Estudio de Tres Algoritmos Heurísticos para Resolver un Problema de Distribución con Ventanas de Tiempo: Sistema por Colonia de Hormigas, Búsqueda Tabú y Heurístico Constructivo de una Ruta

Manuel González de la Rosa[1], Norma Martínez Urbano[2], Venancio García González[2] y Roberto Alejo Eleuterio[3]

[1]Universidad Autónoma del Estado de México,
Unidad Académica Profesional UAEM Tianguistenco,
Paraje El Tejocote, San Pedro Tlaltizapán, Tianguistenco, México CP 52640

[2,3]Universidad Autónoma del Estado de México,
Centro Universitario UAEM Atlacomulco,
Km 60. Carretera Toluca-Atlacomulco, Atlacomulco, México CP 50450
mgonzalezr@uaemex.mx,[3]ralejoe@uaemex.mx

Resumen. En este trabajo se presenta la aplicación y comparación respecto al número mínimo de rutas construidas, de tres algoritmos heurísticos: sistema por colonia de hormigas, búsqueda tabú y heurístico constructivo de una ruta, para un problema de logística de distribución con ventanas de tiempo, que debe resolver diariamente una empresa embotelladora. La empresa debe enviar producto terminado mediante cargas completas de camión, a sus almacenes de distribución que tienen ventanas de tiempo para la recepción de la carga. Cada almacén tiene asociado una demanda, que puede expresarse en número de cargas y debe ser satisfecha diariamente en su totalidad. La solución al problema es un conjunto de rutas que satisfacen la demanda. Los experimentos realizados en una computadora portátil con problemas de prueba, muestran que el algoritmo sistema por colonia de hormigas proporciona mejores soluciones que los otros dos algoritmos heurísticos.

Palabras clave: Heurísticos, búsqueda tabú, sistema por colonia de hormigas, logística de distribución, optimización combinatoria, NP-Completo.

1. Introducción

En los últimos años, ha cobrado mayor relevancia incrementar la competitividad de las empresas y en particular sus actividades logísticas de distribución. Para que una empresa sea competitiva, ésta debe enfocar correctamente su estrategia logística de distribución para lograr abastecer el producto adecuado, en el momento adecuado, al precio adecuado, en las mejores condiciones y cantidades que el cliente lo solicite. Esta importancia logística, ha motivado a investigadores y compañías privadas, a aplicar técnicas de optimización para desarrollar políticas que buscan incrementar la eficiencia en la planeación de la distribución y lograr ventajas competitivas. Una

M.A Cruz-Chávez, J.C Zavala Díaz(Eds):CICos2009, ISBN:978-607-00-1970-8, pp. 4 - 15, 2009.

de las estrategias utilizadas con mayor frecuencia, es el uso de algoritmos heurísticos para la solución de problemas de optimización combinatoria en el transporte de productos terminados.

El problema en estudio se origina en una empresa embotelladora en la ciudad de Toluca. La empresa debe entregar diariamente producto terminado a 12 almacenes regionales. La demanda en cada almacén se conoce con anticipación, ya que la empresa maneja una política de pre-venta. La demanda se satisface mediante camiones que son cargados en su totalidad, por lo que no es necesario decidir sobre la mezcla de productos ni la consolidación de la carga. El producto a distribuir se empaca en palets, en los almacenes regionales el producto se separa y se distribuye a las tiendas minoristas donde acude el consumidor final, lo que implica decidir políticas de ruteo de vehículos que no forman parte del problema de investigación. Ya que se trata de un problema de distribución de bebidas embotelladas, la demanda cambia día con día y semana tras semana de acuerdo a la estación del año. Cada almacén tiene ventanas de tiempo para la recepción del producto, fuera de estas ventanas no es posible atender la entrega. Las ventanas de tiempo, los tiempos de recorrido de la planta a cada almacén, así como los tiempos de descarga se consideran conocidos y constantes, ya que no cambian de un día para otro. Se requieren 20 minutos para cargar un camión en la planta, y solo puede cargarse uno a la vez, esta condición requiere de una estrategia adecuada de secuenciación. No se realiza ruteo ya que un camión se carga con palets en la planta embotelladora, viaja al almacén, se descarga y regresa a la planta para otra posible entrega. De acuerdo a lo anterior debe construirse un plan de distribución al considerar dos tipos de decisiones:

a) Decidir el conjunto de almacenes que debe atender cada ruta construida, y
b) Decidir el orden en que deben cargarse los camiones en la planta

Al considerar los dos tipos de decisiones, debe analizarse la restricción de ventanas de tiempo. El problema tiene similitudes con diversos problemas de transporte tales como el SDVSP *(Single Depot Vehicle Scheduling Problem)*, el SPPTW (*Shortest Path Problem with Time Windows*), el TSPTW (*Traveling Salesman Problem with Time Windows*). Para resolver el SDVSP se han propuesto diversas estrategias, una de las más exitosas ha sido la propuesta por [4], que utiliza un modelo basado en flujo en redes. Esta estrategia puede implementarse en este problema de investigación al modificar los arcos entre un almacén y otro e incluir el regreso a la planta. El SPPTW ha demostrado tener complejidad computacional NP-difícil [2], y aparece como un subproblema al construir un conjunto de nodos de decisión en el tiempo y formar arcos de espera y arcos de entrega, los arcos de espera tendrán un costo mayor a los arcos de entrega ya que se desea minimizar el tiempo ocioso en cada ruta. El TSPTW se caracteriza por ser una extensión de uno de los problemas combinatorios más estudiados, el TSP *(Traveling Salesman Problem)* ya que el TSPTW incluye ventanas de tiempo, al estudiarlo se aprecia la creciente complejidad computacional que depende de la amplitud de las ventanas de tiempo; las técnicas propuestas de reducción de ventanas de tiempo y eliminación de arcos [3] pueden aplicarse al problema en estudio.

El problema de investigación posee características de secuenciación, manejo de ventanas de tiempo y optimización de recursos limitados y es de complejidad

computacional NP-completo [9]. En investigaciones previas se han diseñado estrategias heurísticas para su solución, entre éstas es posible mencionar el algoritmo heurístico constructivo de una ruta (HC1-R) [9], recocido simulado *(Simulated Annealing)* [7], búsqueda tabú (BT) *(Tabu Search)*[8] y sistema por colonia de hormigas (SPCH) *(Ant Colony System)*[10].

La estrategia de solución descrita por [9], consiste en modelar el problema utilizando intervalos de tiempo de 20 minutos, que corresponden al periodo de carga en la planta. Estos intervalos de tiempo están definidos por un instante de decisión inicial y un instante final. Para un horizonte de planeación de 24 horas se tienen un total de 72 instantes o nodos de decisión. De acuerdo a los tiempos de viaje de ida, el tiempo de descarga, el tiempo de regreso a la planta y las ventanas de tiempo en cada almacén, se calculan el nodo inicial y el nodo de regreso para cada posible entrega. De esta forma se construyen para cada almacén y para cada nodo de decisión los arcos de entrega que representan todas las alternativas factibles para las ventanas de tiempo. Los arcos de entrega tienen duraciones diferentes que dependen de los tiempos de viaje hacia cada almacén. Los arcos de espera tienen una duración fija de 20 minutos. Al considerar los arcos de entrega para todos los almacenes, los arcos de espera y los nodos de decisión, es posible representar el proceso de decisión con el fin de construir una ruta, mediante una red la cual se muestra en la figura 1, note que asociado a cada arco se tiene el costo C^B y los periodos de carga de 20 minutos P^B.

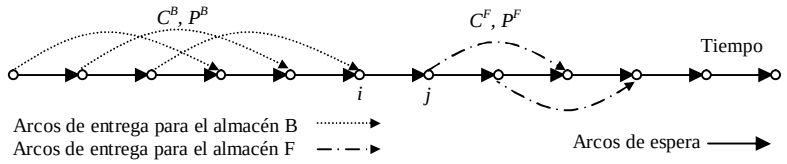

Fig. 1. Grafo que modela la red de decisión para construir una ruta.

Una solución al problema de distribución está formada por, un conjunto de rutas que satisfacen todas y cada una de las demandas asociadas a cada almacén. Una ruta, es una secuencia de entregas que realiza un camión en un intervalo de 24 horas, está descrita por un subconjunto de arcos de entrega y arcos de espera. Al aplicar la estrategia anterior, todos los arcos de entrega son factibles respecto a las restricciones de ventanas de tiempo y la duración del plan de distribución de 24 horas, quedan pendientes por satisfacer las restricciones de demanda para cada almacén, de secuenciación de carga en la planta y secuenciación dentro de cada ruta. La función objetivo del problema es satisfacer la demanda con el número mínimo de rutas.

Todos los posibles arcos de entrega, para cada almacén y para cada nodo de decisión, forman una lista que se muestra parcialmente en la tabla 1. La lista se utiliza como base para formar el espacio de búsqueda y construir las rutas. Cada nodo de decisión está codificado en minutos por lo que el nodo 0 corresponde a las 24:00 horas. En [9] se resuelve el problema utilizando un modelo de flujo en redes con variables binarias, aplicando un algoritmo basado en ramificar y acotar usando CPLEX 8.0 en una computadora PC con procesador Pentium 4 con 512 Mb en memoria RAM. Ya que el problema es NP-completo, declarar la solución óptima

para las instancias difíciles puede consumir varias horas e incluso días de tiempo de cómputo, por tratarse de una planeación que debe realizarse diariamente hace poco practica la implementación de un algoritmo exacto para declarar una solución óptima.

Tabla 1. Lista parcial de los arcos de entrega para cada nodo de envío

Nodo de envío	Nodo de regreso	Almacén
-140	200	9
-120	240	7
-120	220	9
-100	260	7
-100	240	9
-100	160	11
-80	160	6
-80	280	7
-80	140	8
-80	260	9
-80	140	10
-80	180	11
-60	180	6
-60	300	7
-60	160	8
…		

La experiencia computacional con problemas de prueba, construidos al generar la demanda de forma aleatoria, muestran que el algoritmo basado en sistema por colonia de hormigas proporciona mejores soluciones, ya que reporta soluciones con un menor número de rutas.

2. El algoritmo heurístico constructivo de una ruta

El algoritmo heurístico constructivo de una ruta (HC1-R) [9], construye de forma consecutiva a partir de la lista de arcos de entrega, un conjunto de rutas que satisfacen la demanda de cada almacén, al considerar una ruta para cada camión *k*. Para obtener el número mínimo de rutas en una solución, cada ruta debe llenarse con el mayor número de arcos de entrega o con el número mínimo de arcos de espera. El algoritmo HC1-R utiliza una estrategia de descomposición, ya que cada ruta es vista como un problema combinatorio del tipo mochila con una capacidad *P* de 72 períodos. Para construir una ruta, el algoritmo HC1-R incluye solo un arco de entrega a la vez hasta completar la capacidad de la ruta *P*. Los arcos dentro de la ruta deben ser factibles; respecto al orden de carga en la planta y respecto a la secuencia de entregas dentro de la misma ruta. Al incluir un arco de entrega en una ruta, deben resolverse una serie de subproblemas cuyas alternativas se calculan, para cada estado de decisión, y forman al subconjunto *M* de arcos de entrega. Si la demanda de cada almacén no ha sido satisfecha el algoritmo abre otra ruta y repite el proceso hasta

completar el total de entregas D a todos los almacenes. La función objetivo utilizada tiene la forma siguiente:

$$Min \sum_{k \in K} F_k Y_k \tag{1}$$

Donde K es el conjunto de rutas construidas y que forma una solución, F_k es el costo fijo de utilizar una ruta y Y_k es una variable binaria que indica la utilización de la ruta k. El algoritmo HC1-R realiza el siguiente proceso:

Inicio Inicie con demanda satisfecha igual a 0, no se ha construido ninguna ruta

Paso 1. Se incluye un arco de entrega, el subproblema consiste en elegir el arco de entrega que debe incluirse en la ruta para el almacén l con demanda remanente $dr^l > 0$. Se calcula el conjunto de alternativas M y los costos para cada arco de entrega.

Paso 2. Se evalúa cada alternativa al considerar el orden en el cual los arcos de entrega se incluyen en la ruta por la condición de secuenciación en la planta embotelladora y dentro de la ruta. Para evaluar cada alternativa considere los subconjuntos de arcos de entrega ya incluidos en la ruta. Por ejemplo si una ruta k ya incluye envíos a los siguientes almacenes *{a, b, c}*, y se desea incluir el arco *{f}*, los subconjuntos considerados son *{(f,a,b,d), (a,f,b,c), (a,b,f,c), (a,b,c,f)}*. Observe que los subconjuntos crecen rápidamente conforme hay más entregas en la ruta.

Paso 3. Se elige la mejor alternativa considerando el costo mínimo y se actualizan la demanda satisfecha, la demanda remanente y los arcos de entrega en la ruta k. Si no es posible incluir más arcos en la ruta debido a espacio insuficiente o a la secuenciación dentro de la ruta, ésta se cierra, se abre una nueva ruta vacía y se continúa con el paso 1. Si la ruta no se ha cerrado se regresa al paso 1.

Si la demanda total D de todos los almacenes ha sido satisfecha el algoritmo termina y se reportan todas las rutas construidas.

3. El algoritmo búsqueda tabú

El algoritmo búsqueda tabú (BT) [11] es un metaheurístico que está basado en búsqueda local y utiliza una estructura de memoria llamada lista tabú, la cual permite salir de óptimos locales para alcanzar un óptimo global. Este algoritmo es atribuido a Fred Glover quien mencionó en 1986 [1]:

> *"Es mejor una mala decisión basada en información, que una buena decisión al azar, ya que, en un sistema que emplea memoria, una mala elección basada en una estrategia proporcionará claves útiles para continuar la*

> *búsqueda. Una buena elección fruto del azar no proporcionará ninguna información para posteriores acciones".*

Un movimiento es una perturbación que se aplica a la solución actual para generar una nueva solución que comparte algunas de sus propiedades. Al aplicar una serie de movimientos a la solución actual se construye un conjunto de soluciones vecinas, sobre las cuales se realiza la búsqueda local. El algoritmo BT se basa en la prohibición de movimientos que empeoran la solución actual [1]. A partir de los arcos de entrega y los arcos de espera, el algoritmo búsqueda tabú sigue los siguientes pasos:

Inicio. Se construye una solución inicial llamada S_0
Paso 1. Se genera un conjunto de soluciones vecinas $N(S_0)$ mediante la función generadora de vecinos, estas soluciones comparten propiedades similares.
Paso 2. Se evalúa la función objetivo (1) para el conjunto de soluciones vecinas, $N(S_0)$
Paso 3. Se actualiza la mejor solución encontrada
Paso 4. Se actualiza la lista tabú, que está formada por movimientos "prohibidos", ya sea por generar soluciones infactibles o soluciones peores respecto al valor de la función objetivo.
Paso 5. Se aplica el criterio de terminación y en su caso se continua con el paso 1

Para implementar el algoritmo búsqueda tabú se establecen las siguientes estrategias [8], para obtener una solución inicial S_0, se elige el nodo de decisión mínimo k disponible en la planta que sirve para construir la primera ruta, al considerar el nodo de regreso de cada arco de entrega, se elije un nuevo arco de entrega hasta agotar el horizonte de planeación lo cual genera una nueva ruta. Si las demandas de los almacenes aún no están satisfechas, se toma el siguiente nodo de decisión mínimo no utilizado en la posición $k + 1$ para crear la siguiente ruta, repita hasta satisfacer el total de demandas en los almacenes.

El segundo paso es la creación del conjunto de soluciones vecinas $N(S_0)$, la función generadora de vecinos aplica, un movimiento llamado Intercambio de un Arco de la Solución por un Arco Libre (IASPAL), que consiste en intercambiar aleatoriamente un arco de la solución en alguna ruta, por otro que no haya sido utilizado y que no se encuentre dentro de la lista tabú (LT). Este arco se incorpora dentro de una ruta siempre y cuando cumpla con las condiciones de secuencia de carga en planta, secuencia dentro de la ruta, que no haya sido elegido con anterioridad y que al almacén que va dirigido sea el mismo. Para el manejo de la LT, se guardan los últimos t movimientos realizados, lo cual dirige la búsqueda global. Los valores de t son diferentes en cada instancia y oscilan de 3 a 18.

El criterio de intensificación consiste en combinar las mejores rutas dentro de una solución, en especial aquellas que tienen solo uno o dos arcos de entrega. Después de un número determinado de iteraciones, se reporta la mejor solución encontrada hasta el momento. La ejecución del algoritmo BT se realizó con 1000 iteraciones.

4. El algoritmo sistema por colonia de hormigas

Recientes investigaciones sobre algoritmos han tomado como inspiración el comportamiento de insectos que viven en colonias tales como termitas, hormigas, abejas, etc., para resolver problemas combinatorios diversos. Este conocimiento ha sido aplicado en el desarrollo de nuevos algoritmos heurísticos, tales como algoritmos de optimización basados en colonias de hormigas (*Ant Colony Optimization ACO*) [5]. Un requisito para la aplicación de estos algoritmos es que el problema pueda representarse por medio de un grafo formado por nodos y arcos. Cada nodo representa un estado de decisión y los arcos representan alternativas de solución que tienen asociados dos tipos de información, cuya finalidad es guiar el comportamiento de búsqueda de la hormiga artificial. El primer valor η_{ij} es la información heurística que incluye la preferencia de moverse de un nodo *i* a un nodo *j* y no cambia durante todo el proceso. El segundo valor τ_{ij} representa el rastro de feromona que depositan las hormigas en el medio ambiente, modela la deseabilidad de una alternativa y guía el comportamiento de búsqueda de la hormiga artificial. El rastro de feromona es un valor numérico que cambia durante la ejecución del algoritmo y depende de la calidad de las soluciones encontradas. En esta investigación se presenta una variante del algoritmo ACO llamado sistema por colonia de hormigas (SPCH) el cual actualiza la feromona de dos formas: una vez que las *m* hormigas artificiales han terminado su recorrido por el grafo, la cantidad de feromona se actualiza y, posteriormente, se realiza una actualización global de feromona. El valor de *m* varía de acuerdo a la instancia utilizada con el fin de encontrar buenas soluciones de forma rápida, este valor oscila entre 30 y 90 hormigas; los valores cercanos a 30 se utilizan para instancias con demandas menores y los valores cercanos a 90 para instancias con demandas mayores.

Del modelo propuesto por [9], se utilizan los nodos de decisión, los arcos de entrega y los arcos de espera por cada hormiga artificial, vea la figura 1. Asociado a cada arco se tiene un valor inicial que representa el rastro de feromona artificial τ_{ij} sobre el arco *(i, j)*. Cada hormiga construye una ruta y mientras construye su recorrido la hormiga modifica los rastros de feromona sobre los arcos de entrega elegidos, lo que constituye una actualización local de feromona. La hormiga artificial *k*, situada en el nodo *r*, escoge el siguiente nodo *s* que no pertenece a su memoria de arcos ya utilizados M_k aplicando la política de decisión mostrada en la ecuación 2 [6]. Donde τ_{ij} es la cantidad de feromona sobre el arco *(i, j)*; $\eta_{ij} = 1/d_{ij}$; d_{ij} son los periodos de espera en la ruta, ya que las mejores rutas son las que incluyen una mayor cantidad de arcos de entrega, por lo que la función objetivo (1) se modifica en este algoritmo a minimizar los arcos de espera incluidos en una ruta; α y β son parámetros mayores a 1; q es un valor aleatorio [0, 1]; J_k es el conjunto de nodos por ser visitados por la hormiga *k*, y *S* es una variable aleatoria seleccionada de acuerdo a la ecuación 2.

$$s = \begin{cases} \arg\max\left\{ [\tau_{ij}][\eta_{ij}]^{\beta} \right\} & \text{si } q \le q_o \quad \text{explotación} \\ i \in J_k & \\ S \quad \text{en otro caso} & \text{si } q > q_o \quad \text{exploración} \end{cases} \tag{2}$$

Cuando $q < q_0$ la hormiga artificial sigue los rastros de feromona depositados por las hormigas anteriores y cuando $q \geq q_0$ se permite la evaluación de nuevas opciones utilizando el valor de P_{ij} en (3). Una vez que se elige la mejor alternativa, la hormiga viaja a través de este arco y llega a un nuevo nodo de decisión en el cual el proceso se repite hasta que el horizonte de planeación de 24 horas se agota.

$$P_{ij} = \begin{cases} \dfrac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_{l} \tau_{il}^{\alpha} \cdot \eta_{il}^{\beta}} & \text{si (i, j)} \in M_k \\ 0 & \text{en otro caso} \end{cases} \tag{3}$$

De esta forma cada hormiga construye el recorrido que realiza un camión durante un día. La información heurística, representado por η_{ij}, está dada por los arcos de espera en la ruta durante el horizonte de planeación. Si el valor de feromona del arco correspondiente es alto, tendrá más posibilidades P_{ij} de ser elegido. En el proceso de construcción, las hormigas artificiales usan los rastros de feromona dejados por otras hormigas. El rastro artificial de feromona tiene un valor inicial de $\tau_o = 0.0001$ y es un valor numérico entre 0 y 1.

La actualización de feromona se realiza de dos formas: primero una actualización local que se realiza una vez que la hormiga *k* termina de hacer su recorrido por la red de decisión y se actualizan solamente los arcos utilizados por dicha hormiga, la actualización de feromona τ_{ij} se realiza a través de la ecuación 4, ρ es un parámetro con valores entre [0, 1].

$$\tau_{ij} = (1-\rho) \cdot \tau_{ij} + \rho \cdot \tau_o \tag{4}$$

Donde $\tau_0 = 1/ nL_{nn}$, n es el número de nodos y L_{nn} representa el costo de la ruta. La segunda actualización de feromona es una actualización global usando la ecuación 5, esta actualización se realiza una vez que se ha obtenido una solución y considera la hormiga artificial que incluyó la menor cantidad de arcos de espera durante el horizonte de planeación. Solo se actualizan los valores de feromona de los arcos que fueron utilizados por esta hormiga.

$$\tau_{ij} = (1-\delta)\tau_{ij} + \delta \cdot \Delta\tau_{ij} \tag{5}$$

La notación es como sigue: $\Delta\tau_{ij} = 1/L_{gb}$, L_{gb} es el costo de la mejor solución encontrada y δ es el parámetro de evaporación $0< \delta < 1$. El algoritmo se puede dividir en tres procedimientos principales, la construcción de la solución, la actualización de feromona y la evaluación de la solución. Para construir una solución se crean las hormigas artificiales, cada una puede visitar los nodos de la red de decisión eligiendo el siguiente nodo a visitar según la ecuación 2. El procedimiento de actualización de feromona se realiza en los arcos donde se incrementa el depósito de feromona o se reduce por la evaporación lo que evita estancarse en un óptimo local. La evaluación de la solución se realiza para verificar si la solución encontrada es válida y si es mejor que la anterior encontrada, en estos dos últimos procedimientos se busca cumplir con las restricciones del problema.

5. Experiencia computacional

Para evaluar el desempeño de los algoritmos propuestos se utilizaron un total de 38 instancias de prueba que fueron construidas por [9], construidas al variar aleatoriamente la demanda *D* de cada centro de distribución y dejando fijos los demás datos tales como las ventanas de tiempo, los tiempos de viaje, los tiempos de carga y los tiempos de descarga. Las rutas construidas para cada instancia de prueba y por cada algoritmo se muestran en la tabla 2. Ya que el problema de investigación es de complejidad computacional NP-Completo [9], conforme crece la demanda a satisfacer también crece la naturaleza combinatoria del problema, haciendo más difícil encontrar soluciones cercanas al óptimo. Al comparar el algoritmo HC1-R con el algoritmo búsqueda tabú, tenemos que en 15 instancias de prueba el algoritmo búsqueda tabú construye una ruta más; igualando al algoritmo HC1-R en 16 instancias de prueba y mejorando los resultados del HC1-R con dos rutas menos en solo una instancia. Respecto a los algoritmos HC1-R y sistema por colonia de hormigas, en 14 instancias de prueba éste último aventaja al HC1-R al construir una ruta menos y tienen un desempeño similar en 22 instancias de prueba. En general al considerar las 38 instancias de prueba el algoritmo sistema por colonia de hormigas construye un total de 327 rutas, HC1-R construye un total de 345 rutas y el algoritmo búsqueda tabú 368. Los algoritmos fueron programados en lenguaje de programación C++ en una computadora portátil con procesador AMD Sempron a 1.4 Ghz, 1Gb RAM. La ejecución de cada algoritmo se realizó bajo el sistema operativo Windows XP sin otras aplicaciones abiertas. Los tiempos reales de ejecución oscilan entre 2 y 4 minutos para cada instancia. Una comparación gráfica de los algoritmos respecto al número de rutas construidas se muestra en la figura 2, de la cual se puede apreciar un comportamiento similar de los tres algoritmos para demandas menores a 38.

Tabla 2. Rutas construidas por cada algoritmo

Demanda	Número de rutas			Demanda	Número de rutas		
	HC1-R	BT	SPCH		HC1-R	BT	SPCH
20	5	5	4	41	9	10	8
20	5	5	4	41	8	9	8
23	5	5	5	42	11	11	10
24	5	5	5	44	11	12	10
26	6	6	6	44	10	11	10
29	6	7	6	44	9	10	9
32	7	7	7	44	10	11	9
32	8	8	7	44	9	10	9
33	8	8	7	45	11	11	10
34	8	8	7	46	12	13	11
35	8	8	8	46	10	11	10
36	8	8	7	46	10	11	10
38	9	10	9	47	9	11	9
38	8	10	8	49	10	12	10
39	8	9	8	50	10	12	10
39	9	9	9	51	10	11	10
40	8	9	8	53	15	13	13
40	10	10	9	57	12	14	12
41	10	10	9	64	18	18	16

Para demandas mayores el algoritmo sistema por colonia de hormigas construye un menor número de rutas, por lo que es posible concluir que proporciona un mejor desempeño.

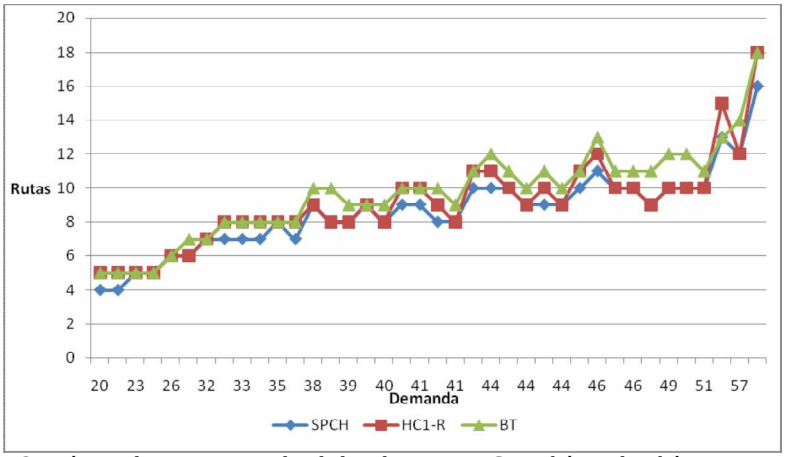

Fig. 2. Número de rutas construidas de los algoritmos HC1-R, búsqueda tabú y sistema por colonia de hormigas, para instancias de prueba con demanda creciente.

6. Conclusiones

Esta investigación compara los resultados obtenidos por tres algoritmos heurísticos, para un problema de logística de distribución con complejidad computacional NP-Completo. La evidencia experimental muestra, que el algoritmo sistema por colonia de hormigas, proporciona mejores resultados que los algoritmos heurísticos HC1-R y búsqueda tabú. La estrategia seguida por el algoritmo sistema por colonia de hormigas para construir una ruta y resolver el problema en estudio considera nodos de decisión en el tiempo, arcos de actividad y penalizar la inclusión de arcos de espera en la evaluación de alternativas, permite explotar adecuadamente los principios que fundamentan a la familia de algoritmos ACO para encontrar la ruta más corta entre dos nodos. La estrategia fundamental del algoritmo HC1-R, es insertar un arco de entrega adicional en una ruta a la vez, sin embargo la naturaleza combinatoria del problema sugiere construir en forma paralela k rutas a la vez lo que implica un mayor esfuerzo de análisis y evaluación. Esta estrategia de inserción paralela puede aplicarse en la función generadora de vecinos y enriquecer el proceso de generación y búsqueda. El desempeño del algoritmo búsqueda tabú es sensible a la calidad de la solución inicial, la función generadora de vecinos, la longitud de la lista tabú y los criterios de aspiración e intensificación, que proporcionan una vasta riqueza de posibilidades que sugieren una investigación más extensa. Los algoritmos utilizados en esta investigación, además de proporcionar soluciones con el mínimo número de rutas, generan soluciones casi óptimas de forma rápida, con tiempos de ejecución de 2 a 4 minutos, lo que permite proponer oportunamente un plan de distribución al personal de la planta embotelladora. Las estrategias de solución presentadas consideran un horizonte de planeación de 24 horas que puede extenderse a 48 ó 72 horas lo cual hace al problema combinatoriamente más rico y difícil de resolver. El problema en estudio también puede modelarse aplicando optimización de flujo en redes lo cual permite extender aún más la presente investigación.

Referencias

1. Arts, Emile., Karel Lenstra, Jan.: Local Search in Combinatorial Optimization. Princeton University Press. Princeton, New Jersey, United States of America (2003).
2. Desrochers, M. and Soumis, F.: A reoptimization algorithm for the shortest path problem with time windows. European Journal of Operational Research. 35(2), 242-254 (1988).
3. Desrochers, M., Desrosiers, J., and Solomon, M.: A new optimization algorithm for the vehicle routing problem with time windows. Operations Research. 40(2), 342-354. (1992).
4. Desrosiers, J., Dumas, Y., Solomon, M. and Soumis, F.: Time constrained routing and scheduling. Handbooks in Operations Research and Management Science. Volume 8 Network Routing. Ball, M.O., Magnanti, T. L., Monma, C. L., and Nemhauser, G. L., (eds)., pp 35-139 (1995).
5. Dorigo, Marco., Di Caro, Gianni., Gambardella, Luca M.: Ant Algorithms for Discrete Optimization. Artificial Life. 5(2), 137-172 (1999)
6. Dorigo, Marco y Gambardella, Luca M.: Ant colonies for the traveling salesman problem. BioSystems. 43(2), 73-81 (1997).

7. González de la Rosa, Manuel y Cruz Cruz, Rafael.: Implementación del algoritmo heurístico recocido simulado para un problema de optimización combinatoria de logística de distribución. Tesis de Licenciatura. Universidad Autónoma del Estado de México. Centro Universitario Atlacomulco. Atlacomulco, México. (2009).
8. González de la Rosa, Manuel y García González, Venancio.: Implementación del algoritmo heurístico búsqueda tabú para un problema de optimización combinatoria de logística de distribución. Tesis de Licenciatura. Universidad Autónoma del Estado de México. Centro Universitario Atlacomulco. Atlacomulco, México. (2009).
9. González de la Rosa, Manuel y Gaytán Iniestra, Juan.: A Heuristic Algorithm for a delivery planning problem with time Windows and Side Constraints. Tesis doctoral. Instituto Tecnológico y de Estudios Superiores de Monterrey. Toluca, México. (2005).
10. González de la Rosa, Manuel y Martínez Urbano, Norma.: Implementación del algoritmo heurístico sistema por colonia de hormigas para un problema de optimización combinatoria de logística de distribución. Tesis de Licenciatura. Universidad Autónoma del Estado de México. Centro Universitario Atlacomulco. Atlacomulco, México. (2009).
11. Melián, Belén., Moreno, Pérez, José A., Moreno, Vega J. Carlos.: Metaheurísticas: una visión global. Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial. 19, 7-28 (2003).