Algoritmo de Colonia de Hormigas para el Problema del Clique Máximo con un Optimizador Local K-opt

Julio C. Ponce[1], Eunice E. Ponce de León[1], Alejandro Padilla[1],Felipe Padilla[1], Alberto Ochoa O. Zezzatti[2]

[1]Departamento de Sistemas Electrónicos – Universidad Autónoma de Aguascalientes
Aguascalientes, México

[2] Facultad de Ingeniería Eléctrica – Universidad Autónoma de Zacatecas
Zacatecas, México

julk_cpg@yahoo.com.mx,{eponce,apadilla}@correo.uaa.mx,
fpadilla2000@yahoo.com, megamax8@hotmail.com

Resumen. *En este artículo se muestra un Algoritmo de Colonias de Hormigas (ACH) para resolver el problema del clique máximo (PCM) el cual contiene un optimizador local K-opt en el momento de construir un clique. Esta implementación se hizo en la forma de seleccionar los nodos que forman parte del clique dentro del proceso de selección de un nodo durante la construcción de la solución, ya que se implemento el método de búsqueda local. Los resultados experimentales en este primer acercamiento nos dan buenos resultados.*

Resumo. *Este artigo da pesquisa descreve o algoritmo da Colônia da Formigas (ACF) pra resolver o problema da encontrar o subgraph completo mais grande dentro do graph (problema del clique máximo (PCM), em espanhol), este algoritmo tem uma ferramenta ótima local k-opt pra construir a subgraph. Esta execução ficou em a forma da escolher os nodos dentro do processo da seleção de um nodo, ficou executado o método da pesquisa local. Os resultados da o experimento em este primeiro aproximação mostraram bom resultados*

Keywords. *Optimización con Colonia de Hormigas, Clique Máximo.*

1. Introducción

El problema del clique máximo es un problema de optimización combinatoria que se clasifica dentro de los problemas NP- duros los cuales son difíciles de resolver.Debido a su complejidad las técnicas convencionales exactas (exhaustivas) tardan mucho tiempo para dar una solución, por lo tanto es necesario desarrollar algoritmos heurísticos que lo resuelvan alcanzando una solución cercana al óptimo en un tiempo razonable. Este problema tiene aplicaciones reales como son: teoría de códigos, diagnóstico de errores, visión computacional, análisis de agrupamiento, recuperación de información, aprendizaje automático, minería de datos, entre otras. Por esta razón es importante usar nuevas técnicas heurísticas y/o metaheurísticas para tratar de resolver este problema, las cuales obtengan mejores resultados en un tiempo polinomial.

Se han utilizados diferentes heurísticas para tratar de resolver este problema, como son: Búsqueda Local [Battiti and Protasi 2001, Katayama, Hamamoto, Narihisa 2004], Algoritmos Genéticos [Marchiori 2002], Búsqueda Tabú [Soriano and Gendrean 1996] y Algoritmos de Optimización de Colonia de Hormigas (OCH) [Solnon and Fenet 2006], este último muestra unos de los resultados mas recientes. Los algoritmos de optimización de colonias de hormigas es una metaheurística bioinspirada basada en el comportamiento de las hormigas naturales, en como establecen el camino más adecuado entre el hormiguero y una fuente de alimento [Dorigo, Maniezzo and Colorni 1996].

2. Descripción del Problema del Clique Máximo

Dado un grafo no dirigido cualquiera G= (V,E), en el cual V={1,2,…,n} es el conjunto de los vértices del grafo y E es el conjunto de aristas. Un clique es un conjunto C de vértices donde todo par de vértices de C esta conectado con una arista en G, es decir C es un subgrafo completo. Un clique es parcial si este forma parte de otro clique, de otra forma es maximal. La meta del PCM es el encontrar un clique máximo, es decir el clique maximal con mayor cardinalidad. Se puede observar gráficamente un ejemplo de un clique en figura (1).

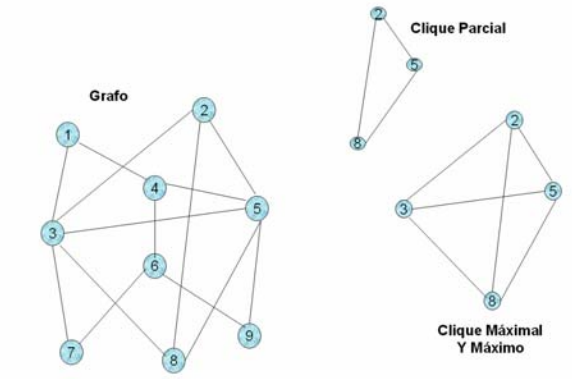

Figura 1. Ejemplo de un clique máximo

3. Algoritmo Ant Clique (AC)

Los algoritmos OCH han sido utilizados para resolver diferentes problemas de optimización combinatoria que están clasificados dentro de los NP-Duros [Dorigo, Di Caro and Gambardella 1999]. La idea principal del OCH es modelar los problemas para buscar el camino de costo mínimo en un grafo. Las hormigas recorren el grafo en busca de buenos caminos (soluciones). Cada hormiga es un agente que tiene un comportamiento simple de modo que no siempre encuentra caminos de calidad cuando esta sola. Las hormigas encuentran mejores caminos como resultado de la cooperación global entre la colonia. Esta cooperación se realiza de una manera indirecta al depositar la feromona, una sustancia que es depositada por una hormiga en su recorrido.

El algoritmo de colonia de hormigas general para el problema del clique máximo propuesto por Fenet y Solnon se muestra en la figura 2

Inicializar los rastros de feromonas
Repetir
Para k k en 1..nbAnts **Hacer**: construya el clique C_k
Actualizar los rastros de feromonas w.r.t. { $C_1, \ldots,$ C_{nbAnts} }
Hasta Alcanzar el Número de Ciclos o Encontrar la solución optima

Figura 2. Seudocódigo del algoritmo del Ant-Clique

Inicialización de la feromona: Las hormigas se comunican a través de la feromona la cual es depositada en las aristas del grafo. La concentración de feromona en la arista (v_i, v_j) es denotada por τ (v_i, v_j), el rastro de feromona inicial es denotada por c, aunque se puede utilizar los límites $\tau_{\min}$ ó $\tau_{\max}$ los cuales están basados en el algoritmo MAX-MIN [Stutzle and Hoos 2000] con el propósito de tener una mayor exploración del espacio de búsqueda.

Construcción de los cliques con las hormigas: Se selecciona aleatoriamente un vértice inicial, y iterativamente se escoge vértices para agregar al clique. Dentro de un conjunto de candidatos que contiene todos los vértices que están conectados con todos los vértices del clique, ver figura 3.

Escoger aleatoriamente el primer vértice $v_f \in V$
$C \leftarrow \{ v_f \}$
Candidatos $\leftarrow \{ v_i / (v_f, v_i) \in E \}$
Mientras Candidato $\neq 0$ **Hacer**
Escoger un vértice $v_i \in$ Candidatos con una probabilidad p(v_i),ver Ec. (2)
$C \leftarrow C \cup \{ v_i \}$
Candidatos $\leftarrow$ Candidatos $\cap \{ v_j / (v_i, v_j) \in E$
Fin Mientras
Regresa C

Figura 3. Algoritmo para construir un clique

La actualización del rastro de feromona utiliza la Ec. (1).

$$\tau_{ij}(t+n) = \rho\tau_{ij}(t) + \Delta\tau_{ij} \qquad (1)$$

$$p(v_i) = \frac{\left[\tau\, c(v_i)\right]^{\alpha}}{\sum v\ j\ candidatos \quad \left[\tau\, c(v_j)\right]^{\alpha}} \qquad (2)$$

4. Algoritmo Propuesto

El algoritmo propuesto está basado en el algoritmo mostrado anteriormente la diferencia se encuentra en la parte de como se escoge el siguiente vértice dentro del proceso de construcción del clique la cual se muestra en la figura 4.

```
Obtener el grado de los vértices candidatos
Ordenarlos por el grado de mayor a menor
Si el número de vértices del grado mayor es 1 Entonces
    Seleccionarlo como siguiente nodo
Sino
    Escoger uno de los vértices con mayor grado con una probabilidad p( v_i ),ver Ec. 2
```

Figura 4. Seudocódigo para seleccionar el vértice candidato

La obtención del grado de los vértices candidatos se debe de hacer en base a la lista de candidatos y no en relación al grafo original ya que esto nos aumenta la probabilidad de encontrar mejores soluciones debido a que se selecciona como siguiente vértice uno que tiene todavía muchas relaciones con los vértices candidatos restantes.

4.1. Diseño de Experimentos

Los algoritmos OCH dependen de los parámetros α que es el factor de peso (importancia) de la feromona, y ρ es el porcentaje de evaporación de la feromona. Si decrecemos el valor de α, las hormigas tienen menos sensibilidad al rastro de feromonas, y si se incrementa ρ, la evaporación de la feromona es más lenta. Cuando se incrementa la habilidad de exploración de las hormigas, estas pueden encontrar mejores soluciones pero esto implica más tiempo. Tomando en cuenta estos parámetros se ejecutó el algoritmo con los siguientes valores: número de hormigas=100, concentración inicial de la feromona τ_0 =0.01, importancia de la feromona α=1, factor de evaporación de la feromona ρ=0.99, concentración máxima que puede tomar la feromona τ_{max}, número de ciclos que se ejecuta el algoritmo Nc=100, estos valores fueron escogidos en base a los resultados obtenidos al implementar un primer algoritmo a principios del 2006 en el cual se colocaron las hormigas dentro del grafo en los vértices con mayor grado[Ponce, Ponce de León, Padilla, Padilla 2006].

Para llevar a cabo el diseño de experimentos tomamos los benchmark de la DIMACS [DIMACS] que son los utilizados actualmente a nivel internacional para este problema.

5. Resultados

Se decidió resolver el problema con la ejecución del software con los parámetros antes mencionados en el algoritmo, en 20 de los 36 benchmarks de la DIMACS , y los resultados obtenidos se muestran en la Tabla 1, donde en la columna "grafo" esta el nombre que tiene el grafo en la DIMACS, La columna "Tamaño del clique benchmark" es el tamaño que tiene el clique máximo(Solución Optima) y Tamaño del Clique encontrado por ACH desarrollado.

Tabla 1. Resultados del algoritmo en los bechmarks

Grafo	Tamaño del Clique benchmark	Tamaño del Clique encontrado por ACH
C125.9	34	34*
C250.9	44	44*
C500.9	57	53
C1000.9	68	63
C2000.9	78	72
DSJC500.5	13	13*
DSJC1000.5	15	13
Brock200_2	12	12*
Brock200_4	17	17*
Brock400_2	29	24
Brock400_4	33	28
Brock800_2	24	21
Brock800_4	26	21
C2000.5	16	16*
C4000.5	18	16
Hamming8_4	16	16*
Keller4	11	11*
P_hat300_1	8	8*
P_hat700_1	11	10
P_hat1500_1	12	12*

En la tabla anterior los resultados que cuentan con un "*" son en los que se alcanzo la solución optima de los benchmarks.

6. Conclusiones

En este artículo presentamos un algoritmo basado en colonia de hormigas el cual introduce un optimizador local k-opt al escoger un nuevo vértice, el cual toma en cuenta el grado de estos lo cual puede incrementar la probabilidad de encontrar cliques más grandes.

El algoritmo propuesto fue ejecutado en 20 benchmarks de la DIMACS para el problema del clique máximo de los cuales en el 50% de los benchmarks en los que se ejecutó el algoritmo obtiene el óptimo, y en la solución más lejana queda a un 19% de la solución óptima. La ventaja de este algoritmo con el reportado en el CISCI'2006 es la velocidad.

Trabajo futuro: Es importante hacer un estudio del comportamiento de los parámetros y del algoritmo, así como hacer un diseño de experimento más amplio para determinar cuales son los mejores valores para los parámetros y comparar los dos algoritmos implementados.

7. Referencias

Battiti R. and Protasi M.(2001) Reactive local search for the maximum clique problem. Algorithmica, 29(4): 610–637.

DIMACS Center for Discrete Mathematics and Theoretical Computer Science fttp://dimacs.rutgers.edu/pub/challenge/graph/benchmarks/

Dorigo, M.; Maniezzo, V. and Colorni, A. (1996) Ant System: Optimization by a Colony of Cooperating Agents. IEEE Transactions on Systems, Man And Cybernetics –Part B: Cybernetics, 26:1, 29-41.

Dorigo M., Di Caro G., and Gambardella L.M.(1999) Ant algorithms for discrete optimization. Artificial Life, 5(2): 137–172.

Fenet S. and Solnon C. (2003) Searching for Maximum Cliques with Ant Colony Optimization EvoWorkshops 2003, LNCS 2611, 236–245.

Katayama, Hamamoto, Narihisa (2004) Solving the Maximum Clique Problem by

K-opt Local Search. ACM Symposium on Applied Computing, 1021-1025.

Marchiori E.(2002) Genetic, iterated and multistart local search for the maximum clique problem. In Applications of Evolutionary Computing, Proceedings of EvoWorkshops2002: EvoCOP, EvoIASP, EvoSTim, volume 2279, 112–121.

Ponce J., Ponce de León E., Padilla F. Padilla A. (2006) Implementación de un Algoritmo de Colonia de Hormigas para el Problema del Clique Máximo. En CISCI'2006 volumen 1: 70-73.

Solnon C. and Fenet S. (2006) A study of ACO capabilities for solving the Maximum Clique Problem Journal of Heuristic, Volume 12 - Number 3. 155-180.

Soriano P. and Gendreau M. (1996) `Diversification strategies in tabu search algorithms for the maximum clique problem', Ann. Oper. Res., Vol. 63: 189-207.

Stutzle T. and Hoos H.H. (2000) MAX–MIN Ant System. Journal of Future Generation Computer Systems, 16: 889–914.