Soluciones aproximadas para el problema de Triangulación de Peso Mínimo utilizando ACO*

Maria Gisela Dorzán, Edilma Olinda Gagliardi, Mario Guillermo Leguizamón[1]
Gregorio Hernández Peñalver[2]

[1] Facultad de Ciencias Físico Matemáticas y Naturales, Universidad Nacional de San Luis, Argentina
{mgdorzan,oli,legui}@unsl.edu.ar
[2] Facultad de Informática, Universidad Politécnica de Madrid, España
gregorio@fi.upm.es

Resumen Muchos problemas de optimización en configuraciones geométricas son NP-duros por lo que interesa obtener soluciones aproximadas. En este trabajo proponemos la utilización de una técnica metaheurística, Optimización basada en Colonias de Hormigas (Ant Colony Optimization - ACO) para la resolución aproximada del problema de Triangulación de Peso Mínimo (Minimum Weight Triangulation - MWT) para un conjunto de puntos en el plano. Además presentamos los resultados obtenidos de la evaluación experimental realizada, mostrando el rendimiento del algoritmo ACO propuesto.

Key words: Metaheurísticas, Geometría Computacional, Triangulaciones, Triangulación de Peso Mínimo, Optimización basada en Colonias de Hormigas.

1. Introducción

En Geometría Computacional hay numerosos problemas que, o bien son de naturaleza NP-dura, o bien son problemas para los cuales no se conocen soluciones eficientes. De todos modos, resulta de interés encontrar soluciones a tales problemas, aunque las mismas sean aproximadas a las óptimas, por medio de métodos de naturaleza heurística. En particular, es interesante el estudio de problemas de optimización geométrica relacionados con ciertas configuraciones geométricas obtenidas a partir de un conjunto de puntos como son las triangulaciones. En este problema se busca optimizar ciertas propiedades que miden la calidad de las configuraciones: peso, número de apuñalamiento, dilación, factor de carga, etc. Dada la dificultad inherente de dichos problemas, los algoritmos aproximados surgen como candidatos alternativos para su aplicación. Estos, pueden dar soluciones cercanas a las óptimas y pueden ser específicos para un problema tratado o formar parte de una estrategia general aplicable en la resolución de distintos problemas, como lo son las técnicas metaheurísticas.

* Parcialmente subvencionado por Proyecto UPM AL09-PAC-12 y por Proyecto Tecnologías Avanzadas de Bases de Datos N 22/F614, UNSL.

Una metaheurística es un proceso de generación iterativo que guía la búsqueda de soluciones combinando inteligentemente diferentes conceptos de diversos campos como: inteligencia artificial [17], evolución biológica [1], inteligencia colectiva [8], entre otros. Una metaheurística da un marco algorítmico general que puede ser aplicado en problemas de optimización con pocas modificaciones que lo adapten a un problema específico. Estos métodos son simples de implementar y han demostrado ser exitosos en encontrar de forma eficiente buenas soluciones para problemas de optimización NP-duros [14].

En particular, tratamos con la técnica Optimización basada en Colonias de Hormigas (Ant Colony Optimization - ACO), la cual es un proceso distribuido, en el que un conjunto de agentes actúan en forma independiente y cooperan esporádicamente en forma indirecta, para llevar a cabo un objetivo común.

En este artículo presentamos el diseño del algoritmo ACO que obtiene soluciones aproximadas al problema de Triangulación de Peso Mínimo (Minimum Weight Triangulation - MWT). Mostramos los resultados obtenidos de la evaluación experimental realizada utilizando ciertas combinaciones de parámetros.

El trabajo se ha organizado de la siguiente forma. En la Sección 2 presentamos definiciones y antecedentes del problema MWT. En la Sección 3 describimos los aspectos teóricos y generales de ACO y presentamos el algoritmo ACO propuesto para la resolución del problema estudiado. En la Sección 4 presentamos aspectos de la experimentación y mostramos los resultados obtenidos para conjuntos de cuarenta y ochenta puntos, con diversas configuraciones experimentales para el algoritmo ACO-MWT propuesto. Terminamos en la Sección 5 con las conclusiones y trabajos futuros.

2. Triangulación de Peso Mínimo (Minimum Weight Triangulation - MWT)

Sea S un conjunto de puntos en el plano. Una triangulación T de S es un conjunto maximal de aristas cuyos extremos son puntos de S, y para cualquier par de aristas, ellas no se cortan entre sí. El peso de una triangulación T es la suma de las longitudes euclídeas de todas las aristas de T. La triangulación que minimiza esta suma se denomina triangulación de peso mínimo de S. Esta triangulación, que optimiza un criterio bastante simple y natural, ha motivado en los últimos cuarenta años gran número de trabajos, primero en el intento de su obtención de forma exacta y después en la búsqueda de triangulaciones que se aproximaran a ella. Uno de los hitos fundamentales de estos trabajos fue la demostración en 2006 por Mulzer y Rote [15] de que el problema de obtener una triangulación con peso inferior a un valor k es un problema NP-duro. Con esto concluía la búsqueda de un algoritmo polinómico para obtener la triangulación de peso mínimo, pero impulsaba la de algoritmos que obtengan buenas aproximaciones con peso cercano al mínimo.

Uno de los tópicos más importantes en Geometría Computacional es el de triangular un conjunto de puntos y es utilizado en varias aplicaciones como computación gráfica, visualización científica, robótica, visión por computadora

y síntesis de imágenes, como así también en matemática y ciencias naturales. El estudio de la triangulación de peso mínimo comenzó antes de 1970. Lo iniciaron Düppe y Gottschalk [4] quienes propusieron que la estrategia *ávida* (*greedy*), en la que en cada paso se añade la arista más corta posible a la triangulación, llevaría al peso mínimo en la triangulación final obtenida. En 1975, Shamos y Hoey [20] indicaron que la triangulación de Delaunay sería la triangulación de peso mínimo. Sin embargo, en 1977 Lloyd [10] presentó ejemplos en donde ni la triangulación obtenida con la estrategia ávida ni la triangulación de Delaunay son la triangulación de peso mínimo. Además ninguna de estas triangulaciones constituye una buena aproximación de la triangulación de peso mínimo [16]. Describimos a continuación algunos de los resultados obtenidos en cuanto a la búsqueda de buenas aproximaciones. En cuanto a la búsqueda de buenas aproximaciones, Plaisted y Hong [18] propusieron una heurística que aproxima la triangulación de peso mínimo dentro de un factor de $O(\log n)$. Su algoritmo requiere un tiempo de $O(n^2 \log n)$ para el peor caso. Lingas [11], Levcopoulos et al. [13] y Levcopoulos y Krznaric [12] realizaron un profundo estudio de la relación entre la triangulación de peso mínimo y GT (triangulación ávida). Desde el punto de vista de metaheurísticas, la mayoría de los trabajos encontrados presentan soluciones a problemas dentro de la Computación Gráfica. En 1992, Sen y Zheng [21] proponen un algoritmo para aproximar la triangulación de peso mínimo utilizando *Recocido Simulado* pero en la mayoría de las pruebas realizadas obtienen soluciones "cercanas" a las óptimas. Cada configuración se representa utilizando una estructura de datos y el operador de vecindad simplemente realiza un *flip* en una arista aleatoria elegida aleatoriamente de la triangulación actual. En 1993, Wu y Wainwright [22] aproximan la triangulación de peso mínimo utilizando un algoritmo genético donde los operadores de recombinación y de mutación son iguales, es decir, ambos realizan un flip para obtener los individuos de la próxima población. Qin, Wang y Gong [19] también utilizan un algoritmo genético para lo que proponen nuevos operadores de recombinación y mutación adaptados al problema. Capp y Julstrom [3] presentan una nueva codificación ponderada de las triangulaciones para aplicar en un algoritmo genético. En los trabajos anteriores la evaluación experimental es muy limitada y no se extraen conclusiones acerca de la calidad de las soluciones encontradas. En 2001, Kolingerova y Ferko [9] presentan una optimización genética cuyo operador de recombinación es denominado DeWall y el operador de mutación realiza un flip en el individuo seleccionado. La debilidad principal del método es la demanda de tiempo. En [5] y [6] mostramos el trabajo previo referido a aproximaciones sobre MWT utilizando metaheurísticas.

3. Optimización basada en Colonias de Hormigas (Ant Colony Optimization - ACO)

La Optimización basada en Colonias de Hormigas es una metaheurística inspirada en el comportamiento que siguen las hormigas para encontrar los caminos más cortos entre las fuentes de comida y el hormiguero, surgida a partir del

trabajo inicial de Dorigo, Maniezzo y Colorni sobre Sistema de Hormigas (*Ant System*). Los algoritmos ACO son aptos para resolver problemas de optimización discretos y también continuos. Se basan en una colonia de hormigas artificiales, representadas por agentes computacionales simples, que trabajan de manera cooperativa y se comunican entre sí mediante rastros de feromona artificiales [7]. Son esencialmente algoritmos constructivos: en cada iteración del algoritmo, cada hormiga construye una solución al problema recorriendo un grafo de construcción. Cada arista (i, j) del grafo representa los posibles pasos que la hormiga puede dar y tiene asociadas dos fuentes de información que guían el movimiento de la hormiga: i) Información de los rastros de feromona artificiales, que miden la "deseabilidad aprendida" del movimiento, τ_{ij}, de i a j, imitando a la feromona real depositada por las hormigas reales; ii) Información heurística, que mide la preferencia heurística, η_{ij}, de moverse desde el nodo i hasta el nodo j, o sea, de recorrer la arista. Una vez que cada hormiga ha generado una solución se evalúa la misma y se puede depositar una cantidad de feromona en función de la calidad de su solución. Esta información representa el sesgo de búsqueda que influye al resto de las hormigas de la colonia en el futuro. Además, el modo de operación genérico de un algoritmo ACO incluye la evaporación de los rastros de feromona. Dicho procedimiento lo lleva a cabo el entorno y se usa como un mecanismo que evita el estancamiento en la búsqueda y permite que las hormigas busquen y exploren nuevas regiones del espacio.

3.1. Algoritmo ACO General

A continuación presentamos los pasos fundamentales de la estrategia *ACO*.

Algoritmo 1 ACO-General

Inicializar
for $c \in \{1, \ldots, C\}$ **do**
  for $k \in \{1, \ldots, K\}$ **do**
    ConstSolucionk
  end for
  EvaluarSolucion
  ActualizarRastro
end for
RetornarMejorSolucion

Inicializar: Incluye la inicialización de los valores de los parámetros que se consideran en el algoritmo. Entre otros, se deben fijar: el rastro inicial de feromona asociado a cada arista, τ_0, que es un valor positivo pequeño, normalmente el mismo para todas las aristas, el número de hormigas en la colonia, K, y los pesos que definen la proporción en la que afectarán la información heurística y de los rastros de feromonas en la regla de transición probabilística, denominados β y α respectivamente. C es la cantidad de ciclos.

ConstSolucionk: Se inicia con una solución parcial vacía, que se extiende a cada paso añadiéndole un componente de solución factible elegido entre los vecinos de la solución actual. La elección de un vecino factible se realiza de manera probabilística en cada paso de la construcción, dependiendo de la variante ACO utilizada. En este trabajo, el modelo de probabilidad con retroalimentación utilizado en los algoritmos de construcción es:

$$P_{ij} = \begin{cases} \frac{\tau_{ij}^{\alpha}.\eta_{ij}^{\beta}}{\sum_{h \in F(i)} \tau_{ih}^{\alpha}.\eta_{ih}^{\beta}}, j \in F(i); \\ 0, \qquad \text{en otro caso.} \end{cases} \tag{1}$$

- $F(i)$ es el conjunto de puntos factibles para el punto i.
- τ_{ij} es el valor de feromona asociado a la arista (i, j).
- η_{ij} es el valor heurístico asociado a la arista (i, j).
- α y β son parámetros positivos que determinan la importancia relativa de la feromona con respecto a la información heurística.

ActualizarRastro: En este paso se aumenta el nivel de feromona de los caminos prometedores y disminuye el de los caminos no tan buenos. Primero, se reducen todos los valores de feromona por medio del proceso de evaporación. Luego, se incrementa el nivel de feromona al conjunto de soluciones buenas. Se utiliza la siguiente fórmula:

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \Delta\tau_{ij} \tag{2}$$

- $\rho \in (0, 1]$ es el factor de persistencia del rastro.
- $\Delta\tau_{ij} = \sum_{k=1}^{K} \Delta^k \tau_{ij}$ es la acumulación de rastro, proporcional a la calidad de las soluciones.
- $\Delta^k \tau_{ij} = \begin{cases} Q/L_k, \text{ si la hormiga } k \text{ utilizó la arista } (i, j); \\ 0, \qquad \text{en otro caso.} \end{cases}$
- Q es una constante que depende del problema; usualmente es igual a 1.
- L_k representa el valor objetivo de la solución k (aparece en el denominador para problemas de minimización).

La evaporación de feromona evita una convergencia demasiado rápida del algoritmo. Además, esta forma de olvidar permite la exploración de nuevas áreas del espacio de búsqueda. La actualización del rastro de feromona se puede realizar al menos de dos formas: *elitista* o no *elitista*. En el caso elitista, se utiliza la mejor solución encontrada para dar un refuerzo adicional a los niveles de feromona. El no elitista utiliza las soluciones encontradas por todas las hormigas para dar un refuerzo adicional a los niveles de feromona. Nuestra propuesta incluye la aplicación de estas dos formas de actualización del rastro de feromona.

3.2. Algoritmo ACO propuesto para MWT (ACO-MWT)

Considerando el Algoritmo 1 descripto en la sección anterior, presentamos el procedimiento *ConstSolucionk* para el problema MWT descripto en el Algoritmo

2. El resto de los procedimientos en el algoritmo ACO-General son los mismos para ACO-MWT. El proceso *ConstSolucionk* mostrado a continuación es una mejora del algoritmo descripto en [5]. La versión actual es mejor que la anterior ya que el procedimiento que calcula la intersección de aristas se invoca menos cantidad de veces. El proceso *ConstSolucionk* funciona de la siguiente manera: cada hormiga construye una triangulación del conjunto de puntos S, comenzando en un punto de S, seleccionado aleatoriamente. En cada paso, el algoritmo agrega una nueva arista (i, j) a la solución S_k siempre que la arista (i, j) no interseque con las aristas de la solución (parcial) S_k. Si no existe intersección, i es un punto factible para j y viceversa. Si el punto actual no posee puntos factibles, el próximo punto de referencia se selecciona de acuerdo a uno de los siguiente criterios: i) en forma aleatoria; ii) que tenga mayor cantidad de aristas factibles; iii) tenga menor cantidad de aristas factibles.

Algoritmo 2 ConstSolucionk

```
S_k ← ∅
i ← SeleccionarPuntoInicial(S)
while S no está triangulado do
  F_i ← PuntosFactibles(i, S_k)
  if F_i = ∅ then
    i ← SeleccionarPunto(S, S_k)
    F_i ← PuntosFactibles(i, S_k)
  end if
  j ← SeleccionarPuntoProb(F_i)
  if not IntersecaSolucion(i, j, S_k) then
    S_k ← S_k ∪ (i, j)
    i ← j
  end if
  ActualizarFactibles(i, j)
end while
```

A continuación describimos los principales componentes del Algoritmo 2 para comprender mejor su comportamiento:

- *SeleccionarPuntoInicial*(S): retorna un punto $p \in S$, seleccionado aleatoriamente.
- *PuntosFactibles*(i, S_k): retorna un conjunto de puntos $p \in S$, tal que la arista (i, p) puede no intersecar con las aristas de la solución S_k. Note que esta función puede retornar puntos que no son factibles de p.
- *SeleccionarPunto*(S, S_k): retorna un punto $p \in S$, tal que el procedimiento *PuntosFactibles*(p, S_k) retorne al menos un punto. p es seleccionado de acuerdo a uno de los criterios nombrados anteriormente.
- *SeleccionarPuntoProb*(F_i): retorna un punto $j \in F_i$ elegido de acuerdo a la Ecuación 1, donde η_{ij} es $1/d_{ij}$, y d_{ij} es la distancia euclídea entre i y j.
- *IntersecaSolucion*(i, j, S_k): retorna Verdadero si la arista (i, j) interseca al menos una de las aristas de la solución S_k; retorna Falso, en otro caso.

- *ActualizarFactibles*(i, j): actualiza los conjuntos de puntos factibles de los puntos i y j, es decir, los puntos i y j dejan de ser factibles entre sí.

4. Evaluación Experimental

En este trabajo presentamos un modelo ACO para el problema MWT. El algoritmo ACO propuesto está representado por un Ant System (AS), el cual es una instancia particular de la clase de algoritmos ACO. A continuación mostramos la fase inicial de la experimentación, donde obtuvimos resultados preliminares que guiarán la experimentación futura. En esta fase tratamos de encontrar configuraciones adecuadas de parámetros para el algoritmo ACO-MWT para obtener triangulaciones de peso pequeño, cercano al mínimo.

Para toda la evaluación experimental, generamos colecciones de conjuntos de puntos a través de la implementación de un generador de puntos, el cual utiliza diferentes funciones de generación aleatoria de la librería CGAL [2]. Los puntos son no colineales, de distribución uniforme y sus coordenadas están en el rango de 0 a 1000. Generamos 5 colecciones de 10 conjuntos de puntos en el plano con diferente cardinalidad (40, 80, 120, 160 y 200). La denominación para estos conjuntos es la siguiente: para 40 puntos tenemos los conjuntos LD401, LD402,.., LD410; para 80 puntos LD801, LD802,.., LD810; y así sucesivamente para el resto de las cardinalidades. Este tipo de conjuntos de puntos no se encuentran disponibles en investigaciones relacionadas al tema.

El algoritmo ACO-MWT está implementado en C, debido a la facilidad del lenguaje y además utilizamos algunos procedimientos de código abierto disponible en la web. El estudio experimental se realizó utilizando los siguientes valores para los parámetros: (α-β-ρ-*elit*-*criterio*), donde α: 1; y β: 1 y 5; son los parámetros de la Ecuación 1. ρ: 0.1, 0.25 y 0.5; es utilizado en la Ecuación 2. Si *elit* es igual a 1, el rastro se actualiza de manera elitista; en otro caso, la actualización se realiza de forma no elitista. *criterio* se refiere a la manera de selección del punto en el procedimiento *SeleccionarPunto*(S, S_k) en ACO-MWT. Si *criterio* es igual a 1, el punto se elige aleatoriamente; si es igual a 2, se elige el punto que posea más cantidad de puntos factibles; y si es igual a 3, se elige el punto que tenga menor cantidad de puntos factibles. En esta etapa experimental, el parámetro *criterio* es 1 para todos los casos.

Los parámetros considerados en el estudio experimental son: la cantidad de ciclos C es 1000; la cantidad de hormigas K es 50. Para cada configuración dada anteriormente, se realizaron 30 ejecuciones utilizando una semilla diferente en cada una para la generación de números aleatorios. En esta etapa se realizaron ejecuciones para los siguientes conjuntos: LD401, LD402, LD403 y LD404, para los cuales se experimentó con 12 configuraciones de parámetros de acuerdo a las combinaciones posibles de los valores dados anteriormente. Con los resultados obtenidos, calculamos la media, mediana, mejor y varianza considerando la función objetivo (peso).

El trabajo numérico se realizó utilizando el cluster paralelo BACO del Instituto de Física Aplicada de la Universidad Nacional de San Luis, CONICET.

Está compuesto de 60 PCs, cada uno con un procesador Pentium-4 de 3.0 GHz y 90 PCs, cada uno con un procesador Core 2 Quad de 2.4 GHz, bajo el sistema batch de cola CONDOR.

A continuación analizamos el rendimiento del algoritmo ACO-MWT. En las Tablas 1, 2, 3 y 4 mostramos los resultados teniendo en cuenta las cuatro mejores configuraciones de parámetros, es decir aquellas que obtuvieron los pesos más pequeños. Las tres mejores configuraciones son iguales para los conjuntos LD401, LD402 y LD403. Aunque las mejores configuraciones para LD404 no son iguales al resto, podemos inferir que el algoritmo ACO-MWT tiene un comportamiento similar para estos conjuntos. La Tabla 5 es un resumen de las tablas anteriores y muestra que los mejores pesos son obtenidos utilizando configuraciones con β:5, *elit*:0 y ρ entre 0.1 y 0.5, es decir, obtenemos mejores resultados dándole más importancia a la información heurística y realizando la actualización del rastro de una manera no elitista.

La Figura 1 muestra los boxplots de los pesos obtenidos para las 30 ejecuciones (semillas) para LD401, LD402, LD403 y LD404 para las mejores configuraciones. Las medianas son similares en todas la figuras para las tres primeras configuraciones. Si bien los mejores pesos obtenidos son outliers, podemos observar que el 50 % de los valores (valores entre el primer y tercer cuartil) no son tan dispares ni se encuentran muy alejados de los valores mínimo y máximo.

Tabla 1. Resultados para LD401.

Config.	Media	Mediana	Mejor	Varianza
1-5-10-0	5495729,60	5497978,50	5475256,00	6184,00
1-5-50-0	5495148,80	5496132,50	5479181,50	6318,28
1-5-25-0	5496158,40	5496210,75	5485930,50	5205,64
1-1-25-0	5518407,47	5520808,00	5489734,50	14043,18

Tabla 2. Resultados para LD402.

Config.	Media	Mediana	Mejor	Varianza
1-5-10-0	4678068,28	4691654,94	4668377,67	3771,33
1-5-50-0	4678523,50	5251132,76	4671639,77	3717,04
1-5-25-0	4679215,71	4691425,24	4673630,73	3006,15
1-1-10-0	4691305,47	5246345,65	4674216,85	6697,87

Tabla 3. Resultados para LD403.

Config.	Media	Mediana	Mejor	Varianza
1-5-10-0	5497635,73	5498311,25	5481187,50	9448,96
1-5-25-0	5498397,33	5499293,00	5481921,00	7435,30
1-5-50-0	5500451,20	5501121,50	5487679,00	6702,13
1-5-25-1	5527743,47	5530953,25	5492354,00	13851,06

Tabla 4. Resultados para LD404.

Config.	Media	Mediana	Mejor	Varianza
1-1-25-0	5802545,07	5804710,50	5767539,00	14508,18
1-1-50-0	5801372,27	5803196,50	5771443,00	12773,52
1-5-10-0	5796995,20	5798162,75	5774475,50	7757,17
1-1-10-0	5800032,00	5802819,75	5778582,50	12406,87

Tabla 5. Tabla resumida para LD401, LD402, LD403 and LD404.

α	β	ρ	*elit*
1 (100 %)	5 (68.75 %)	0.1 (37.5 %)	0 (93.75 %)
	1 (31.25 %)	0.25 (37.5 %)	1 (6.25 %)
		0.5 (25 %)	

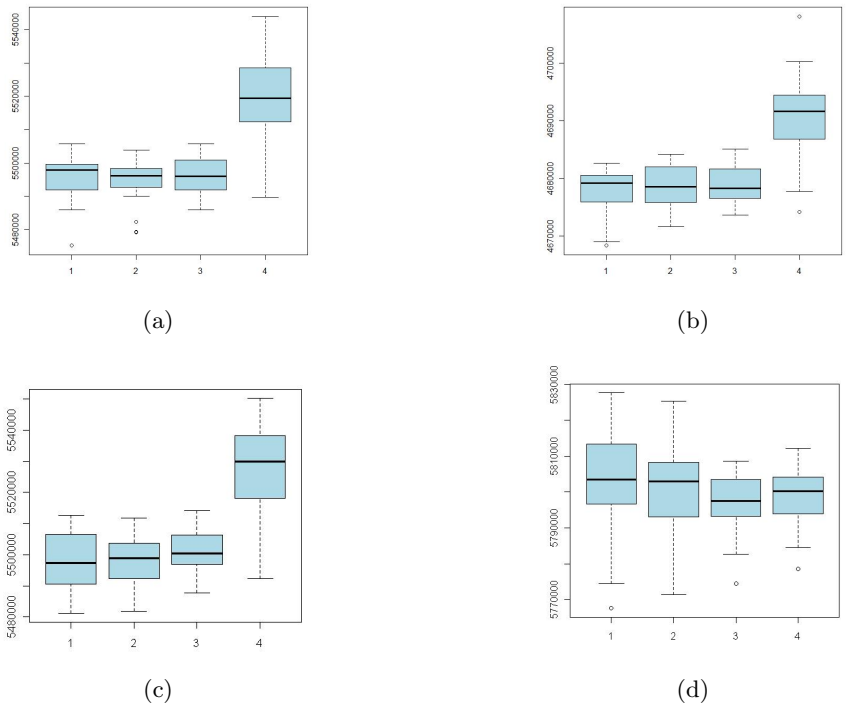

Figura 1. Boxplots para los conjuntos a) LD401, b) LD402, c) LD403 y d) LD404 correspondientes a las cuatro mejores configuraciones.

5. Conclusiones

En este artículo presentamos el diseño de un algoritmo ACO para la obtención de soluciones aproximadas al problema de triangulación de peso mínimo. Además mostramos los resultados obtenidos en la primera etapa experimental que nos permiten mostrar cuáles son las configuraciones de parámetros del algoritmo ACO-MWT más propicias para el problema de optimización presentado. En función de los resultados obtenidos en esta primera etapa, actualmente la experimentación se realiza sobre diversos lotes de pruebas, los cuales son obtenidos por diversos generadores de puntos, manteniéndose lotes de 40, 80, 120, 160 y 200 puntos. De esta manera, pretendemos realizar un análisis estadístico en profundidad, a fin de fundamentar las configuraciones que conjeturan ser las más adecuadas para el problema en cuestión. También, a futuro, pretendemos constatar el rendimiento del algoritmo ACO-MWT comparándolo con otras técnicas metaheurísticas, a fin de poder realizar un análisis elaborado acerca del comportamiento de estas técnicas en la obtención de triangulaciones de mínimo peso.

Referencias

1. Bäck T., Fogel D., Michalewicz Z.: Handbook of Evolutionary Computation. IOP Publishing Ltd and Oxford University Press (1997)
2. Computational Geometry Algorithms Library (CGAL). http://www.cgal.org/
3. Capp K., Julstrom B.: A weight-coded genetic algorithm for the minimum weight triangulation problem. Proc.of ACM symposium on Applied Computing (1998)
4. Düppe R., Gottschalk H.: Automatische Interpolation von Isolinien beiwillkürlichen Stützpunkten. Allgemeine Vermessungsnachrichten 77, 423-426 (1970)
5. M. Dorzán, E. Gagliardi, M. Leguizamón y G Hernández Peñalver, Algoritmo ACO aplicado a la obtención aproximada de Triangulaciones de Peso Mínimo. XXXV Conferencia Latinoamericana de Informática (2009)
6. M. Dorzán, E. Gagliardi, M. Leguizamón, M. Taranilla y G. Hernández Peñalver, Algoritmos ACO aplicados a problemas geométricos de optimización. XIII Encuentros de Geometría Computacional (2009)
7. Dorigo M., Stützle T.: Ant Colony Optimization. Massachusetts Institute of Technology (2004)
8. Kennedy J., Eberhart R.: Swarm Intelligence. Morgan Kaufmann Publishers (2001)
9. Kolingerova I., Ferko A.: Multicriteria-optimized Triangulations. The Visual Computer, Springer Verlag, Vol. 17, No. 6, 2001, s.380-395 (2001)
10. Lloyd E.: On triangulations of a set of points in the plane. In 18th Annual Symposium on Foundations of Computer Science. IEEE Computer Society, 228-240 (1977)
11. Lingas A.: A linear-time heuristic for minimum weight triangulation of convex polygons. In Proc. 23rd Allerton Conf. Commun. Control Comput. (1985)
12. Levcopoulos C., Krznaric D.: A nearoptimal heuristic for the minimum weight triangulation of convex polygons. Unpublished (1997)
13. Levcopoulos C., Lingas A., Sack J.: Heuristics for optimum binary search trees and minimum weight triangulation problems. Theoret. Comput. Sci, 66(2):181-203 (1989)
14. Michalewicz Z., Fogel D.: How to Solve It: Modern Heuristics. 2nd Edition, Springer (2004)
15. Mulzer W. y Rote G., Minimum weight triangulation is NP-hard. Proceedings of the 22nd Annual ACM Symp. on Computational Geometry. pp. 1-10 (2006)
16. Manacher G., Zobrist, A.: Neither the greedy nor the Delaunay triangulation of a planar point set approximates the optimal triangulation. Inf. Process. Lett. 9, 1, 31-34 (1979)
17. I. Osman y J. Kelly, MetaHeuristics: Theory & Application. Kluwer Academic Publishers (1996)
18. Plaisted D., Hong J.: A heuristic triangulation algorithm. J. Algorithms, 8:405-437 (1987)
19. Qin K., Wang W., Gong M.: A genetic algorithm for the minimum weight triangulation. In Proceedings of 1997 IEEE International Conference on Evolutionary Computation, pages 541-546 (1997)
20. Shamos M., Hoey D.: Closest-point problems. In 16th Annual Symp. on Found.of Computer Science. IEEE Computer Society, Long Beach, Calif., 151-162 (1975)
21. Sen S., Zheng S.: Near-optimal triangulation of a point set by Simulated Annealing. Proceedings of the 1992 CM/SIGAPP Symposium on Applied Computing, pp. 1000-1008 (1992)
22. Wu Y., Wainwright R.: Near-optimal triangulation of a point set using Genetic Algorithms. Proceedings of the Seventh Oklahoma Conference on AI, pp. 122-131, November (1993)