Estructura Híbrida de Vecindad para el Problema del Árbol de Expansión Mínima (MST)

Beatriz Martínez-Bahena[1], Marco Antonio Cruz-Chávez[2], Ocotlán Díaz-Parra[3]

[1]Postgrado en Ingeniería y Ciencias Aplicadas, [2]CIICAp, [3]FCAeI, Universidad Autónoma del estado de Morelos, Avenida Universidad 1001. Col. Chamilpa, C.P. 62210. Cuernavaca, Morelos, México.

[2]Autor de Correspondencia: mcruz@uaem.mx

{bmartinezb, odiazp} @uaem.mx

Resumen. En este trabajo se presenta un análisis experimental de cinco estructuras de vecindad incluyendo una estructura híbrida, la cual se compone de cuatro estructuras, par aleatorio, dos pares aleatorios, tres pares aleatorios y cuatro pares aleatorios. Se prueba la eficacia y eficiencia para cada una utilizando el problema del árbol de expansión minima. Este análisis identifica a la estructura de vecindad que permita llevar a cabo una mejor exploración y explotación del espacio de soluciones para el problema del árbol de expansión mínima. La estructura híbrida propuesta en este trabajo de investigación demuestra ser la mejor en eficacia a comparación con las otras estructuras, además en eficiencia es competitiva.

Palabras Clave: estructura híbrida, árbol de expansión mínima, estructura de vecindad.

1 Introducción

El problema del árbol de expansión mínima MST (por sus siglas en inglés, Minimum Spanning Tree) es un problema de optimización combinatoria y es uno de los problemas más importantes en el área de computación distribuida y redes de comunicación [11]. Fue formulado por Otakar Borukva en 1926, quien lo planteó para resolver el problema de hallar la forma más económica de distribuir energía eléctrica en el sur de Moravia, Republica Checa [3]. La formulación de este problema ha sido útil para realizar muchas investigaciones en diversos campos como sistemas eléctricos e hidráulicos, transporte, diseño de redes de telecomunicaciones, sistemas computacionales, sistemas telefónicos y en otros problemas de investigación de operaciones, donde se desea optimizar ya sea costos, distancias, longitudes u otra medida entre los puntos de consumo [8]. El problema del MST dentro de la teoría de la complejidad se clasifica como P este tipo de problema es un conjunto de problemas de decisión que puede ser resulto por un algoritmo determinístico en tiempo polinomial [15] [16].

M.A. Cruz-Chávez (Ed): CICos 2011, ISBN. 978-607-00-5091-6. pp. 26 – 38, 2011

De acuerdo a trabajos presentes en la literatura [7]; [2] para la Reducción de los Atributos en la Teoría de Conjuntos en Bruto se demuestra que la implementación de una estructura de vecindad híbrida da buenos resultados en su aplicación, también en el trabajo de investigación de Cruz [6] utilizaron una estructura híbrida para el problema del Agente Viajero obteniendo muy buenos resultados tanto en eficiencia como en eficacia, pero algo importante de mencionar es que se comprobó en el trabajo de investigación de Martínez [13], para el problema de Máquinas en Paralelo no Relacionadas no se obtuvieron buenos resultados con respecto a la estructura de vecindad híbrida.

En este artículo se presenta el análisis de cinco diferentes estructuras de vecindad para el problema del árbol de expansión mínima, este análisis de la implementación de cada estructura de vecindad dentro de un algoritmo de búsqueda local iterada para evaluar su eficiencia y eficacia de cada estructura de vecindad y en base a las pruebas experimentales determinar cuál estructura fue la mejor encontrada. Una vecindad está definida como el conjunto de soluciones cercanas a una solución inicial dada. La parte fundamental de una vecindad radica en su tamaño y estructura [1]. En cuanto al tamaño de la vecindad mientras más grande sea mayor será la calidad de las soluciones óptimas locales, es decir una vecindad amplia produce una heurística más eficaz [4].

La estructura de vecindad híbrida propuesta en este trabajo de investigación, se compone de un grupo de estructuras de vecindad tales como par aleatorio, dos pares aleatorios, tres pares aleatorios y cuatro pares aleatorios, permitiendo alcanzar mejores soluciones en una vecindad, la implementación de esta técnica de vecindad permitirá una mejor explotación del espacio de soluciones. Este tipo de estructura ha sido empleado en diferentes problemas de optimización anteriormente mencionados.

El presente artículo se divide en las siguientes secciones. La sección uno, introducción, La sección dos, define el problema del Árbol de Expansión Mínima, el cuál es utilizado para probar el rendimiento de las estructuras de vecindad utilizadas en esta investigación. La sección tres presenta la estructura de vecindad híbrida que se utiliza en este trabajo, así como el funcionamiento de cada estructura de vecindad empleada. La sección cuatro, explica la búsqueda local iterada. La sección cinco detalla las pruebas experimentales. La sección seis presenta las conclusiones obtenidas en esta investigación.

2 Problema del Árbol de Expansión Mínima

El problema del árbol de expansión mínima en la literatura [17][9] es representado mediante un grafo, el cuál es definido como un grafo no dirigido, conexo y ponderado $G= (V, E)$, donde $V= \{v1, v2…..vn\}$ es un conjunto finito de vértices y $E = \{eij \mid eij = (vi, vj), vi, vj \in V\}$ es un conjunto finito de aristas .Se dice que un grafo es ponderado si en cada arista tiene asociado un número real positivo denotado con $W = \{wij \mid$

$w_{ij} = w(v_i, v_j)$, $w_{ij} > 0$, $v_i, v_j \in V\}$ representando distancia, costo u otra medida por lo cual es un grafo ponderado. El grafo es no dirigido debido a que las aristas no tienen una dirección, un grafo es conexo si todos los vértices están conectados. En la figura 1 se muestra un ejemplo de un grafo no dirigido, conexo y ponderado.

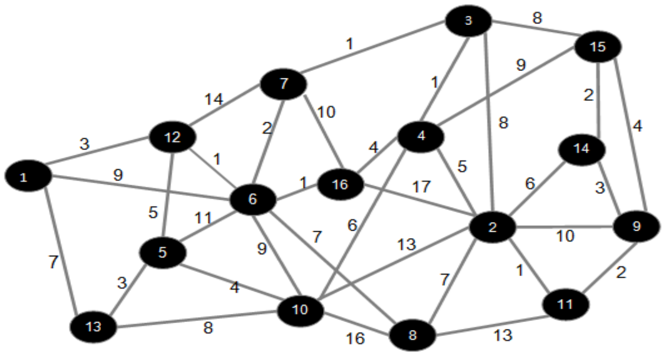

Fig. 1. Grafo no dirigido, conexo con 35 vértices, 16 aristas y con costo aleatorio en cada arista.

Se puede decir en términos de grafos, que para encontrar un árbol de expansión mínimo debe de cumplir ciertas condiciones [18]:

1. Ser un subgrafo de *G* sin ciclos con *n*-1 aristas, donde *n* es el número total de vértices.
2. Ser un subgrafo de *G* donde todos los vértices estén conectados.
3. La suma total de los costos de todas las aristas asociadas al subgrafo sea la mínima.

En la figura 1 se presenta un grafo no dirigido donde su árbol de expansión mínima con las características anteriormente mencionadas se presenta en la figura 2.

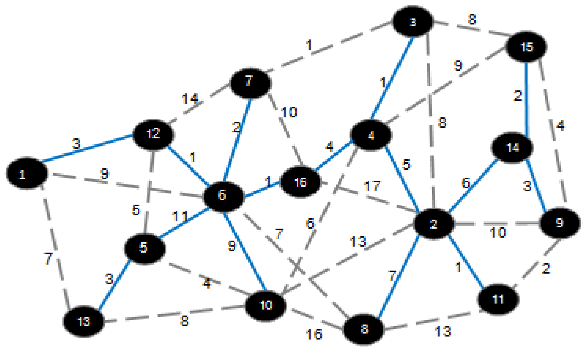

Fig. 2. Ejemplo Árbol de Expansión Mínima

De acuerdo a esto, el problema del árbol de expansión mínima se puede formular mediante un modelo matemático [9] de la siguiente manera:

$$\min c = \sum_{e \in E} w_e x_e \tag{1}$$

$$sujeto \quad a:$$

$$\sum_{e \in E} x_e = n - 1 \tag{2}$$

$$\sum_{e \in (S,S)} x_e \leq |S| - 1 \qquad \forall S \subseteq V \tag{3}$$

$$x_e \in \{0,1\} \qquad \forall e \in E \tag{4}$$

La ecuación (1) es la función objetivo del problema para minimizar la suma total del costo de todas las aristas que conforman al árbol de expansión mínima. El conjunto de restricciones (2), denota que todos los vértices deben estar conectados, por lo cual la suma total de aristas sea igual a n-1, donde n representa el número total de vértices, por lo cual también se cumple la restricción de que no existan ciclos. En el conjunto de restricciones (3) se cumple la restricción de que las aristas de ET no puedan formar ciclos, donde (S, S) denota todas las aristas que van desde un vértice en el conjunto S a otro vértice en el conjunto S. El conjunto de restricciones (4), indica si una arista conecta a un par de vértices o no, si x_e =1 la arista conecta un vértice i con un vértice j, de lo contrario x_e =0.

3 Estructura de Vecindad Híbrida

Las estructuras de vecindad son técnicas utilizadas con el propósito de mejorar una solución, para lo cual es necesario moverse paso a paso desde una solución inicial hacia una solución vecina que proporcione el valor mínimo de la función objetivo [5]. Estas técnicas son utilizadas en problemas de optimización las cuales permiten la mejor exploración del espacio de soluciones mediante su implementación dentro de un algoritmo de búsqueda local.

Una vecindad se define como un conjunto de todas aquellas soluciones que pueden ser alcanzables a partir de una solución inicial s, por medio de un movimiento σ que puede ser una perturbación, inserción o eliminación entre elementos que conforman la solución s para realizar la explotación del espacio de soluciones [15]. El tipo de movimiento define el tipo de estructura y tamaño de la vecindad [12].

De acuerdo a esto, una estructura de vecindad se define como una función N(s) presentada en la ecuación (1).

$$N(s) = \{s' \in S : s \xrightarrow{\sigma} s'\} \qquad (1)$$

Una función de vecindad N(s) específica para cada solución s $\in$ S es un conjunto N(s) $\in$ S, el cual es llamado vecindario de s, esto indica que cada solución s' es un vecino de s si s' $\in$ N(s). S representa el conjunto total de soluciones posibles de una instancia del problema.

A continuación se mencionan los movimientos realizados de cada una de las estructuras simples de vecindad aplicados estos al problema MST. Estas estructuras conforman también a la estructura híbrida de vecindad. Cabe mencionar que las estructuras de vecindad son un aspecto importante, las cuales permiten tener una mejor explotación del espacio de soluciones.

Un Par Aleatorio. Este procedimiento inicia con una solución factible s a partir de la cual se elige un número aleatorio num1 considerado raíz, se elige otro número aleatorio num2 considerado vértice vecino, entre los cuales se realiza una perturbación, dicho movimiento genera un ciclo simple y se elimina una arista que pertenezca al ciclo generado. Si al eliminar esa arista quedan vértices sin conectar, se vuelve a conecta la arista eliminada y se elimina otra arista y así sucesivamente hasta que no queden vértices sin conectar. En la figura 3 se muestra el movimiento realizado por la estructura de un par aleatorio.

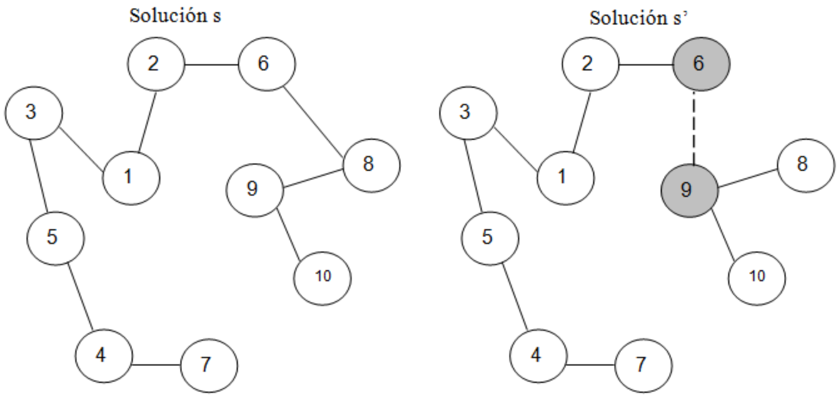

Fig. 3. . Estructura de vecindad un par aleatorio.

Dos Pares Aleatorios. Para esta estructura de vecindad, se lleva a cabo el mismo procedimiento explicado para un par aleatorio, la diferencia está en que requiere generar dos números aleatorios considerados como raíz, los cuáles deben ser distintos y elegir otros dos números aleatorios considerados como vértices vecinos, estos números pueden ser iguales, por lo tanto se eliminan dos aristas que pertenezca a los números generados. La figura 4 muestra el movimiento realizado por la estructura utilizada.

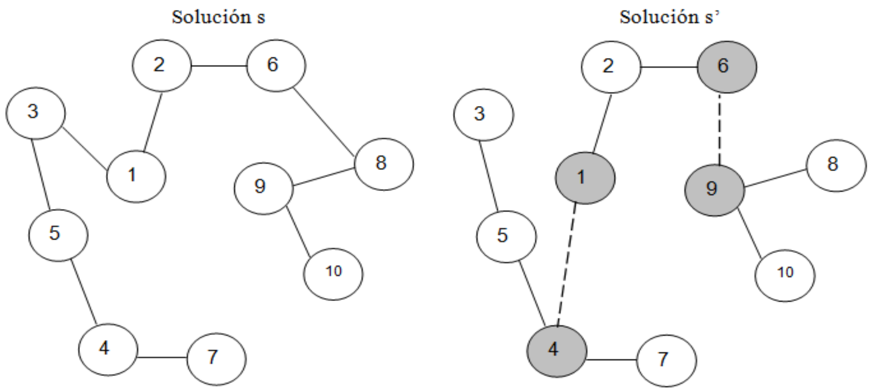

Fig. 4. Estructura de vecindad dos pares aleatorios.

Tres Pares Aleatorios. Una técnica de búsqueda por vecindad que requiera tres pares aleatorios, se lleva a cabo el mismo procedimiento explicado para un par aleatorio, la diferencia está en que requiere generar tres números aleatorios, considerados raíz, los cuáles deben ser distintos, se eligen otros tres números aleatorios considerados como vértices vecinos, estos números pueden ser iguales, por lo tanto se eliminan tres aristas que pertenezcan a los números generados. En la figura 5 se muestra el movimiento realizado por la estructura de vecindad de tres pares aleatorios.

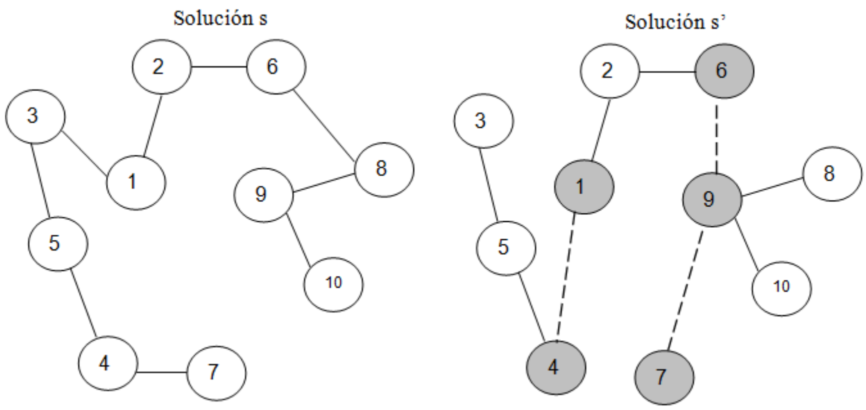

Fig. 5. Estructura de vecindad tres pares aleatorios

***Cuatro Pares aleatorios*.** Una técnica de búsqueda por vecindad que requiera cuatro pares aleatorios, se lleva a cabo el mismo procedimiento explicado para las anteriores estructuras, la diferencia está en que se requiere generar cuatro números aleatorios, considerados raíz, los cuáles deben ser distintos y elegir otros cuatro números aleatorios considerados como vértices vecinos, estos números pueden ser iguales, por lo tanto se eliminan cuatro aristas que pertenezca a los números generados. En figura 6 se observa el movimiento realizado por la estructura.

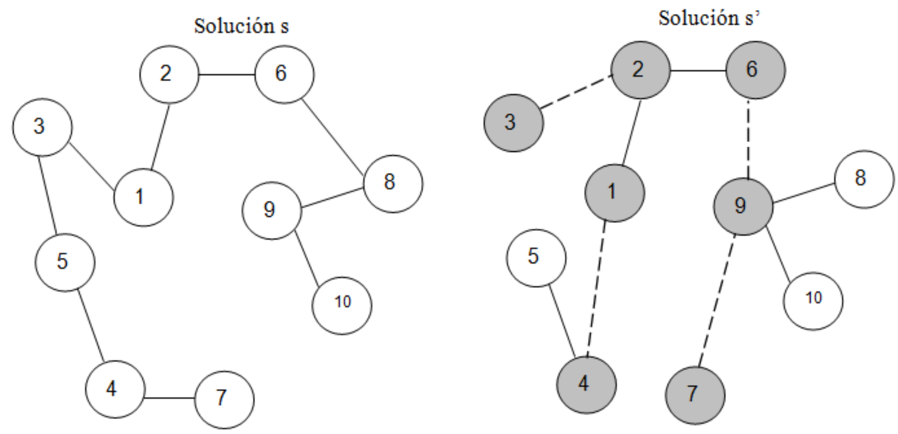

Fig. 6. Estructura de vecindad de cuatro pares aleatorios.

Considerando las funciones de las estructuras de vecindad presentadas anteriormente y su desempeño reportado en la literatura [6]; [7]; [2], se propone el desarrollo de una estructura de vecindad híbrida la cual es una combinación de las estructuras explicadas anteriormente, donde el tipo de movimiento a aplicar se determina aleatoriamente durante la ejecución del algoritmo. En la figura 7 se muestra el funcionamiento de forma general de la estructura de vecindad híbrida.

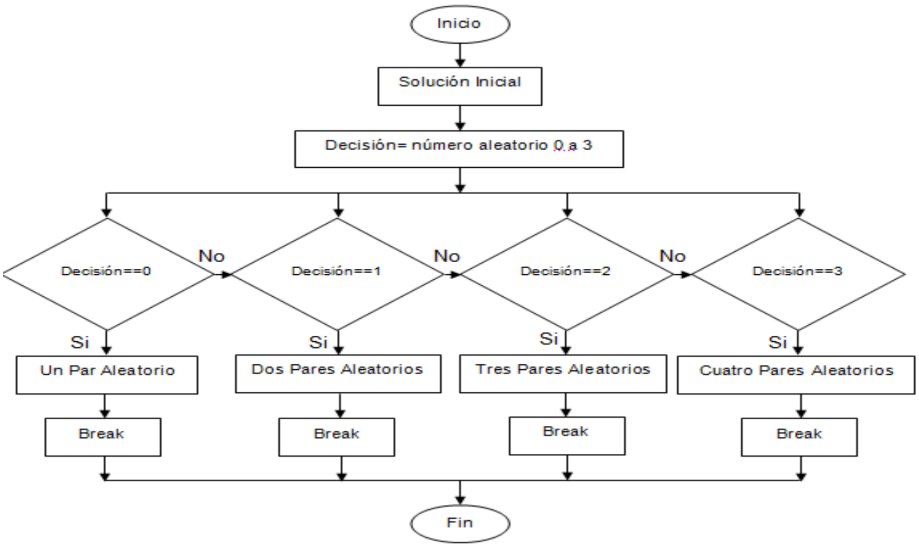

Fig. 7. Diagrama de flujo estructura de vecindad híbrida.

4 Búsqueda Local Iterada

El procedimiento implementado se basa en una búsqueda local iterada, más conocida como Iterated Local Search (ILS), la cual es una heurística que aplica iterativamente un procedimiento de búsqueda local que genera una sucesión de soluciones que se aproxima a un resultado mucho mejor de lo que se esperaría aplicando repetidamente el mismo método de búsqueda local [14]. De acuerdo a los estudios de Hoos y Stützle [10] donde afirman que ILS es una de las metodologías más sencilla y eficaz para evitar quedar atrapados en óptimos locales.

El procedimiento de búsqueda local iterada requiere de una estructura de vecindad y de conocer una función objetivo que se requiera maximizar o minimizar, empieza con una solución cualquiera s y el conjunto de soluciones en N(s), del cual se elige una solución s' a través de un movimiento σ, el tipo de movimiento a realizar para seleccionar un vecino define la estructura de la vecindad que mejore la función objetivo por medio de una proceso estocástico es decir f (s') $\leq$ f (s), si esto se cumple se reemplaza la solución s por la solución s' que la mejore, esto se repite hasta alcanzar el criterio de paro de la búsqueda local y se sigue iterando la búsqueda local hasta que la solución no siga mejorando.se evalúa por medio de un procedimiento (f (s') <= f (CS_ILS)), si esto se cumple se reemplaza la solucion s' por la solución CS_ILS, esto se repite hasta alcanzar el criterio de paro ILS. Después de esto se escoge otra solución s del espacio de soluciones y se evalúa sus vecinos s' que mejore la función vecina s' encontrada en la búsqueda local, se evalúa por medio de un procedimiento (f (s') <= f (CS_ILS)), si esto se cumple se reemplaza la solucion s' por la solución CS_ILS, esto se repite hasta alcanzar el criterio de paro ILS. En la figura 8, se muestra el algoritmo general de búsqueda local iterada.

```
ENTRADA:Estructura de datos;
CS_ILS=M; // donde M tiene un valor muy grande
Hacer
    Generar solución inicial s
    Hacer
        s'=Estructura de vecindad
        Si (f (s') ≤ f (s)) entonces
            s'= mejor solución encontrada
            s = s'
        Fin-si
    Mientras Criterio de Paro LS // LS =Búsqueda Local
    Si (f (s') ≤ f (CS_ILS)) entonces
        CS_ILS= s';
    Fin-si
Mientras Criterio de Paro ILS // ILS=Búsqueda Local Iterada
SALIDA: Solución del MST
```

Fig. 8. Algoritmo general de búsqueda local iterada.

5 Resultados Experimentales

Para realizar las pruebas experimentales correspondientes al algoritmo de Búsqueda Local Iterada, se utilizó el equipo del laboratorio de Optimización, Cluster CIICAp: Procesador Max 3.20GHz Memoria distribuida de 7GB, 7 nodos, Cluster VERACRUZ: Procesador Max 3.20GHz, Memoria distribuida de 57 GB, 14 nodos, Cluster UPEMOR: Procesador Max 2.0GHz, Memoria distribuida de 20 GB, 5 nodos, sistema operativo Linux y compilador GCC. Cabe aclarar que se utilizó esta infraestructura para poder generar las pruebas en un menor tiempo posible para un algoritmo secuencial, no se realizó la paralelización de dicho algoritmo.

Las pruebas experimentales para cada una de las estructuras de vecindad fueron realizadas con el algoritmo de Búsqueda Local Iterada y de acuerdo a los resultados obtenidos se determina cual estructura es la mejor. Las instancias de prueba utilizadas para el problema del árbol de expansión mínima fueron dos, de 100 y 200 vértices, estas instancias fueron generadas de forma aleatoria. Cada estructura de vecindad fue ejecutada 30 veces, el número de pruebas se tomó en base a la estadística como se realizan las pruebas de hipótesis el valor de 30 o más es suficiente ser una muestra representativa para poder demostrar un comportamiento especifico del espacio de soluciones [19], teniendo un número total de 100 iteraciones realizadas durante la ejecución de la búsqueda local Iterada.

5.1 Pruebas Eficacia

En la tabla 1 para el problema MST de 100 vértices con ILS se observa que la estructura con peor eficacia es la de un par aleatorio, y en cuanto a la mejor solución encontrada de las 30 pruebas es la estructura híbrida. La estructura de cuatro pares aleatorios es la mejor en cuanto a la peor solución de mejor calidad, al promedio y a la desviación estándar, la desviación estándar permite conocer que tan dispersas están las soluciones con respecto a la media. Y la estructura híbrida en cuanto a promedio queda en tercer lugar de las cinco y en segundo lugar con respecto a la desviación estándar.

Table 1. Resultados para 100 Vértices. 30 ejecuciones de ILS para cada estructura.

Tipo de Estructura	Mejor Solución	Peor Solución	Promedio	Desv.Est.
Par Aleatorio	3937	5008	4155.2	132.06
Dos Pares Aleatorios	3776	5157	4094.1	134.68
Tres Pares Aleatorios	3806	5144	4037.9	126.48
Cuatro Pares Aleatorios	3773	***4900***	***4013.7***	***109.16***
Híbrida	***3769***	4941	4056.9	122.66

En la tabla 2 para el problema MST de 200 vértices con ILS se observa que la estructura con peor eficacia es la de un par aleatorio, y en cuanto a la mejor solución encontrada de las 30 pruebas es la estructura híbrida también en promedio. La estructura de cuatro pares aleatorios es la mejor en cuanto a la peor solución de mejor calidad. La estructura de tres pares aleatorios es la que tiene la mejor desviación estándar y la estructura híbrida queda en segundo lugar de las cinco.

Tabla 2. Resultados para 200 Vértices. 30 ejecuciones de ILS para cada estructura.

Tipo de Estructura	Mejor Solución	Peor Solución	Promedio	Desv. Est.
Par Aleatorio	8178	10221	8630.7	187.47
Dos Pares Aleatorios	7995	9969	8493.4	200.68
Tres Pares Aleatorios	8162	9912	8515.4	***158.97***
Cuatro Pares Aleatorios	8088	***9776***	8490.8	183.05
Híbrida	***7987***	10095	***8468.5***	174.20

5.2 Pruebas Eficiencia

La figura 9 presenta los resultados obtenidos del tiempo de ejecución promedio del problema MST de 100 y 200 vértices con 30 pruebas para cada instancia aplicando cada estructura de vecindad. En la figura 9 se puede observar que en la instancia de 200 vértices el tiempo de ejecución para cada estructura de vecindad se incrementa considerablemente, debido al aumento en el tamaño del espacio de soluciones. Entonces, podemos decir que una vecindad de mayor tamaño necesariamente requiere mayor tiempo para ser explorada, por ello el tiempo de ejecución para realizar las 30 ejecuciones con 200 vértices aumenta.

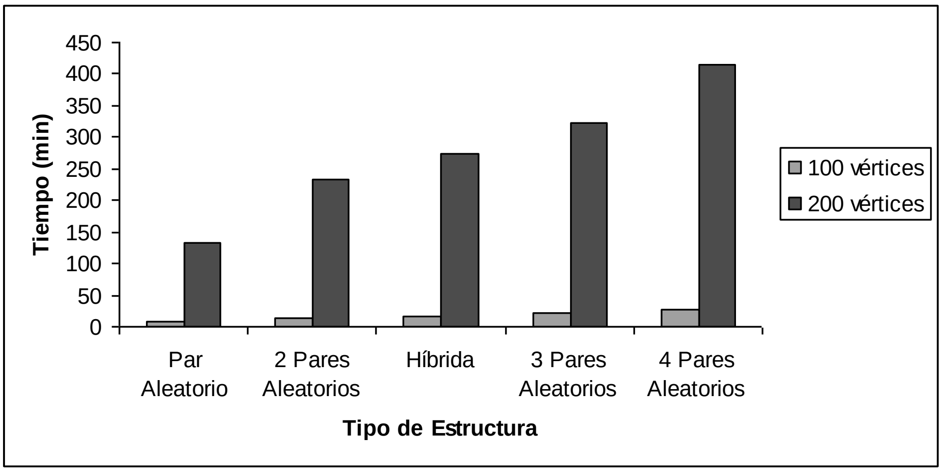

Fig. 9. Resultados de tiempo de ejecución. 100 y 200 vértices para cada estructura.

Se observa en la figura 9 que la estructura de par aleatorio en ambos casos es la mejor en eficiencia, ya que tiene el mejor tiempo de ejecución. La estructura híbrida propuesta en este trabajo de investigación muestra un comportamiento competitivo en ambos casos, es decir no es la que cuenta con la mejor eficiencia, pero tampoco es la de peor eficiencia. Esta estructura se encuentra en un punto intermedio con respecto a las otras estructuras. El comportamiento de la estructura híbrida es lógico, debido a que se compone de las otras cuatro estructuras las cuales se eligen de forma aleatoria. Y la estructura con peor eficiencia es la de cuatro pares aleatorios ya que en ambos casos el tiempo de ejecución es el mayor a comparación con las otras estructuras.

Trabajo Futuro

Implementar la estructura de vecindad híbrida a metaheurísticas como recocido simulado, búsqueda tabú y otras para intentar mejorar el desempeño de dichos algoritmos.

Conclusión

Se concluye que la estructura de vecindad híbrida trabaja con eficacia para el problema del árbol de expansión mínima, comparando los resultados obtenidos en las pruebas experimentales de estructuras de vecindad realizadas para las instancias de 100 y 200. En cuanto a la eficiencia se encontró que la estructura híbrida resulta ser competitiva con respecto al resto de las estructuras analizadas. Se esperaría que al aplicar este tipo de estructuras de vecindad híbridas a metaheurísticas como Recocido Simulado, Búsqueda Tabú, Algoritmos Meméticos, Colonia de Hormigas y para problemas NP-Completos, mejore en cuanto a la eficiencia de dichos algoritmos.

Referencia

[1] Ahuja, R. K., Ergun Ö., Orlin, J. B., Punnen, A. P., Study of Very Large Scale Neighborhood Searching Techniques. On-line: http://mit.ocw.universia.net/15.053/s02/ pdf/s02-lec23b.pdf (2000)

[2] Arajy, Y. Z., Abdullah, S.: Hybrid Variable Neighbourhood Search Algorithm for Attribute Reduction in Rough Set Theory, Data Mining and Optimisation Research Group (DMO) Centre for Artificial Intelligence Technology Universiti Kebangsaan Malaysia, 43600 Bangi Selangor, Malaysia, ISSN: 978-1-4244-8136-1 (2010)

[3] Borůvka, O. "Příspěvek k řešení otázky ekonomické stavby elektrovodních sítí (Contribution to the solution of a problem of economical construction of electrical networks)" (in Czech). Elektronický Obzor 15: 153–154 (1926)

[4] Cruz-Chávez M. A., Díaz-Parra O., Juárez Romero D., Barreto Sedeño E., Zavala Díaz C, Martínez Rangel M. G.: Un Mecanismo de Vecindad con Búsqueda Local y Algoritmo Genético para Problemas de Transporte con Ventanas de Tiempo, CICos 2008, 6to Congreso Internacional de Cómputo en Optimización y Software, ACD, ISBN(e) 978 607 00-0165-9, ISBN (i) 978-970-9750-26-3, pp 23-32, 25-27 Junio, México (2008)

[5] Cruz-Chávez, M. A., Rivera-López, R.: A Local Search Algorithm for a SAT Representation of Scheduling Problems, Lecture Notes in Computer Science, Springer-Verlag Pub., Berlin Heidelberg, ISSN: 0302-9743, Vol.4707, No. 3, pp. 697-709 (2007)

[6] 6. Cruz-Chávez, M.A, Martínez-Oropeza, A., Serna Barquera, S. A.: Neighborhood Hybrid Structure for Discrete Optimization Problems, Electronics, Robotics and Automotive Mechanics Conference, CERMA2010, IEEE Computer Society, ISBN 978-0-7695-4204-1, pp. 108-113 (2010)

[7] Hansen, P., Mladenovic, N.: Variable Neighborhood Search: Principles and Applications, Montréal, Canada, European Journal of Operational Research 130, pp 449-467 (2001)

[8] Hillier, F. S., Lieberman, G. J.: Introducción a la Investigación de Operaciones, Novena Edición, ISBN: 978-607-15-0308-4 (2010)

[9] Hochbaum, D. S., Moreno-Centeno, E.: Network Flows and Graphs. Berkeley IEOR 266. Aug 28 - Sep 11 (2008)

[10] HOOS, H.H.; STÜTZLE, T.: Stochastic local search: foundations and applications, Morgan Kaufmann Publishers, San Francisco, CA (2005)

[11] Maleq, K. M. A.: Distributed Approximation Algorithms for Minimum Spanning Trees and other Related Problems with Applications to Wireless Ad Hoc Networks, Thesis Doctored (2007)

[12] Martínez, M. A. A.: Algoritmo Basado en Búsqueda Tabú para el Cálculo del Índice de Transmisión de un Grafo, Vol. 1, No. 1, pp. 31-39 (2006)

[13] Martínez, O. A.: Solución al Problema de Máquinas en Paralelo no Relacionadas mediante un Algoritmo de Colonia de Hormigas, Tesis de Maestría (2010)

[14] Micheli, M.: Algoritmo de búsqueda local iterativa para la programación de piezas en un sistema flow shop híbrido, Universidad Politécnica de Catalunya, Departamento de Organización de Empresas (2009)

[15] Papadimitriou, C. H., Steiglitz, K.: Combinatorial Optmization. Algorithms and Complexity. ISBN: 0-486-40258-4. USA (1998)

[16] Talbi, El-Ghazali.: Metaheuristics: from design to implementation. ISBN 978-0-470-27858-1 (2009)

[17] S.P.V., S.J. P. A Novel Minimum Spanning Tree Based Clustering Algorithm for Image Mining, European Journal of Scientific Research ISSN 1450-216X Vol.40 No.4 pp.540-546 © EuroJournals Publishing, Inc. (2010)

[18] Cotta, C., Cowling P.: Evolutionary Computation in Combinatorial Optimization 9th European Conference, EvoCOP 2009 Tubingen, Germany (2009)

[19] Walpole, R. E., Myers, R. H., Myers, S. L.: Probabilidad y estadística para ingenieros, 6a.ed. PRENTICE-HALL HISPANOAMERICANA, S. A. ISBN: 970-17-0264-6, pp.752, México (1999)