesto por De los Cobos (2010) [3].

Cuando la función objetivo no mejora después de una cantidad determinada de iteraciones siguiendo el proceso de búsqueda tabú basado en memoria de corto plazo, es conveniente emplear diversificación a partir de la memoria de largo plazo basada en frecuencias. La estrategia de diversificación emplea la memoria de largo plazo, usando un vector ordenado de forma ascendente según el número de cambios de un evento (cuadro 8). De ese vector ordenado se toman los primeros elementos (que corresponden a aquellos sucesos que menos han cambiado) donde el número de eventos y el número de iteraciones con los que se hace diversificación, son parámetros adicionales de búsqueda tabú. La programación de los eventos seleccionados es cambiada de manera forzada, para que se direccione la búsqueda a regiones no visitadas (o poco frecuentadas) (Franco, 2008) [4].

La heurística que se empleará en este estudio, la hemos clasificado como intuitiva y mejora a través de la experiencia almacenada. Tomará como solución inicial, la permutación natural generada por un algoritmo de inserción, la *longituTabu* será el número máximo de prohibiciones que se establecerá sobre la posición de determinado nodo, los *CiclosDiver* es un parámetro que le permite al algoritmo revisar el número de iteraciones en las que el óptimo local no sea ha actualizado, permitiendo de ésta forma la diversificación para explorar nuevas regiones.

Algoritmo 3.1 : EraDeterminístico Inicialización

1. Ingrese: *MatrizDistancia*, *longitudTabú*. *MaxIter*, *CiclosDiver*, *ListaTabú*.
2. Generar la solución inicial(permutación natural) y la distancia $d(x)$.
3. Mientras $\text{Iter} \leq \text{MaxIter}$.
 - 3.1. Actualice *MatrizTabu*.
 - 3.2. Menor *distancia* $\leftarrow d(x)$.
 - 3.3. Generar las $n - 1$ permutaciones.
 - 3.4. Determinar la distancia de cada permutación.
 - 3.5. Clasifique los movimientos en prohibidos y no prohibidos.
 - 3.6. Determinar la menor distancia de los movimientos no prohibidos y seleccione la permutación.

- 3.7. Si todos los movimientos son prohibidos, aplique aspiración. Sino pare.
- 3.8. Actualice $d(x)$.
- 3.9. Si $d(x) < Op(x)$, entonces actualice $Op(x)$. Pase a la instrucción 3.1.
- 3.10. Si en k iteraciones no se actualiza $Op(x)$, entonces diversifique. Pase a la instrucción 3.1.
4. El óptimo encontrado es $Op(x)$.
5. Fin.

Este método funciona de manera iterativa. Una vez cumplido el criterio de parada, se necesita evaluar la solución inicial y la resultante, a fin de observar la mejora. Si el valor es inferior al inicial, se considera que se ha encontrado un nuevo valor mínimo y además se almacena en una lista llamada OptimoLocal, nombre dado a la trayectoria mínima y se almacena la lista de nodos en el orden en que se da esta condición (López, 2011) [7].

4 Resultados numéricos

Con el algoritmo expuesto, se hará el tratamiento del Problema de Agente Viajero, aplicando los procesos a los casos particulares mencionados en la sección 3.2. Seguidamente se mostrará en forma resumida, el resultado obtenido en 10 puntos generados aleatoriamente sobre una circunferencia de radio 6 cm (ver Figura 1, que está asociada a la Tabla 1 de distancias). Se sabe que el óptimo global es 36,79 cm.

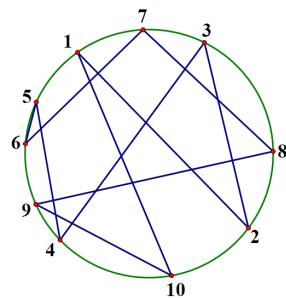


Figura 1: Escenario 1.

d_{ij}	1	2	3	4	5	6	7	8	9	10
1	0	11,85	6,15	9,11	3,13	5,09	3,35	10,60	7,62	11,72
2	11,85	0	9,22	9,13	11,93	11,53	10,86	3,89	10,34	4,37
3	6,15	9,22	0	11,83	8,62	9,94	3,03	6,23	11,29	11,38
4	9,11	9,13	11,83	0	6,77	4,95	10,93	11,16	2,08	5,66
5	3,13	11,93	8,62	6,77	0	2,08	6,24	11,70	4,94	10,64
6	5,09	11,53	9,94	4,95	2,08	0	7,92	11,98	2,97	9,52
7	3,35	10,86	3,03	10,93	6,24	7,92	0	8,62	9,91	11,97
8	10,60	3,89	6,23	11,16	11,70	11,98	8,62	0	11,76	7,76
9	7,62	10,34	11,29	2,08	4,94	2,97	9,91	11,76	0	7,41
10	11,72	4,37	11,38	5,66	10,64	9,52	11,97	7,76	7,41	0

Tabla 1: Matriz de distancias.

Las condiciones iniciales son:

$$\left\{ \begin{array}{l} \text{PermutacionNatural} \longrightarrow (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1) \\ \text{LongInicial} \longrightarrow 89.18 \\ \text{NumerodeIteraciones} \longrightarrow 200 \\ \text{Lon.Tabu} \longrightarrow 4 \\ \text{NumerodeCiclosparadiversificar} \longrightarrow 20 \end{array} \right.$$

El procedimiento determinístico lleva el control de tres tablas dinámicas, la matriz de permutaciones (Tablas 2 y 3), la matriz tabú (Tabla 4) y la matriz de frecuencias (Tabla 5). El manejo dinámico de la matriz tabú y el análisis de la matriz de frecuencia es semejante al dado por De los Cobos et al., 2010 [3].

Iter	Permutaciones										Dist	Tabú
	1	2	3	4	5	6	7	8	9	10		
0												

Tabla 2: Permutación inicial.

En la iteración que se muestra en la Tabla 3, se observa que la permutación mejor admisible para generar el siguiente vecindario, es aquella cuya suma es 78,76 (2, 1, 3, 4, 5, 6, 7, 8, 9, 10, 2). Quedando prohibido el nodo uno en la posición uno y el nodo dos en la posición dos. Así las matrices de memoria (Tabú y Frecuencia) quedan de la forma representada en la Tabla 4 y la Tabla 5, en donde las columnas representan las ciudades y las filas las posiciones.

En la Tabla 4 se ha marcado como prohibido, que la ciudad 1 esté en la posición 1 y que la ciudad 2 esté en la posición 2. Por otro lado, en la Tabla 5 se

Iter	Permutaciones											Dist	Tabú
0	1	2	3	4	5	6	7	8	9	10	1	89.18	
1	2	1	3	4	5	6	7	8	9	10	2	78.76	
	1	3	2	4	5	6	7	8	9	10	1	80.78	
	1	2	4	3	5	6	7	8	9	10	1	90.94	
	1	2	3	5	4	6	7	8	9	10	1	88.84	
	1	2	3	4	6	5	7	8	9	10	1	85.68	
	1	2	3	4	5	7	6	8	9	10	1	96.70	
	1	2	3	4	5	6	8	7	9	10	1	91.39	
	1	2	3	4	5	6	7	9	8	10	1	90.82	
	1	2	3	4	5	6	7	8	10	9	1	81.08	

Tabla 3: Permutaciones de la iteración 1.

	1	2	3	4	5	6	7	8	9	10	
1	4	0	0	0	0	0	0	0	0	0	0
2	0	4	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0

Tabla 4: Matriz Tabú1.

ha marcado que el mejor admisible lo genera al ubicar la ciudad 1 en la posición 2 y la ciudad 2 en la posición 1.

Una vez almacenada esta información, se obtiene la iteración 2, como se muestra en la Tabla 6. Se observa que el mejor admisible es aquel arreglo donde la suma es 73, 41, quedando tabú la permutación cuya suma es 89, 18.

Continuando de esta forma se llega a la iteración 33, cuyo resultado se muestra en la Tabla 7. Hasta este momento, en las últimas 20 iteraciones el óptimo local no se actualiza, por lo que el algoritmo llama a un procedimiento para diversificar la búsqueda.

Es importante señalar que la diversificación no se hace en forma aleatoria, sino mediante el análisis de la información recopilada por la matriz de frecuencias.

	1	2	3	4	5	6	7	8	9	10
1	0	1	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0

Tabla 5: Matriz de Frecuencias 1.

Iter	Permutaciones											Dist	Tabú
1	2	1	3	4	5	6	7	8	9	10	2	78.76	
2	1	2	3	4	5	6	7	8	9	10	1	89.18	1
	2	3	1	4	5	6	7	8	9	10	2	73.41	
	2	1	4	3	5	6	7	8	9	10	2	83.57	
	2	1	3	5	4	6	7	8	9	10	2	78.42	
	2	1	3	4	6	5	7	8	9	10	2	75.26	
	2	1	3	4	5	7	6	8	9	10	2	86.28	
	2	1	3	4	5	6	8	7	9	10	2	80.97	
	2	1	3	4	5	6	7	9	8	10	2	80.40	
	2	1	3	4	5	6	7	8	10	9	2	80.73	

Tabla 6: Iteración 2.

El criterio para ubicar el nodo en cierta posición, será la menor frecuencia registrada. Se observa (ver Tabla 8) que el nodo 1 tiene la menor frecuencia (0) en las posiciones 1, 6, 7, 8, 9 y 10, así que, es buen candidato para ocupar la posición 1. Análogamente, el nodo 2 es buen candidato para la posición 3, el nodo 3 es buen candidato para la posición 1, pero ésta ya está ocupada por lo que se situará en la posición 5, el nodo 4 es buen candidato para la posición 6, el nodo 5 es buen candidato para la posición 1, pero ya está ocupada por lo que se situará en la posición 2, así se continúa hasta generar la nueva permutación de búsqueda que será (1, 5, 2, 8, 3, 4, 7, 6, 9, 10) para continuar con la iteración 35.

El óptimo se encuentra en la iteración 144 en 0,0626 seg., con la distancia de 36,79cm (Tabla 10).

Esta nueva permutación (5, 6, 9, 4, 10, 2, 8, 3, 7, 1, 5), ha transformado el re-

Iter	Permutaciones										Dist	Tabú	
32	4	2	3	7	1	5	6	9	10	8	4	59.24	
33	2	4	3	7	1	5	6	9	10	8	2	54.58	
	4	3	2	7	1	5	6	9	10	8	4	69.77	
	4	2	7	3	1	5	6	9	10	8	4	63.68	
	4	2	3	1	7	5	6	9	10	8	4	65.47	
	4	2	3	7	5	1	6	9	10	8	4	65.14	
	4	2	3	7	1	6	5	9	10	8	4	63.17	1
	4	2	3	7	1	5	9	6	10	8	4	64.21	1
	4	2	3	7	1	5	6	10	9	8	4	69.79	
	4	2	3	7	1	5	6	9	8	10	4	58.09	1

Tabla 7: Iteración 33.

	1	2	3	4	5	6	7	8	9	10	
1	0	3	0	2	0	0	0	0	0	0	
2	1	2	1	3	0	0	0	0	0	0	
3	1	0	3	1	0	0	2	0	0	0	
4	1	0	2	1	0	1	3	0	0	0	
5	1	0	0	1	0	2	1	0	0	0	
6	0	0	0	0	4	3	1	0	0	0	
7	0	0	0	0	5	3	0	0	2	0	
8	0	0	0	0	2	0	0	0	3	0	
9	0	0	0	0	0	0	0	3	0	3	
10	0	0	0	0	0	0	0	3	0	2	

Tabla 8: Matriz de frecuencias de la iteración 33.

corrido dado en la Figura 1, en uno donde no hay intersecciones entre los caminos trazados (Figura 2).

Con el fin de visualizar el papel que desempeña la diversificación, se muestra en la Figura 3, el gráfico para los 200 admisibles generados en cada vecindario.

Es claro ver que, en el momento de encontrar un óptimo local y ubicarse en un valle profundo (De los Cobos et al, 2010) [3], la diversificación permite explorar nuevas regiones para poder mejorar el óptimo local.

4.1 Comparación con TSPLIB

Con base a las observaciones expuestas en la sección anterior, se decidió realizar un segundo experimento en el cual se buscó explorar nuevas poblaciones, cono-

Iter	Permutaciones										Dist	Tabú	
34	1	5	2	8	3	4	7	6	9	10	1	77.96	
35	5	1	2	8	3	4	7	6	9	10	5	76.80	
	1	2	5	8	3	4	7	6	9	10	1	94.49	
	1	5	8	2	3	4	7	6	9	10	1	80.72	
	1	5	2	3	8	4	7	6	9	10	1	82.62	
	1	5	2	8	4	3	7	6	9	10	1	74.99	
	1	5	2	8	3	7	4	6	9	10	1	66.19	
	1	5	2	8	3	4	6	7	9	10	1	78.92	
	1	5	2	8	3	4	7	9	6	10	1	82.06	
	1	5	2	8	3	4	7	6	10	9	1	80.41	

Tabla 9: Iteración después de la diversificación.

Iter	Permutaciones										Dist	Tabú	
143	5	9	6	4	10	2	8	3	7	1	5	42.52	
144	9	5	6	4	10	2	8	3	7	1	9	46.12	
	5	6	9	4	10	2	8	3	7	1	5	36.79	
	5	9	4	6	10	2	8	3	7	1	5	45.49	
	5	9	6	10	4	2	8	3	7	1	5	51.85	
	5	9	6	4	2	10	8	3	7	1	5	49.86	
	5	9	6	4	10	8	2	3	7	1	5	48.90	1
	5	9	6	4	10	2	3	8	7	1	5	53.44	1
	5	9	6	4	10	2	8	7	3	1	5	47.71	1
	5	9	6	4	10	2	8	3	1	7	5	48.75	

Tabla 10: Iteración 144.

ciendo con antelación las soluciones óptimas para cada instancia tratada. Para ello se utilizaron resultados disponibles en TSPLIB (Reinelt, 2004) [8].

Para comparar los resultados obtenidos en los experimentos, se le aplicó la medida de la eficiencia con la ecuación (De los Cobos et al, 2010) [3], con la ecuación:

$$e = 1 - \frac{Z - Z_T}{Z_T}.$$

En la Tabla 11 se observa que, en cualquier caso, la eficiencia está muy cercana a uno, lo que nos indica que el algoritmo propuesto genera resultados satisfactorios y de no ser así, éste da la oportunidad de dirigir la búsqueda hacia nuevas regiones mediante la diversificación o mediante un permutación inicial que no sea la natural.

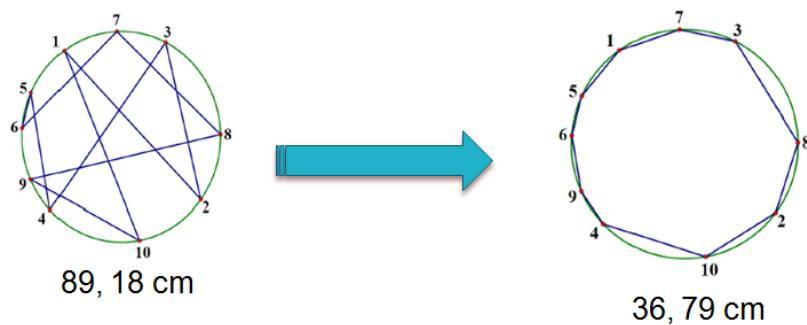
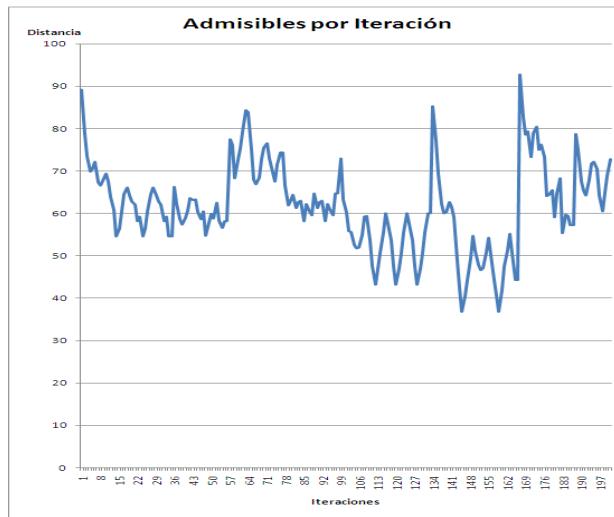


Figura 2: Escenario de transformación.

4.2 El agente viajero en Upala

El propósito del presente trabajo es encontrar soluciones óptimas al problema del agente viajero usando un algoritmo determinístico basado en la técnica Búsqueda Tabú(BT). Para esto, se consideran tres modelos reales, dos para el Agente de la Salud (AS) y otro para el Asesor de Matemáticas (AM), cuyas coordenadas se determinaron con un dispositivo GPS MAP Garmin 60CX a cargo del Ing. Carlos Matarrita y las distancias son por carretera (esto según datos del Instituto de Fomento y Asesoría Municipal (IFAM)).

Se resolvió utilizando la técnica de la búsqueda tabú, cada vecindario fue de tamaño $n - 1$, se utilizó una portatil DELL Vostro 1000 con 1 GB de memoria RAM, procesador Mobile AMD Sempron(tm) Processor 3600 + 2 GHz así como el paquete comercial Visual Basic 6.0 con SP6, encontrándose los resultados dados en la Tabla 12 y los escenarios se muestran en las Figuras 4, 5 y 6.

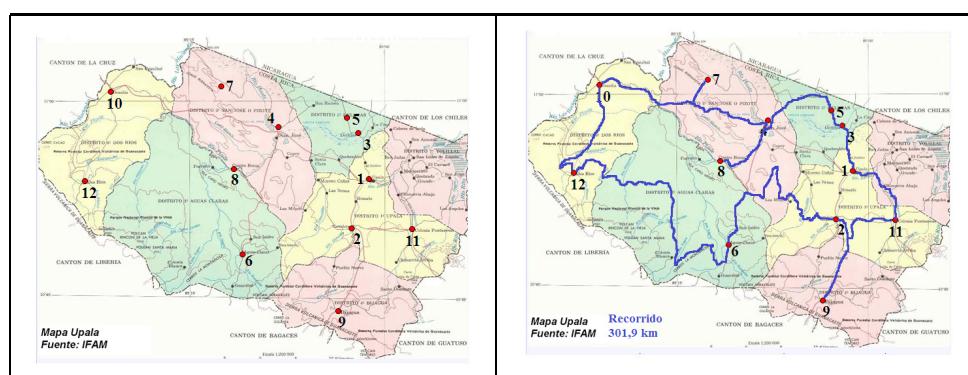
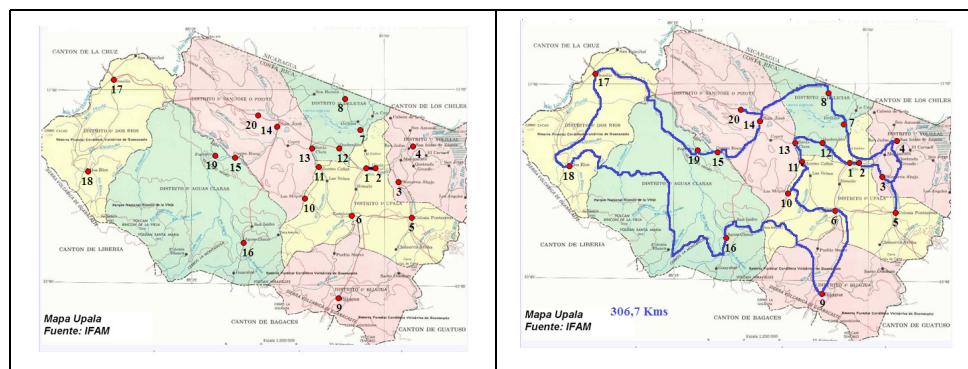
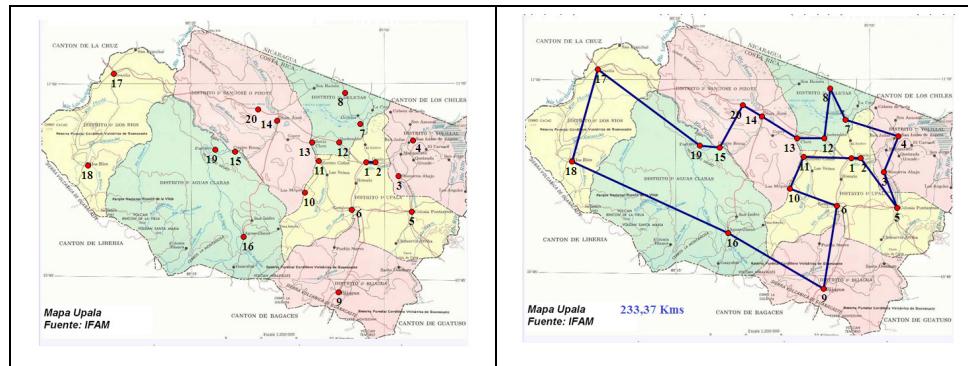
**Figura 3:** Diversificaciones.

Instancia	Long. Tabú	Long. Diver.	Num. Iter.	Long. Inicial	Opt. Lit.	Opt. Enc.	Tiempo (seg.)	e
Angeles6	3	4	5	684,00	93,00	93,00	0,000	1,00
Angeles8	4	15	15	444,00	128,00	128,00	0,005	1,00
C.Carmen11	7	500	3300	437,84	421,21	421,21	0,546	1,00
Dos Bocas11	7	2	2	479,60	464,15	464,15	0,000	1,00
gr17	11	600	4000	7722,00	2085,00	2085,00	3,867	1,00
gr21	11	600	1500	6908,00	2707,00	2863,00	1,094	1,05
gr24	13	600	2700	3057,00	1272,00	1465,00	1,986	1,13
bayg29	11	600	1100	5049,00	1610,00	2063,00	1,161	1,22
baygs29	15	10000	15000	5752,00	2020,00	2901,00	14,892	1,30
fri26	13	40	100	1140,00	937,00	944,00	0,093	1,01
gr48	27	20	1000	6074,00	5046,00	6001,00	7,747	1,16
Berlin52	27	100	1000	8903,00	7542,00	8741,00	8,841	1,14

Tabla 11: Comparación con TSPLIB, De los Cobos y Angeles Baez.

Instancia	LT	LD	Número Iteraciones	Recorrido Inicial	Recorrido Propuesto	Ahorro	Tiempo en seg.
AM12	4	50	2000	635,40 km	301,90 km	333,50 km	0,452
AS20A	11	1000	5000	303 km	233,37 km	69,63 km	2,599
AS20T	11	1000	10000	895,29 km	306,7 km	588,59 km	5,103

Tabla 12: Resultados obtenidos. AS: Agente de la Salud, AM: Asesor de Matemáticas, LT: Longitud Tabú, LD: Longitud Diversificación.



En la instancia para el agente de salud (Figura 4), hubo la necesidad de dar dos soluciones para el PAV, uno para la vía aérea y otro para la vía terrestre. Lo anterior se debió al problema ocasionado por la caída del puente que comunica a Quebradón-Méjico, Méjico-Santa Clara y Delicias-Santa Clara.

En la instancia AM12 (Figura 5) no se necesitaron muchas iteraciones para encontrar un recorrido satisfactorio, considerando que el tamaño de la instancia es mayor a Ciudad del Carmen (De los Cobos, 2010, [3]).

5 Conclusiones

Con este algoritmo se ha comprobado, lo que en la literatura se ha señalado sobre la búsqueda tabú, en donde se menciona que las metaheurísticas pueden encontrar en tiempo razonable, soluciones aproximadas a la óptima, para el problema del agente viajero.

El tipo de codificación para representar el PAV basado en este algoritmo tabú garantiza que los valores que se obtengan para la función objetivo sean de tipo determinístico; por la estructura del mismo, se facilita la generación de soluciones vecinas y el cálculo del valor de la función objetivo.

Se ha determinado el conjunto de parámetros iniciales (longitud tabú, número de iteraciones y número de diversificaciones) que hacen que el método tenga un mejor desempeño computacional para cada uno de los casos de prueba, calibrándolos por ensayo y error.

Se verificó la eficiencia de la técnica búsqueda tabú comparando los resultados obtenidos por el algoritmo, con las mejores respuestas reportadas en la TSPLIB para los casos gr17, gr21, gr24, fri26, bayg29, gr48 y Berlin52, siendo los más representativos los casos gr17 y fri26.

El tipo de codificación propuesta permite implementar otras técnicas metaheurísticas tales como los Algoritmos Genéticos, Colonia de Hormigas o Algoritmos Híbridos (BT y AG) que requieren de una adaptación a los métodos de búsqueda.

La filosofía del algoritmo basado en la técnica búsqueda tabú no es complicada, solamente necesita para su implementación el contexto del problema y la historia del proceso. Además, es flexible y puede ser aplicado a cualquier problema de combinatorio en el que se defina con claridad y precisión los elementos que intervienen.

Los resultados muestran que el algoritmo produce soluciones óptimas o cercanas al óptimo para todas las instancias probadas.

Se recomienda que la longitud tabú esté en el intervalo $[\frac{n}{2} - 3, \frac{n}{2} + 3]$, donde n representa la cantidad de nodos a visitar y que no sobrepasen más de 10 diver-

sificaciones por cada exploración.

De acuerdo a los resultados obtenidos al realizar las pruebas, se propone como parte de una futura investigación, emplear técnicas enfocadas a la diversificación o la exploración mediante un híbrido, con el ánimo de explorar regiones no analizadas.

Con el fin de escapar de los óptimos locales, se propone implementar permutaciones iniciales no naturales, para así, no alcanzar un valor crítico de iteraciones sin mejorar.

Aunque en este algoritmo, los movimientos tienen una distancia de uno, en futuras líneas de investigación se podrían tomar movimientos con mayores distancias entre dos puntos del ciclo para acelerar la convergencia. Además, utilizar más, las capacidades de la búsqueda tabú, con el propósito de mejorar el desempeño del algoritmo.

Referencias

- [1] Baez, A. (2009) “Problema del agente viajero usando búsqueda tabú”. Proyecto Final, Programación Científica, Universidad Autónoma de Nuevo León, Monterrey. Disponible en: <http://es.scribd.com/doc/41965624/Busqueda-Tabu-Problema-del-agente-viajero-Angels-Baez-Olvera>, consultado el 09-Ago-2012, 11:30 a.m.
- [2] Barros, H.J. (2005) “Optimización de ruteo de vehículos empleando búsqueda tabú”. Memos de Investigación, Ingeniería Civil y Ambiental, Universidad de los Andes, Bogotá.
- [3] De los Cobos, S.; Goddard, J.; Gutiérrez, M.; Martínez, A. (2010) *Búsqueda y Exploración Estocástica*. Universidad Autónoma Metropolitana, México D.F.
- [4] Franco, J.; Toro, E.; Gallego, R. (2008) “Problema de asignación óptima de los salones resuelto con búsqueda tabú”, *Revista Ingeniería & Desarrollo* **24**: 149–175.
- [5] Glover, F.; Melián, B. (2003) “Búsqueda tabú”, *Revista Iberoamericana de Inteligencia Artificial* **19**: 29–48.
- [6] López, C.A.; Mendoza, J.A.; Cuartas, E. (2008) “Algoritmo para la exploración de todos los valores posibles en el problema del agente viajero (TSP)”, *Scientia et Technica* **14**(39): 399–403.

- [7] López, E. (2011) *El Agente Viajero: Un Algoritmo Determinístico*. Tesis de Licenciatura en Matemática, Universidad Nacional, Heredia, Costa Rica.
- [8] Reinelt, G. (2004) “TSPLIB, Travelling salesman problem”, Universität Heidelberg, en: <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95>.
- [9] Soto, D.; Soto, W.; Pinzón, Y. (2008) “Una metaheurística híbrida aplicada a un problema de planificación de rutas”. *Revista Avances en Sistemas e Informática* 5(3): 135–145.