

**Análisis comparativo de una metaheurística en base a algoritmo genético vs un método de ramificación y corte para un caso de entrega y recolección con restricciones de ventana de horario
(Comparative analysis of a metaheuristic based on a genetic algorithm versus a branch & cut method for a pickup and delivery problem with time windows constraints)**

López, F.

Arca Coca Cola, San Jerónimo # 13 Pte., Mty., N. L., México, fabian.lopez@e-arca.com.mx

Palabras claves: Algoritmos genéticos, logística de ruteo, metaheurísticas, secuenciación

Resumen. En la solución de problemas combinatorios, es importante evaluar el costo-beneficio entre la obtención de soluciones de alta calidad en detrimento de los recursos computacionales requeridos. El problema planteado es para el ruteo de un vehículo con entrega y recolección de producto y con restricciones de ventana de horario. En la práctica, dicho problema requiere ser atendido con instancias de gran escala (nodos ≥ 100). Existe un fuerte porcentaje de ventanas de horario activas ($\geq 90\%$) y con factores de amplitud $\geq 75\%$. El problema es NP-hard y por tal motivo la aplicación de un método de solución exacta para resolverlo en la práctica, está limitado por el tiempo requerido para la actividad de ruteo. Se propone un algoritmo genético especializado, el cual ofrece soluciones de buena calidad (% de optimalidad aceptables) y en tiempos de ejecución computacional que hacen útil su aplicación en la práctica de la logística. Para comprobar la eficacia de la propuesta algorítmica se desarrolla un diseño experimental el cual hará uso de las soluciones óptimas obtenidas mediante un algoritmo de ramificación y corte sin límite de tiempo. Los resultados son favorables.

Key words: Genetic algorithms, routing logistics, metaheuristics, scheduling

Abstract. In an attempt to solve the combinatorics problems, it is important to evaluate the cost-benefit ratio between obtaining solutions of high quality and the loss of the computational resources required. The problem presented is for the routing of a vehicle with pickup and delivery of products with time window constraints. This problem requires instances of great scale (nodes ≥ 100). A strong active time window percentage exists ($\geq 90\%$) with factors of amplitude $\geq 75\%$. The problem is NP-hard and hence, the application of an exact method of solution, is

limited by the time frame required for routing activity. A specialized genetic algorithm is proposed, which offers solutions of high precision and in computational times that makes its practical application useful. An experimental design is developed with good results that makes use of optimum solutions obtained by means of branch and cut algorithm without time limit.

Introducción y planteamiento del problema

El problema de entrega y recolección de producto con restricciones de ventanas de horario (Pickup and Delivery Problem with Time Window, PDP-TW) puede ser visto en dos variantes principales. Se distingue la variante para un solo vehículo SPDP-TW y la de vehículos múltiples MPDP-TW (Savelsberg, 1995). El primer caso se trata de un TSP-TW restrictivo (traveling salesman problem) mientras que el segundo se trata de un VRP-TW restrictivo (vehicle routing problem). Solomon desarrolló 87 instancias de prueba para el VRP-TW, la más grande es de 100 clientes (Solomon, 1984). Hasta 1999 había 17 instancias que aún permanecían sin poder ser resueltas. En la Universidad de Rice en Houston se lograron resolver 10 instancias (Cook & Rich, 1999). Nuestro problema se enfoca en el primer caso (SPDP-TW). La familia de problemas del tipo PDP-TW es más difícil de resolver que la del VRP-TW. La razón de lo anterior es debido a que el primero es una generalización del segundo (Palmgren, 2001). El problema será planteado en: (1) objetivo y (2) condiciones de operación. La modelación matemática puede ser revisada en el anexo A.

I. Objetivo del problema

El objetivo es determinar la ruta óptima para un vehículo de distribución. Una ruta se define como la secuencia de llegada a cada uno de los clientes saliendo a partir de un centro de distribución y regresando al mismo al final de la ruta. Se define una ruta óptima como aquella que logre visitar todos los clientes de tal forma que el costo (o distancia o tiempo) incurrido sea el mínimo posible.

II. Condiciones de operación para el problema

- a) El vehículo tiene una capacidad finita de carga.
- b) El vehículo saliendo del centro de distribución, debe atender a un grupo de clientes geográficamente dispersos y luego regresar al punto de origen.

- c) Se tiene una matriz de costo que define el tiempo o distancia requerida para ir de un cliente a otro.
- d) Cada cliente tiene un requerimiento de volumen de producto a entregar y un volumen a recoger.
- e) La cantidad de tiempo requerido para realizar la entrega y la recolección en cada cliente se considera el mismo por tratarse de cargas de trabajo similares.
- f) La ventana de horario identificada para cada cliente esta definida por una hora de apertura y una hora de cierre la cual puede tener diferente amplitud dependiendo de las características de cada cliente.
- g) La amplitud de la ventana de horario en cada cliente es igual a la diferencia entre la hora de cierre y la hora de apertura para la atención de cada cliente.
- h) La ventana de horario para un cliente puede ser modificada únicamente cuando, en el recorrido de la ruta óptima, existan otros clientes geográficamente cercanos y cuyas ventanas de horario sean diferentes.
- i) La visita a cada cliente debe ser aplicada dentro de la ventana de horario. No es permitido llegar antes de la hora de apertura ni después de la hora de cierre.
- j) De acuerdo a la secuencia de ruteo, se tendrá una descarga de producto a entregar en el cliente y una carga a recoger del cliente. En todo momento deberá de ser respetada la capacidad de carga del vehículo.

Revisión bibliográfica de investigaciones previas

A continuación se expone un resumen bibliográfico de las investigaciones previas desarrolladas por Applegate et al., 1998; Dumas et al., 1995; Eijl Van, 1995 y Lenstra, 1990.

Tabla 1: Estudios de investigación previos.

Investigador	Año	Taxonomía del Problema	Algoritmo	Dimensión	Ventanas de Horario	Desempeño Computacional
Baker	1988	ATSP-TW	Ramificación y Corte (BC)	≤ 50 nodos	Cerradas, Poco Traslape	
Dumas, Desrosiers y Soumis	1993	PDP	Flujo de Redes			
Bruggen, Lenstra y Schuur	1993	PDP-TW	Herrística Lin-Kernigham y Enfriamiento Simulado	≤ 50 nodos	Cerradas, Poco Traslape	Instancias fuera de estas condiciones requerían ≥ 10 Mins
Applegate y Bixby	1994	TSP-TW	Ramificación y Corte (BC)			
Van Eijl	1995	VRP	Ramificación y Corte (BC)	≤ 15 nodos	Abiertas	
Coth y Vigo	1995	PDP-TW	Meta-Heurística Tabu Search	$50 \leq n \leq 100$ nodos		Instancias para $n \geq 100$ nodos requerían ≥ 1 Hora
Dumas y Solomon	1995	VRP-TW	Método de Descomposición de Dantzig-Wolfe	$100 \leq n \leq 200$ nodos	Cerradas, Poco Traslape	Instancias para $100 \leq n \leq 200$: $20 \leq \text{mins} \leq 30$
Balas y Simonetti	1996	TSP-TW	Programación Dinámica	$40 \leq n \leq 50$ nodos	Cerradas, Poco Traslape	≥ 2 Horas
Mingozi, Bianco, y Ricciardelli	1997	TSP-TW	Híbrido: BC y Programación Dinámica	≤ 120 nodos		
Ascheuer, Jünger y Reinelt	2000	ATSP-TW	BC con generación de hiperplanos	$50 \leq n \leq 70$ nodos	Abiertas y con bajo % de incidencia de TW	$10 \leq \text{mins} \leq 20$
Ascheuer, Fischetti, y Grottschel	2001	ATSP-TW	Híbrido: BC con Heurísticas de Intercambio	$70 \leq n \leq 233$ nodos	Cerradas, Poco Traslape	$5 \text{ mins} \leq \text{Tiempo} \leq 5 \text{ hrs}$

El punto de partida será el trabajo presentado por Ascheuer, Fischetti, y Grottschel en el 2001 para el problema del TSP-TW el cual es un problema relativamente cercano en definición al SPDP-TW. Ascheuer, Ascheuer et al. (2001) mencionan en su trabajo que resulta interesante la falta de investigación en los métodos de ramificación y corte (Branch & Cut, BC) para la solución tanto de la versión simétrica (TSP-TW) como la asimétrica del problema (ATSP-TW). Como antecedente en este rubro mencionan a Baker, el cual en la década de los 80's resolvió mediante algoritmos de BC instancias de hasta 50 nodos con ventanas de horario moderadamente cerradas. Matemáticamente hablando, entre más cerradas sean las ventanas de horario más pequeño es el espacio solución a ser explorado al momento de resolver computacionalmente el problema. Las instancias que han sido típicamente experimentadas, están caracterizadas por ventanas de horario con un bajo

porcentaje de traslape (Ascheuer et al., 2001).

Típicamente, la dificultad computacional para la solución de problemas de redes ha sido medida en términos de su tamaño (cantidad de nodos). Sin embargo, la dificultad del problema SPDP-TW depende fundamentalmente de la estructura de las ventanas de horario que se definan. Los resultados experimentales con el TSP-TW hechos por Ascheuer et al. (2001), arrojan que este problema es particularmente difícil de resolver para instancias en las que la cantidad de nodos activos que contengan restricciones de ventana de horario estén por arriba del 50%. En lo referente al tamaño de las instancias experimentadas, solo una de las heurísticas revisadas pudo resolver instancias de más de 69 nodos (Ascheuer et al., 2000).

Ascheuer et al. (2001) experimentan con un método de solución especializado con la finalidad de obtener mejores soluciones. Se experimentaron con instancias de hasta 233 nodos. Para una instancia de 69 nodos se requirió de 5.95 minutos de tiempo computacional. Todas las instancias mayores ocuparon más de 5 horas de solución computacional; excepto una de 127 nodos (Tsitsiklis, 1992). Ascheuer et al. (2001) en base a las experiencias computacionales, concluyen que las instancias del ATSP-TW en el límite de 50 a 70 nodos, pueden ser resueltas hasta la optimalidad por el BC.

Metodología

La propuesta de solución está compuesta por 4 rutinas de pre-procesamiento y una post-procesamiento en relación a la ejecución de la rutina del Algoritmo Genético. A continuación se enumeran las 6 fases:

1. Fase de descomposición topológica de la red basado en un algoritmo SPP (Shortest path problem en inglés): las esquinas topológicas requeridas para la modelación de las restricciones de tránsito en la red son atendidas en una fase de preproceso lo cual contribuye a disminuir la carga computacional durante la fase posterior de optimización. De lo anterior si partimos de un valor de N_1 nodos en una red, obtendríamos un valor N_2 , donde $N_1 \approx 4N_2$.

Algoritmo genético

2. Fase de compresión/clusterización por maximización de afinidad geográfica (vecindad): los N_2 nodos de la red se agrupan para obtener una cantidad reducida de N_3 meta-nodos (*donde: $N_3 < N_2$*). Las estructuras de las ventanas de horario asociadas a los N_2 nodos requieren ser compatibles entre si para poderse agrupar. Se utilizó en el algoritmo un factor de compresión del 50% para la agrupación.
3. Fase de compresión/discriminante por heurística de los “ k ” nodos más cercanos: se eliminan los arcos de la red que tengan mayor costo y por tanto menor probabilidad de aparecer en la solución óptima. Para cada nodo N_3 se mantienen en la red solo los “ k ” arcos de menor costo, donde $k \ll N_3$. Se utilizó un factor discriminante del 20% para evitar dejar fuera del espacio de búsqueda la solución óptima.
4. Fase de generación agresiva de cortes: partiendo de los N_3 meta-nodos, el objetivo es encontrar lo más rápidamente posible una solución factible que cubra las restricciones de ventana de horario y de carga del vehículo. La lógica para generar los cortes, identifica el nodo del tour que tiene la mayor desviación respecto a la ventana de horario y/o la capacidad de carga del vehículo (nodo pivote). Luego se verifican los nodos del tour que pueden ser identificados como “afines” a efecto de re-secuenciar la posición del nodo pivote en el tour. El corte asegura que el nodo pivote “ k ” utilice al menos uno de los arcos que lo conecten a uno de los nodos afines “ j ”.

$$\begin{aligned}
 & \text{sea } I = \{1..N_3\} \text{ (nodos de la red)} \\
 & k \in I \text{ (nodo pivote)} \\
 & j \in I \setminus \{k\} \text{ (nodos afines a } k) \\
 & \sum_{j=1}^m (x_{jk} + x_{kj}) \geq 1 \quad \forall k \in I
 \end{aligned}$$

Los cortes generados se guardan en un pool de restricciones. El procedimiento continúa hasta que se encuentra la primera solución factible al problema.

5. Fase evolutiva: el objetivo es aproximar la solución óptima para la versión

compacta de la red. Mantener en el pool de restricciones un corte innecesariamente equivale a estar en riesgo de dejar fuera del espacio de búsqueda mejores soluciones que la que actualmente se tiene. La experiencia computacional indica que la cantidad de cortes que llegan a acumularse en el "pool" de restricciones es significativo (15-40 cortes). El objetivo es identificar cuáles de los cortes del "pool" son necesarios eliminar. La eliminación de cortes no puede ser visto de forma individual para cada corte, ya que la presencia y/o la eliminación de un corte puede comprometer simultáneamente la presencia y/o la eliminación de otro(s). Identificar cuáles cortes conviene eliminar, debe ser visto cómo un sub-problema combinatorio el cual es apropiado para atenderse mediante estrategias de recombinación evolutiva. Una codificación binaria permite representar la eliminación (0) y la presencia (1) del corte en el pool. El algoritmo genético aplica operadores de selección tipo torneo con un factor de cruzamiento al 50%. El método de reproducción es a través de dos puntos de cruce aleatorios a lo largo de la longitud cromosómica. El factor de mutación es inicializado al 5% y autoajustado en cada generación de acuerdo al % de individuos en la población que sean idénticos genéticamente hablando. Al aumentar el nivel de degeneración en la población, se aplica una curva exponencial de crecimiento en el % de mutación con un límite asintótico al 50%. El factor de elitismo está limitado a un 15% de la población.

6. Fase de descompresión para desagregar la ruta propuesta para la red original: el post-procesamiento tiene el objetivo de traducir la solución obtenida en la red compacta en una que sea topológicamente equivalente a la red original. La primera rutina de desagregación está enfocada en determinar la secuencia óptima en la cual los N_3 meta-nodos se deben desagregar para volver a formar los N_2 nodos obtenidos en la fase 2. Lo anterior se logra a través de un algoritmo de secuenciamiento de baja escala lo cual matemáticamente puede ser resuelto hasta la optimalidad en un tiempo polinomial. La segunda rutina de reconversión, hace uso de la información topológica generada en la fase #1. Su objetivo es sustituir la secuencia del tour definido por los N_2 nodos de acuerdo a la cadena de movimientos cardinales que son requeridos para obtener los N_1 nodos de

la red original.

Objetivo e hipótesis del problema

El objetivo es comprobar si el método propuesto basado en un Algoritmo Genético (AG), es viable para encontrar soluciones razonablemente buenas ($\geq 90\%$ optimalidad) y a un bajo costo computacional (≤ 5 minutos) para resolver el problema del SPDP-TW, resumiendo:

H₀: $\mu_{\text{Grupo experimental}} \leq 90\%$ ($\mu_{\text{Grupo Control}}$)

H_a: $\mu_{\text{Grupo experimental}} > 90\%$ ($\mu_{\text{Grupo Control}}$)

donde μ es la media poblacional correspondiente a los porcentajes de optimalidad alcanzados en la medición del 5to. minuto computacional para instancias de tamaño $100 \leq w \leq 120$, (done w = cantidad de nodos en la red). La comprobación de la hipótesis se hará a través de una prueba de diferencias de medias de "T" de Student:

$$t = \frac{\bar{D} - \mu_D}{\frac{S_D}{\sqrt{n}}}, \text{ donde: } \bar{D} = \frac{\sum_{i=1}^n D_i}{n}, \text{ y } S_D = \sqrt{\frac{\sum_{i=1}^n (D_i - \bar{D})^2}{n-1}}$$

Justificación científica y relevancia social

- El problema planteado es de naturaleza combinatoria y está catalogado como NP-Hard (Tsitsiklis, 1992).
- En relación al área de aplicación, la variante menos investigada del SPDP-TW es la que tiene que ver con la distribución física de producto (Mitrovic, 1998).
- Las instancias del problema de investigación encontradas en la práctica de las empresas son arriba de 70 clientes por ruta y con ventanas de horario anchas (anchura $\geq 50\%$). Esta propiedad ocasiona que el problema tenga un espacio solución grande. Esta característica no es enfrentada en la investigación de Ascheuer et al., 2001.
- Los algoritmos de solución exacta (BC), para solucionar instancias como

las antes descritas, requieren un tiempo computacional que excede los límites prácticos de la operación de una empresa.

- e) Existe la necesidad de desarrollar algoritmos que puedan ofrecer soluciones que aunque no sean matemáticamente óptimas, si sean razonablemente de calidad ($\geq 90\%$ del óptimo) y que puedan obtenerse en un tiempo computacional práctico (≤ 5 minutos).
- f) A partir de investigaciones de campo en empresas del área metropolitana de Monterrey (López, 2004).
 - Más del 20% de las empresas enfrentan problemas relacionados con el SPDP-TW.
 - Más del 32% de las empresas resuelven el problema a través de la empírica del administrador.
 - Menos del 12% de las empresas del AMM ha utilizado tecnología APS (Advanced Planning Scheduling) para dar tratamiento al problema sin que exista evidencia de éxito.

Variable para la prueba experimental

El porcentaje de optimalidad relativa: será la variable a utilizar para la prueba estadística de la hipótesis. Esta métrica de calidad de la solución se mide con respecto a la solución registrada por el grupo control mediante un algoritmo de Ramificación y Corte (BC). Mientras en el AG se busca medir con que rapidez se mejora una solución, con el Algoritmo BC se obtiene el valor de referencia para medir el desempeño del AG.

Desarrollo experimental

La técnica que será aplicada para la prueba de la hipótesis será a través de un “Diseño de Experimentos”. El objetivo del diseño de experimentos es determinar si acaso al utilizar un determinado tratamiento, se produce alguna mejora o no en el proceso (Fisher, 1971). La medición de la variable “porcentaje de optimalidad relativa” se hará mediante la aplicación de 4 instrumentos experimentales:

- a) Algoritmo de Ramificación y Corte (BC) como método de solución exacta (Grupo Control).

- b) Algoritmo Genético básico #1: motor de optimización “Evolver” © Versión 6.0 de Palisade Corporation.
- c) Algoritmo Genético básico #2: motor de optimización “Solver” © Versión 4.0 de Frontline System.
- d) Algoritmo Genético propuesto #3.

Para los 4 instrumentos descritos, la medición del “porcentaje de optimalidad” se aplicó en 4 momentos sucesivos de tiempo durante la ejecución del experimento: al minuto #3, #5, #8 y #10. El 1er. instrumento basado en el algoritmo BC, estuvo sujeto a un límite de tiempo de 5 horas, ya que su objetivo fue obtener la referencia de optimalidad para cada unidad experimental. El diseño experimental se aplicó para una muestra de 40 instancias las cuales tuvieron una complejidad matemática lo suficientemente intensa como para consumir un mínimo de 10 minutos de esfuerzo computacional para su solución exacta mediante el algoritmo BC.

Tabla 2: Matriz resultante del diseño del experimento.

Estadístico (*m, s*) para la Variable "% de Optimalidad Relativa"

		Instrumentos de Medición a ser comparados			
		Algoritmo BB (Grupo Control)	Algoritmo Genético Básico (Evolver)	Algoritmo Evolutivo Básico (Frontline)	Algoritmo Propuesto
Mediciones a ser aplicadas después de cierto tiempo computacional	3er Minuto	(<i>m11, s11</i>)	(<i>m12, s12</i>)	(<i>m13, s13</i>)	(<i>m14, s14</i>)
	5to Minuto	(<i>m21, s21</i>)	(<i>m22, s22</i>)	(<i>m23, s23</i>)	(<i>m24, s24</i>)
	8avo Minuto	(<i>m31, s31</i>)	(<i>m32, s32</i>)	(<i>m33, s33</i>)	(<i>m34, s34</i>)
	10mo Minuto	(<i>m41, s41</i>)	(<i>m42, s42</i>)	(<i>m43, s43</i>)	(<i>m44, s44</i>)

La matriz experimental se expone en la Tabla # 2 la cual está formada por 16 tratamientos (4 instrumentos algorítmicos y 4 intervalos de tiempo). Sobre la base de 40 unidades experimentales, entonces estamos hablando de un total

de 640 mediciones. Los parámetros estadísticos para la variable “porcentaje de optimalidad relativa” son:

1. Media muestral (***m***).
2. Desviación estándar muestral (***s***).

La prueba estadística utilizada para la comprobación de la hipótesis fue la Prueba “T” de Student. La aplicación de la Prueba “T” en cada estadístico (m_{ij}, s_{ij}) calculará la probabilidad de que el instrumento algorítmico “j” en el intervalo de tiempo “i” obtenga un “% de optimalidad” mayor al 90% “ $P(x > 90\%)$ ”.

Resultados

Las instancias aplicadas en el experimento fueron desarrolladas a partir de redes arriba de 100 nodos ($100 \leq w \leq 120$). El experimento dio tratamiento a unidades experimentales con ventanas de horario activas por arriba del 70% y con una amplitud mínima del 75%. Podemos confirmar que las condiciones de complejidad matemática estuvieron plenamente cubiertas y por tanto la prueba aplicada al AG propuesto es significativo. Para la implementación de los AG's # 2, 3 y 4, los parámetros operativos fueron fijados empíricamente a un mismo valor para hacer comparables sus desempeños y así eliminar la existencia de cualquier fuente de variabilidad no planificada en el experimento. Las condiciones computacionales aplicadas para el experimento fueron:

- a) Sistema operativo “Windows XP ©”.
- b) Procesador computacional INTEL © a 2.4 Ghz.
- c) Memoria de acceso inmediato de 128 Mhz.

A continuación en la Tabla 3 se muestran los valores calculados para el estadístico “T”.

Tabla 3 Matriz de valores calculados para el estadístico T.

Resultados: Matriz de valores calculados para el estadístico T							
Instrumentos de Medición a ser comparados							
		Algoritmo BB (Grupo Control)	Algoritmo Genético Básico (Evolver)	Algoritmo Evolutivo Básico (Frontline)	Algoritmo Propuesto		
					P(x > 90%)	P(x > 92.5%)	P(x > 95%)
					Mediciones a ser aplicadas después de cierto tiempo computacional	3er Minuto	NA
	5to Minuto	-2.4260	-0.1159	-0.3068	3.3130	1.5392	0.1111
	8avo Minuto	-1.2800	0.1616	0.3173	4.8510	4.1050	0.8300
	10mo Minuto	-0.7001	0.4001	0.9030	6.2980	5.5770	1.3277

La comprobación de la hipótesis se tiene al comparar el valor calculado de "T" versus su valor crítico. Para un 95% de confiabilidad la referencia mínima es "1.685". Para un 99% de confiabilidad la referencia mínima sería "2.426". Para cada valor calculado de "T" es posible establecer su correspondiente nivel de significancia. En la Tabla 4 se muestran las probabilidades "P Valor" para el estadístico "T", donde $P(x > \text{opt}\%)$.

Tabla 4: Valores de probabilidad "p valor".

Matriz de Probabilidad "P Valor" para el Estadístico T, $P(x > \text{Opt}\%)$							
Instrumentos de Medición a ser comparados							
		Algoritmo BB (Grupo Control)	Algoritmo Genético Básico (Evolver)	Algoritmo Evolutivo Básico (Frontline)	Algoritmo Propuesto		
					P(x > 90%)	P(x > 92.5%)	P(x > 95%)
					Mediciones a ser aplicadas después de cierto tiempo computacional	3er Minuto	NA
	5to Minuto	<1%	45%	38%	100%	93%	54%
	8avo Minuto	10%	56%	62%	100%	100%	79%
	10mo Minuto	24%	65%	81%	100%	100%	90%

Discusión

- a) El Algoritmo BC invariablemente obtiene sino la solución óptima, si una

- mejor solución que los AG's.
- b) Los tiempos computacionales para el BC, oscilan entre 20 minutos y 5 horas (límite de tiempo) dependiendo de cada unidad experimental tratada.
 - c) Se probaron instancias sencillas (nodos ≤ 70 , nodos con ventanas activas $\leq 60\%$) para revisar el desempeño del BC. El resultado fue favorable con tiempos computacionales abajo de los 3 minutos.
 - d) El AG #3 propuesto obtiene soluciones satisfactorias (\geq del 90% de optimalidad) y en tiempos computacionales razonables ($3 \leq t \leq 5$ minutos).
 - e) En los AG's básicos (#1 y #2), los porcentajes de optimalidad resultan ser significativamente inferiores ya que estos nunca sobrepasan del 90%.
 - f) Aunque el AG #1 evoluciona mucho más rápido que el AG #2, no obstante el AG #2 ofrece mejores porcentajes de optimalidad que el AG #1.
 - g) Aunque en los primeros 3 minutos de ejecución el AG #1 ofrece mejor desempeño que el AG #2. No obstante al avanzar el tiempo computacional, el AG #1 se atasca en un óptimo local degenerando prematuramente la población de organismos.
 - h) El AG #2 sigue mejorando la población y obtiene una mejor solución. La diferencia observada en el desempeño de estos 3 AG's se explica como sigue:
 - El AG #2 obtiene mejores soluciones que el AG #1 debido a la propiedad generacional en su método de reproducción (Goldberg, 1995).
 - El AG #3 (propuesto) obtiene mejores soluciones que los otros dos debido a que aprovecha del AG #2 la propiedad generacional de reproducción y además debido a la explotación de la estructura del problema.

Conclusiones y recomendaciones

Con los resultados expuestos concluimos los siguientes puntos:

- I. El Algoritmo BC (grupo control) explota eficazmente la estructura matemática del problema alcanzando la solución óptima para instancias

comparables a las de Ascheuer et al. (2001).

- II. El Algoritmo BC obtiene la solución óptima para 38 de las 40 instancias que son particularmente difíciles de resolver y en las cuales la investigación se concentra.
- III. La hipótesis de investigación es comprobada: Podemos establecer a un 100% de confiabilidad que el AG propuesto alcanza un porcentaje de optimalidad $\geq 90\%$ en un tiempo computacional ≤ 5 minutos.
- IV. El AG propuesto ofrece soluciones al problema de investigación dentro de un rango de optimalidad aceptable y en tiempos de ejecución computacionales que hacen factible su implementación en la práctica:
 - El AG propuesto alcanza un porcentaje de optimalidad $\geq 90\%$ en un tiempo computacional ≤ 3 minutos a un 99% de confiabilidad.
 - El AG propuesto alcanza un porcentaje de optimalidad $\geq 92.5\%$ en un tiempo computacional ≤ 5 minutos a un 93% de confiabilidad.
 - El AG propuesto alcanza un porcentaje de optimalidad $\geq 92.5\%$ en un tiempo computacional ≤ 8 minutos un 100% de confiabilidad.
- V. Sin embargo, también debemos establecer que el AG propuesto solo puede asegurar un 90% de confiabilidad para alcanzar soluciones $\geq 95\%$ de optimalidad y en tiempos computacionales ≤ 10 minutos.
- VI. La expectativa de solución de los Algoritmos Genéticos analizados, se ve sensiblemente afectada en la medida en la cual se requieran soluciones que se acerquen al óptimo verdadero (exacto).
- VII. El AG propuesto solo ofrece un 54% de confiabilidad cuando se requiere alcanzar un porcentaje de optimalidad $\geq 95\%$ en un tiempo computacional ≤ 5 minutos. Lo anterior se traduce en el experimento al poderse alcanzar una optimalidad igual o mayor del 95% solo en 22 instancias de las 40 unidades experimentales que fueron aplicados para un límite de 5 minutos de tiempo computacional.

El investigador prevé que el desempeño obtenido por ambas propuestas algorítmicas (BC y AG), puedan servir para generar nuevas líneas de investigación para atender:

- Instancias múltiples de los problemas de ruteo tales como el VRP-TW y el MPDP-TW.
- Instancias “no estáticas” de los problemas de ruteo, las cuales son frecuentes encontrar en las empresas en ambientes de despacho

dinámico.

Referencias

- Applegate, D., R. Bixby & V. Chvátal. 1998. On the solution of traveling salesman problems. *Documenta Mathematica Extr. jrositasm@yahoo.com* Vol. ICM III: 45-56.
- Ascheuer, N., M. Fischetti & M. Grottschel. 2001. *Solving ATSP With Time Windows by Branch-and-Cut*. Springer-Verlag, Alemania.
- Ascheuer, N., M. Jünger & G. Reinelt. 2000. A branch & cut algorithm for the asymmetric traveling salesman problem with precedence constraints. *Computational Optimization and Applications*, 17(1): 2-7.
- Cook, W. & J. Rich. 1999. A parallel cutting-plane algorithm for the vehicle routing problem with time windows, *Computational and Applied Mathematics*, p: 5.
- Dumas, Y., J. Desrosiers & M. Solomon. 1995. An algorithm for the traveling salesman problem with time windows. *Operations Research*, 43(2): 23-25.
- Eijl Van, C. 1995. A polyhedral approach to the delivery man problem, Technical Report 95-19. Department of Mathematics and Computer Science, Eindhoven University of Technology, The Neatherlands, pp. 12-14.
- Fisher, R. 1971. *The Design of Experiments*, Hafner Press & Macmillan Publishers, London.
- Goldberg, D. 1995. *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley Pub Co, University of Massachusetts.
- Lenstra, K. 1990. A Variable Depth Approach for the Single-Vehicle Pickup and Delivery Problem with Time Windows, "COSOR No. 90-48", Eindhoven University of Technology.
- López, F. 2004. Aplicación de un AG para un problema de logística de ruteo. Tesis Doctoral. FAPYA, UANL.
- Mitrovic, S. 1998. Pickup and Delivery Problem with Time Windows. Technical Report SFU CMPT TR 1998-12. Toronto.
- Palmgren, M. 2001. A Column Generation Algorithm for the Log Truck Scheduling Problem. Department of Science and Technology (ITN), Linköping University. Norrköping.
- Savelsberg, M. 1995. Local Search in Routing Problem with Time Windows. *Annals of Operations Research*. Rotherdam.
- Solomon, M. 1984. On the worst-case performance of some heuristics for the vehicle routing and scheduling problem with time window constraints. Report 83-05-03. The Wharton School, Colombus.
- Tsitsiklis, J. 1992. Special cases of traveling salesman and repairman problems with time windows. *Networks* No. 22. Detroit.