Problema del Clique Máximo: Estrategias altamente paralelizables

Simón Castillo
Departamento de Computación y
Tecnología de la Información
Universidad Simón Bolívar
Sartenejas, Caracas, Venezuela
Email: simoncastillo@gmail.com

Alejandro Flores
Departamento de Computación y
Tecnología de la Información
Universidad Simón Bolívar
Sartenejas, Caracas, Venezuela
Email: alejandroflores.af@gmail.com

I. INTRODUCCIÓN

El problema del MAXCLIQUE (abreviado MCP por sus siglas en inglés) es un problema clásico de optimización combinatoria, el cual posee importantes aplicaciones en diferentes dominios como teoría de códigos, diagnóstico de fallas o visión por computadoras. De manera informal, el problema consiste en encontrar un subgrafo completo de tamaño máximo. En general, este problema es inmanejable, por lo que se ha invertido una cantidad de tiempo y esfuerzo en lograr diseñar algoritmos que generen soluciones óptimas de este problema en un tiempo razonable.

Durante los últimos años, la utilización de cliques como parte fundamental para determinar la estructura de diversos problemas ha ido considerablemente en aumento. Dichos problemas residen en las áreas de las ciencias sociales, biológicas, financieras, entre otras; donde cada uno de ellos dependen de la búsqueda de cliques en redes de gran densidad. Esto convierte al problema computacionalmente costoso, tanto para realizar el cálculo, como para ser representado por un computador (i.e. los grafos originales no pueden ser cargados en memoria principal, por lo que muchas veces es necesario guardar la estructura del grafo en memoria secundaria e ir cargando a medida que se vaya necesitando).

La literatura cuenta con gran cantidad de aportes que intentan resolver este importante problema de la teoría de grafos. En particular, diversas técnicas meta-heurísticas se han aplicado para tratar de conseguir cliques máximos en un tiempo de computo razonable. En particular, el estado del arte se ha enfocado en el uso de técnicas basadas en la búsqueda local como lo son la búsqueda tabú, la búsqueda k-opt, vecindades variables, entre otras. Asímismo, las técnicas meta-heurísticas basadas en población han recibido considerables contribuciones, en particular los algoritmos genéticos, para la cual el problema en principio se presta muy bien, y, en menor medida, la optimización por colonias de hormigas.

A partir de finales de la década pasada, los procesadores multi-núcleo se han vuelto altamente disponible en el mercado de consumidores. Este hecho presenta nuevas posibilidades para problemas computacionales altamente costosos donde se podría conseguir más y mejores resultados de ser posible paralelizar la carga de computo en múltiples instancias que realicen la búsqueda de soluciones en paralelo para luego consolidarlas en una mejor solución. Para el conocimiento de los autores, para el problema no se ha intentado paralelizar con éxito. El principal aporte que presentaremos en el presente trabajo es un estudio de dos técnicas de búsqueda meta-heurística que resultan comportarse sumamente bien al ser paralelizadas en múltiples núcleos. En primera instancia, consideraremos la técnica presentada por [14], conocida como *Dynamic Local Search* (DLS), considerada una técnica del estado del arte para la búsqueda de cliques. Posteriormente, consideraremos la optimización por colonia de hormigas, basados en el trabajo por [7], [17], la cuál expandiremos utilizando búsqueda local aprovechando DLS.

El presente trabajo está organizado como sigue: en la sección II presentamos un breve análisis de diferentes métodos encontrados en la literatura disponible, en la sección III presentamos un resumen general de la notación y conceptos básicos que utilizaremos, en la sección IV presentamos la descripción de DLS junto al proceso empleado para volver el algoritmo paralelo, V presentamos el marco de trabajo para la optimización por medio de colonias de hormiga y de igual forma su proceso de paralelización, en las VI presentamos la evaluación experimental de las técnicas empleadas y finalmente en la sección VII presentamos conclusiones y direcciones futuras.

II. TRABAJO PREVIO

MAXCLIQUE es unos de los problemas originales de Karp [11] que fueron demostrados como NP-completo, es decir, a menos que P = NP, algoritmos exactos están garantizados a devolver una solución sólo en un tiempo que incrementa exponencialmente con el número de nodos en el grafo. Más aún, [1] probó que para algún $\epsilon > 0$ la aproximación del número de clique dentro de un factor de $|V|^{\epsilon}$ es NP-duro; de hecho, el mejor algoritmo aproximado en tiempo polinomial requiere $O(|V|/(\log|V|)^2)$ [4].

Aunque clásicamente MAXCLIQUE es encontrado en la literatura como su variante en forma de problema de decisión NP-completo, el problema de optimización se presta a diversas técnicas de optimización combinatoria. Desde un punto de vista práctico, algoritmos exactos son capaces de dar respuesta con instancias de grafos en el orden de los cientos de nodos [21], [15], [19]. Sin embargo, para poder manejar instancias mucho más complejas (i.e. grafos con una cardinalidad de órdenes de magnitud mayores), diversas técnicas del estado del arte emplean diversas metaheurísticas. De igual forma, es importante destacar que el problema de MAXCLIQUE es polinomialmente equivalente a INDEPENDENTSET así como al MINVERTEXCOVER, por lo que cualquier heurística que funcione para estos problemas dará buenos resultados para el problema MAXCLIQUE [3].

A pesar del considerable esfuerzo que se ha volcado en el desarrollo de métodos basados en heurísticas, no existe un solo método que obtenga los mejores resultados. De hecho, a pesar de la dificultad que puede existir en comparar diversos métodos (i.e. diferencias en donde se corren las pruebas), es posible diferenciar varios métodos que podemos considerar estado del arte: *Reactive Local Search* [2], un método basado en búsqueda tabú que adapta automaticamente el parámetro de tenencia de la tabla tabú, *Deep Adaptive Greedy Search* [9], la cuál utiliza un algoritmo voraz iterativo, *búsqueda k-opt* [12], basado en *Variable Depth Search* aumentado con una heurística simple de agregación o remoción de vértices del mejor clique conseguido hasta el momento, *VNS* [10], basado en *vecindades variables* con búsqueda voraz en la vecindades, y *Dynamic Local Search-Maximum Clique* [14], una heurística que alterna fases de mejoramiento iterativo (donde los vértices son agregados al clique), con búsqueda en los *plateau* (donde vértices agregados al clique son cambiados por vértices que no se encuentran en el clique).

Las técnicas basadas en búsqueda local toman un lugar importante en este problema. En particular, como podemos ver en [8], [18], [20], la búsqueda tabú resulta ser un método no solo muy estudiado sino sino que arroja resultados prometedores. En particular, [8], [18], de los primeros en estudiar la búsqueda tabú para el problema del clique máximo, implementan dos variantes de la búsqueda tabú: una que funciona de forma determinista y una que funciona de forma probabilista. En estos métodos se varía considerablemente los parametros de tenencia y se estudia su efecto sobre el problema. Sin embargo, resulta interesante para nuestros intereses estudiar la búsqueda k-opt presentada en [12]. Esta simple heurística consiste en la en una búsqueda basada en vecindarios variables donde se alternan las fases de intensificación (agregar vértices al clique) con fases de eliminación de vértices cuya remoción permita agregar nuevos vértices posteriormente. Esta simple heurística, cuyo proceso de selección de vértices esta basado no más que en el grado de los vértices, resulta ser sumamente efectiva para el problema del clique máximo. De esta forma, no solo son capaces de dar resultados que son considerados del estado del arte, sino que es capaz de hacerlo en un tiempo de ejecución muy rápido.

A partir de las ideas implementadas en la búsqueda k-opt se derivan dos heurísticas muy similares: *Deep Adaptive Greedy Search* (DAGS) y *Dynamic Local Search* (DLS). Estas dos heurísticas, de igual forma, alternan fases de intensificación de la búsqueda con remoción de vértices (búsqueda *plateau*). Sin embargo, agregan un mecanismo numérico de diversificación de la búsqueda que los autores de DLS en [14] llaman *tabla de penalización*. Esta tabla, asigna a cada vértice una penalización al final de cada ciclo de búsqueda, donde los vértices asociados a un clique aumentan su valor de penalización, y designa que el criterio de selección de cada vértice sea el vértice con el mínimo valor de penalización disponible. Este criterio no solo permite diversificar considerablemente la búsqueda sino que permite disminuir considerablemente los estados (cliques) previamente visitados y permite visitar rápidamente el espacio de búsqueda.

De igual forma, los algoritmos basados en el funcionamiento de la naturaleza son capaces de conseguir buenos resultados al problema. En particular, la investigación de métodos basados en algoritmos genéticos constituye una parte considerable de la literatura asociada al problema del MAXCLIQUE. En [16] se presenta una nueva técnica híbrida basada en algoritmos genéticos y se realiza una comparación de varios algoritmos evolutivos. Sin embargo, a pesar de ser capaz de dar mejores resultados que los otros algoritmos evolutivos, el método presentado no supera los resultados obtenidos con métodos no evolutivos. Por otra parte, los cliques son utilizados considerablemente para estudiar la estructura interna de las moléculas en las ciencias biológicas y la química orgánica, por lo que diversos métodos de búsqueda de cliques en espacios tridimensionales han sido estudiados [6], [13]. Otro método basado en observaciones de la naturaleza, aunque mucho menos estudiado que los algoritmos genéticos, que ha resultado en buenos resultados con respecto al estado del arte constituye la optimización por colonias de hormigas (*ant colony optimization*, ACO).

Aunque no tan influyentes como otros trabajos en la literatura del clique máximo, en [7], [17] se estudian las capacidades de ACO en dos instanciaciones genéricas del problema las cuales generan cliques maximales sucesivos y utilizando una heurística voráz para seleccionar cada vértice nuevo a ser agregado. Posteriormente, estos cliques conseguidos son mejorados por búsqueda local basada en un criterio de selección muy parecido al utilizado en GRASP y similar al usado en DLS. Los resultados obtenidos muestran que ACO es competitivo con otros métodos heurísticos del estado del arte, y que al agregar búsqueda local el tiempo de corrida mejora considerablemente. Además, en estos trabajos no solo se realiza un estudio de la calidad de las soluciones sino como el problema del clique máximo se presta para ser explotado por ACO. Asimismo, en estos trabajos, como resultado importante se presentan dos heurísticas para la asignación de las feromonas que las hormigas siguen: `Vertex-AC` y `Edge-AC`. La primera, considera colocar las feromonas sobre cada vértice mientras que la segunda considera colocar las feromonas sobre cada lado. Para nuestros intereses, únicamente consideraremos la primera heurística, `Verte-AC` la cual trataremos de expandir con el uso de DLS.

III. DEFINICIONES

Usaremos la siguiente notación: $G = (V, E)$ denota un grafo no dirigido donde, sin perder generalidad, $V = \{1, 2, 3, \ldots, n\}$ y $E \subseteq \{(i, j) : i, j \in V\}$. $N(i)$ con $i \in V$ representa la vecindad del vértice i (i.e. los vertices adyacentes a i).

Definición 1. *Dado un grafo no dirigido $G = (V, E)$, donde V es el conjunto de vértices y $E \subseteq V \times V$ el conjunto de lados, un **clique** es un conjunto de vértices $C \subseteq V$ tal que cada pareja de vértices distintos de C está conectado por un lado en E, es decir, el subgrafo inducido por C es completo.*

Definición 2. *Un* clique *es **parcial** si está estrictamente incluído en otro (es decir, su cardinalidad puede aumentarse añadiendo vértices pertenecientes a E a dicho clique); de lo contrario es **maximal**.*

Definición 3. *Un **clique máximo** es un clique maximal de cardinalidad máxima.*

En la Fig. 1 se encuentran ejemplos de estas definiciones para un grafo G_1. Claramente, podemos observar que todo clique máximo es necesariamente máximal por definición, pero todo clique máximal no necesariamente es máximo. El problema de enumerar los cliques maximales de un grafo ha recibido considerable atención en la literatura, en particular el algoritmo de Bron-Kerbosch [5] es capaz de enumerar todos los cliques maximales de un grafo utilizando búsqueda recursiva. De esta forma, es posible conseguir un algoritmo ingenuo para el problema del clique máximo, el cual funciona enumerando todos los cliques de un grafo y tomando el de máxima cardinalidad. Todas las técnicas meta-heurísticas que estudiaremos parten de la noción de conseguir cliques maximales y mejorarlos de diversas maneras.

Definición 4. *Dado un grafo $G = (V, E)$, la cardinalidad del clique máximo se denota $\omega(G)$. Este valor se conoce como el número de clique.*

Definición 5. *Dado un grafo $G = (V, E)$ arbitrario, decidir si G tiene un clique de tamaño k es un problema* NP*-completo.*

Definición 6. *Dado un grafo $G = (V, E)$ arbitrario, conseguir un clique máximo en el grafo G, es decir, conseguir un subgrafo completo G' de cardinalidad máxima es* NP*-duro.*

En este trabajo nos dedicaremos a estudiar el problema de optimización, es decir, el problema de conseguir un clique de tamaño máximo.

Definición 7. *Dado un grafo $G = (V, E)$ arbitrario y un clique C en este grafo, llamamos a su **conjunto de mejoramiento** al conjunto de $I \subseteq V$ tal que para $x \in I$ se tiene que x está conectado a todos los elementos de C en G.*

El conjunto de mejoramiento también se conoce como el conjunto de candidatos. Un clique máximal claramente tiene un conjunto de mejoramiento vacío.

Definición 8. *Dado un grafo $G = (V, E)$ arbitrario y un clique C en este grafo, llamamos a su **conjunto de nivel** al conjunto $L \subseteq V$ tal que para $x \in L$ se tiene que x esta conectado a todos los elementos de C en G menos* exactamente *uno. Para cada elemento $x \in L$, el elemento $y \in C$ tal que no existe $(x, y) \in E$ se le conoce como el **dual** de x.*

El conjunto de nivel de un clique C en un grafo G permite realizar intercambio elemento a elemento por su dual. Este proceso resulta ser la base para el proceso de búsqueda por fases de los diversos algoritmos que combinan métodos de intensificación con remoción de vértices.

IV. BÚSQUEDA DINÁMICA LOCAL

V. OPTIMIZACIÓN POR COLONIA DE HORMIGAS

VI. EXPERIMENTOS Y RESULTADOS

VII. CONCLUSIONES Y FUTURAS DIRECCIONES

REFERENCES

[1] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of np. *J. ACM*, 45(1):70–122, 1998.

[2] Roberto Battiti and Marco Protasi. Reactive local search for the maximum clique problem. Technical report, Algorithmica.

[3] Immanuel M. Bomze, Marco Budinich, Panos M. Pardalos, and Marcello Pelillo. The maximum clique problem. In *Handbook of Combinatorial Optimization*, pages 1–74. Kluwer Academic Publishers, 1999.

[4] Ravi Boppana and Magnús M. Halldórsson. Approximating maximum independent sets by excluding subgraphs, 1992.

[5] Coen Bron and Joep Kerbosch. Algorithm 457: finding all cliques of an undirected graph. *Commun. ACM*, 16(9):575–577, September 1973.

[6] JohnD. Eblen, CharlesA. Phillips, GaryL. Rogers, and MichaelA. Langston. The maximum clique enumeration problem: Algorithms, applications and implementations. In Jianer Chen, Jianxin Wang, and Alexander Zelikovsky, editors, *Bioinformatics Research and Applications*, volume 6674 of *Lecture Notes in Computer Science*, pages 306–319. Springer Berlin Heidelberg, 2011.

[7] Serge Fenet and Christine Solnon. Searching for maximum cliques with ant colony optimization. In Stefano Cagnoni, ColinG. Johnson, JuanJ.Romero Cardalda, Elena Marchiori, DavidW. Corne, Jean-Arcady Meyer, Jens Gottlieb, Martin Middendorf, Agnès Guillot, GüntherR. Raidl, and Emma Hart, editors, *Applications of Evolutionary Computing*, volume 2611 of *Lecture Notes in Computer Science*, pages 236–245. Springer Berlin Heidelberg, 2003.

[8] Michel Gendreau, Patrick Soriano, and Louis Salvail. Solving the maximum clique problem using a tabu search approach. *Annals of Operations Research*, 41(4):385–403, 1993.

[9] A. Grosso, M. Locatelli, and F. Della Croce. Combining swaps and node weights in an adaptive greedy approach for the maximum clique problem. *Journal of Heuristics*, 10(2):135–152, March 2004.

[10] Pierre Hansen, Nenad Mladenović, and Dragan Urošević. Variable neighborhood search for the maximum clique. *Discrete Appl. Math.*, 145(1):117–125, December 2004.

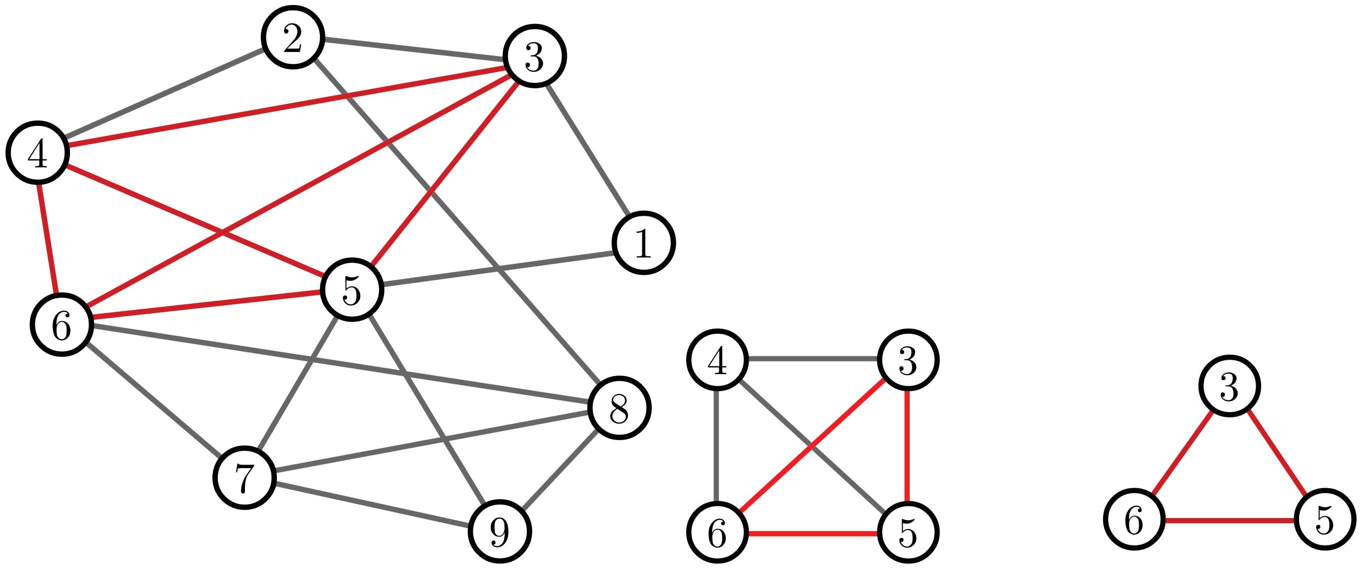
[11] Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103, 1972.

[12] Kengo Katayama. Solving the maximum clique problem by k-opt local search. In *In Proceedings of the 2004 ACM Symposium on Applied computing*, pages 1021–1025, 2004.

[13] Noël Malod-Dognin, Rumen Andonov, and Nicola Yanev. Maximum cliques in protein structure comparison. In Paola Festa, editor, *Experimental Algorithms*, volume 6049 of *Lecture Notes in Computer Science*, pages 106–117. Springer Berlin Heidelberg, 2010.

[14] Wayne Pullan and Holger H. Hoos. Dynamic local search for the maximum clique problem. *Journal of Artificial Intelligence Research*, 25:159–185, 2006.

[15] Y. Shinano, T. Fujie, Y. Ikebe, and Ryuichi Hirabayashi. Solving the maximum clique problem using pubb. In *Parallel Processing Symposium, 1998. IPPS/SPDP 1998. Proceedings of the First Merged International ... and Symposium on Parallel and Distributed Processing 1998*, pages 326–332, 1998.



(a) Grafo G_1 con su clique máximo resaltado en rojo

(b) Clique máximal y parcial del grafo G_1

Fig. 1: Diversos cliques según las definiciones expuestas en el grafo G_1 formado por 9 vértices y 18 lados.

[16] Alok Singh and AshokKumar Gupta. A hybrid heuristic for the maximum clique problem. *Journal of Heuristics*, 12(1-2):5–22, 2006.

[17] Christine Solnon and Serge Fenet. A study of aco capabilities for solving the maximum clique problem. *Journal of Heuristics*, 12(3):155–180, 2006.

[18] Patrick Soriano and Michel Gendreau. Diversification strategies in tabu search algorithms for the maximum clique problem. *Annals of Operations Research*, 63(2):189–207, 1996.

[19] David R. Wood. An algorithm for finding a maximum clique in a graph. Technical report, 1997.

[20] Qinghua Wu and Jin-Kao Hao. An adaptive multistart tabu search approach to solve the maximum clique problem. *Journal of Combinatorial Optimization*, pages 1–23, 2011.

[21] Patric R.J. Östergård and Patric R. J. A fast algorithm for the maximum clique problem. *Discrete Appl. Math*, 120:197–207.