

Extensión de algoritmos ACO usando conceptos de *Tabu Search*

Franco Arito Guillermo Leguizamón

Laboratorio de Investigación y Desarrollo en Inteligencia Computacional
Departamento de Informática
Facultad de Ciencias Físico Matemáticas y Naturales
Universidad Nacional de San Luis
e-mail: {farito, legui}@unsl.edu.ar

Resumen

En los algoritmos ACO (del inglés Ant Colony Optimization) las hormigas artificiales iterativamente construyen soluciones a un problema de optimización combinatorio. La construcción de dichas soluciones está guiada por rastros de feromona los cuales actúan como un mecanismo de adaptación que permite sesgar la muestra de nuevas soluciones hacia regiones prometedoras del espacio de búsqueda. Adicionalmente, el sesgo de la búsqueda está influenciado por información heurística dependiente del problema.

En este trabajo, que describe parte de la línea de investigación relacionada al uso de memoria externa en algoritmos ACO, se propone la incorporación de conceptos de la metaheurística *Tabu Search* (TS) en el proceso de construcción de soluciones. Estos conceptos se refieren específicamente a la forma en la que TS utiliza la historia del proceso de búsqueda para evitar visitar soluciones ya analizadas.

Palabras Clave: Ant Colony Optimization, Tabu Search, Quadratic Assignment Problem.

1. Introducción

Los algoritmos ACO generan soluciones candidatas para un problema de optimización con un mecanismo de construcción donde la elección de la componente de solución a ser agregada en cada paso de la construcción está influenciada probabilísticamente por rastros de feromona e

información heurística [3]. En este trabajo, se examina la posibilidad de alternar la forma en la que se eligen las componentes de solución, introduciendo una memoria externa como mecanismo auxiliar para tomar las decisiones en cada paso de la construcción de una solución. Esta modificación está inspirada en la metaheurística *Tabu Search*¹, la cual usa explícitamente la historia de la búsqueda, para escapar de óptimos locales e implementar una estrategia explorativa. Este tipo de enfoque pertenece a una de las tendencias actuales en algoritmos ACO, en los cuales se incorpora una memoria externa como alternativa al mecanismo de elección de componentes de soluciones [1, 16, 17].

El algoritmo *Tabu Search* simple aplica búsqueda local con el criterio de “*best-improvement*” como componente básico y usa una memoria de corto plazo, para poder escapar de óptimos locales y evitar ciclos en la búsqueda. La memoria de corto plazo está implementada como una *lista tabú* que mantiene registro de las soluciones visitadas más recientemente y prohíbe movimientos hacia ellas. De esta forma, se restringe la vecindad de la solución actual a soluciones que no pertenecen a la

¹El significado de la palabra tabú designa a una conducta, actividad o costumbre prohibida por una sociedad, grupo humano o religión, es decir, es la prohibición de algo natural, de contenido religioso, económico, político, social o cultural por una razón de utilidad social. La búsqueda tabú no se refiere obviamente a ninguna de estas temáticas, sino al hecho de imponer restricciones para guiar un proceso de búsqueda con el objetivo de superar regiones del espacio de búsqueda. Estas restricciones operan de varias formas, como por ejemplo mediante la exclusión directa de alternativas de búsqueda clasificadas como “prohibidas”.

lista tabú, previniendo de ciclos infinitos y forzando a la búsqueda a aceptar movimientos que incluso pueden generar soluciones peores. Sin embargo, la implementación de la memoria a corto plazo como una lista que contiene soluciones completas puede no ser efectiva, por lo que en lugar de almacenar soluciones, se almacenan *atributos de soluciones* (e.g., componentes de soluciones, movimientos o diferencias entre dos soluciones).

La lista tabú, identificada usualmente con el uso de memoria a corto plazo, es solo una de las maneras posibles de aprovechar la historia de la búsqueda. La información recolectada durante todo el proceso de búsqueda también puede ser muy útil, especialmente para una guía estratégica del algoritmo. Este tipo de memoria a largo plazo generalmente se agrega a TS de acuerdo a cuatro principios: lo reciente, frecuencia, calidad e influencia. La memoria basada en lo reciente registra para cada solución (o atributo) la iteración más reciente en la que estuvo involucrada. En contraposición, la memoria basada en frecuencia mantiene registro de cuántas veces cada solución (o atributo) ha sido visitada (usado). Esta información identifica las regiones del espacio de soluciones donde la búsqueda estuvo concentrada o donde permaneció por mayor número de iteraciones. Es decir, es información que representa la memoria del proceso de búsqueda y que puede ser explotado para diversificar la búsqueda. Por su parte, el atributo calidad hace referencia a la habilidad para diferenciar la bondad de las soluciones visitadas a lo largo del proceso de búsqueda. De esta forma, la memoria puede ser utilizada para la identificación de elementos comunes a soluciones buenas o a ciertos caminos que conducen a ellas. La calidad constituye un fundamento para el aprendizaje basado en refuerzos, donde se premian las acciones que conducen a buenas soluciones y se penalizan aquellas que, por el contrario, conducen a soluciones pobres. Por último, la cuarta dimensión de memoria, referida a la influencia, considera el impacto de las decisiones tomadas durante la búsqueda, no solo en lo referente a la calidad de las soluciones, sino también en lo referente a la estructura de las mismas.

2. Uso de memoria externa en algoritmos ACO

El enfoque abordado agrega un mecanismo con características determinísticas en la construcción de soluciones, en contraposición con la filosofía presente en los algoritmos ACO, los cuales construyen soluciones en forma totalmente probabilística (basada en los rastros de feromona y en la información heurística). Las estructuras de memoria utilizada permiten que las hormigas en ciertas ocasiones² no tomen decisiones en forma aleatoria sino que elijan componentes de soluciones, de manera determinística, influenciadas por los valores registrados en dicha memoria. Como esta memoria almacena información específica de la historia de la búsqueda desde el comienzo del algoritmo, permite efectivamente enfocarse en regiones del espacio de búsqueda no visitadas (no registradas en la memoria y con valores que reflejen ésto) o por el contrario concentrarse en regiones ya visitadas y que sean prometedoras (también registradas en la memoria y con valores que lo reflejen). Estos usos de la memoria reflejan los mecanismos de intensificación y diversificación que, aunque ya estén presentes en los algoritmos ACO, son de utilidad para evitar convergencia prematura y lograr un mayor rendimiento frente a problemas de optimización con características particulares.

Se debe recalcar que aunque el mecanismo en estudio está inspirado en Tabu Search, existen ciertas diferencias en cuanto al uso de la memoria y también tiene sus diferencias respecto de otros algoritmos ACO que utilizan el enfoque de memoria explícita[1, 16, 17]. En este sentido, cabe destacar por un lado, que en TS las memorias basadas en lo reciente y en frecuencia se complementan la una a la otra para lograr el balance entre intensificación y diversificación. Así, la memoria basada en lo reciente es utilizada como memoria a corto plazo y aquella basada en frecuencia, como memoria a largo plazo (en el enfoque propuesto no existe esta distinción). Por otro lado, en los algoritmos ACO citados previamente, se propone almacenar en la memo-

²Para esto se utilizan parámetros que indican el nivel de exploración que tiene el algoritmo en determinados instantes de tiempo

ria partes de soluciones que pueden estar constituidas por las mejores soluciones encontradas durante la búsqueda (en el enfoque propuesto la memoria almacena atributos de soluciones recolectados durante el proceso de la búsqueda, los cuales representan indirectamente patrones interesantes de las soluciones). En lo que sigue se describen los principios considerados en nuestra propuesta.

2.1. Memoria basada en lo reciente

Este tipo de memoria almacena componentes de las soluciones que han cambiado en el pasado reciente. La forma habitual de explotar este tipo de memoria es etiquetando las componentes seleccionadas de soluciones visitadas recientemente. Lo que se intenta lograr es “prohibir” o “incentivar” ciertas elecciones, que impidan o favorezcan a que las hormigas construyan soluciones ya evaluadas en el pasado reciente y con lo cual se logra abarcar una mayor región del espacio de búsqueda o concentrarse en una región particular. Es importante tener en cuenta que en algunas instancias, un buen proceso de búsqueda resultará en volver a visitar una solución encontrada anteriormente. El objetivo más general es continuar estimulando el descubrimiento de nuevas soluciones de alta calidad.

2.2. Memoria basada en frecuencia

Este tipo de memoria almacena componentes de las soluciones que forman parte de una solución con mayor frecuencia, i.e., la mayor cantidad de veces presentes en una solución y en una posición particular de la solución. La forma habitual de explotar este tipo de memoria es etiquetando las componentes seleccionadas de soluciones más frecuentemente escogidas. Esta memoria, permite “prohibir” que una hormiga escoja una componente de solución porque es muy frecuentemente escogida en las soluciones. Se busca a través de la “prohibición”, generar soluciones que efectivamente se diferencien de las ya generadas, extendiendo de esta manera, la exploración del espacio de búsqueda. Inversamente, se puede usar esta información para “promover” su elección por considerarla atractiva ya que una gran mayoría de las hormigas la han elegido como parte de sus soluciones y por lo tanto, se considera deseable de que forme parte de una nueva solución.

3. Trabajos realizados

Como parte de la evaluación empírica de la propuesta, se ha implementado el enfoque de memoria explícita en el algoritmo $MAX - MIN$ Ant System [12] hibridado con aplicación de búsqueda local. Para la realización de los experimentos se ha considerado como caso de estudio, el problema de asignación cuadrática [5] (QAP, del inglés, Quadratic Assignment Problem). La formulación general de QAP describe las distancias entre n ubicaciones y los flujos entre n objetos. El objetivo es encontrar una asignación de objetos a ubicaciones, de manera tal de minimizar el producto de la distancia entre cada par de ubicaciones por el flujo entre el par de objetos asignados a ese par de ubicaciones. Distintos algoritmos ACO han sido aplicados a QAP [7, 9, 15, 4, 6]. Uno de los motivos de la elección de QAP como caso de estudio es que muchos tipos de instancias del mismo tienen características interesantes para estudiar el beneficio del enfoque de memoria explícita. En cuanto a la elección del algoritmo $MAX - MIN$ Ant System, se debe a que actualmente es uno de los mejores algoritmos ACO [3] para resolver QAP. Cabe destacar además, que dicho enfoque de memoria explícita puede ser fácilmente adaptado para ser aplicado a otros problema de optimización.

Las instancias utilizadas son las disponibles en QAPLIB³, una librería de instancias de benchmark para QAP. Los algoritmos ACO están actualmente entre los mejores algoritmos aproximados para resolver QAP, en particular para instancias estructuradas presentes en la vida real (ver [14] para una descripción detallada de los tipos de instancias). Sin embargo, en instancias no estructuradas (generadas aleatoriamente), su performance es inferior a métodos de mejora iterativa tales como búsqueda local iterada [10, 11], búsqueda tabú [13, 8], entre otros.

Las instancias de QAPLIB de tipo no estructuradas, generadas aleatoriamente tienen características que hacen que los algoritmos ACO inicialmente propuestos para este problema no sean competitivos. Una de estas características es que la *distancia* entre las soluciones que son óptimas

³<http://www.seas.upenn.edu/qaplib>

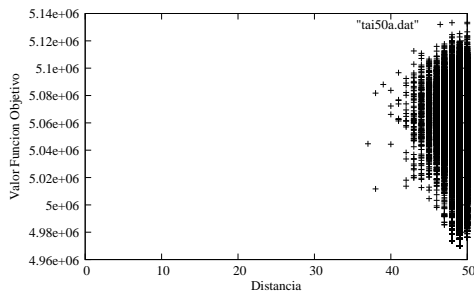


Figura 1: Instancia `tai50a` tomada de QAPLIB

locales es muy grande. Aquí asumimos que la distancia $d(\phi, \phi')$ entre dos soluciones ϕ y ϕ' para el QAP es:

$$d(\phi, \phi') = |\{i \in \{1, \dots, n\} \mid \phi_i \neq \phi'_i\}|$$

Si la distancia entre dos soluciones que son óptimos locales es grande, significa que esas soluciones tienen pocas componentes en común. La dificultad que este tipo de característica plantea a los algoritmos ACO, es que necesitan de un mecanismo de diversificación que le permita a las hormigas elegir (en ocasiones) componentes de soluciones que difieran de manera significativa de las componentes de otras soluciones ya construidas; y de esta manera adaptarse a las grandes distancias entre las soluciones que implica esta característica.

La Figura 1 muestra 10000 soluciones generadas por el algoritmo $\mathcal{MAX} - \mathcal{MLN}$ Ant System para la instancia `tai50a` de QAPLIB. Las soluciones con distancias grandes respecto del óptimo tienen, en general, valores objetivo bajos; y las soluciones con distancias cercanas al óptimo tienen, en general, valores objetivo más altos. Por lo tanto para hacer frente a este tipo de instancias, se incorpora el uso de la memoria como mecanismo de diversificación adicional.

Los resultados obtenidos hasta el momento han sido alentadores, tanto si se compara el algoritmo $\mathcal{MAX} - \mathcal{MLN}$ Ant System con y sin la memoria externa. Igualmente, cuando se compara nuestra propuesta con otros algoritmos ACO que utilizan otro tipo de memoria externa.

4. Conclusión y Trabajo futuro

Los trabajos futuros a corto plazo involucran un estudio detallado de la mejor configuración de los parámetros que el enfoque de memoria requiere; así como la implementación paralela del algoritmo $\mathcal{MAX} - \mathcal{MLN}$ Ant System para resolver el QAP, con instancias de mayor tamaño.

Una de las líneas a seguir es la aplicación de algoritmos ACO con el enfoque de memoria, aplicados a otros problemas de optimización combinatoria. Asimismo, se planea la aplicación del enfoque de memoria a otros algoritmos ACO eficientes, como el algoritmo Ant Colony System [2].

5. Recursos Humanos

La propuesta particular planteada aquí junto con los estudios derivados de la misma han dado lugar a la formación de becarios de grado y posgrado. En particular se pretende continuar en este y otros temas relacionados a través del contacto realizado por los autores con grupos de la Universidad de Málaga (España) y del INRIA, Lille (Francia).

Referencias

- [1] A. Acan. An external memory implementation in ant colony optimization. In M. Dorigo, M. Birattari, C. Blum, L. M. Gambardella, F. Mondada, and T. Stützle, editors, *ANTS 2004*, volume 3172 of *LNCS*, pages 73–84. Springer, Heidelberg, 2004.
- [2] M. Dorigo and L. M. Gambardella. Ant Colony System: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, 1997.
- [3] M. Dorigo and T. Stützle. *Ant Colony Optimization*. MIT Press, Cambridge, MA, 2004.
- [4] L. M. Gambardella, É. D. Taillard, and M. Dorigo. Ant colonies for the quadratic assignment problem. Technical report, IDSIA-4-97, IDSIA, Lugano, Switzerland, 1997.

- [5] T. C. Koopmans and M. J. Beckmann. Assignment problems and the location of economic activities. *Econometrica*, 25(1):53–76, 1957.
- [6] V. Maniezzo and A. Coloni. The ant system applied to the quadratic assignment problem. *IEEE Transactions on Knowledge and Data Engineering*, 1999.
- [7] V. Maniezzo, A. Coloni, and M. Dorigo. The ant system applied to the quadratic assignment problem. Technical report, IRIDIA/94-28, IRIDIA, Université Libre de Bruxelles, Belgium, 1994.
- [8] A. Misevicius. A tabu search algorithm for the quadratic assignment problem. *Computational Optimization and Applications*, 30(1):95–111, 2005.
- [9] T. Stützle. $MAX - MIN$ for the quadratic assignment problem. Technical report, AIDA-97-4, FG Intellektik, FB Informatik, TU Darmstadt, Germany, 1997.
- [10] T. Stützle. Iterated local search for the quadratic assignment problem. Technical report, AIDA-99-03, FG Intellektik, FB Informatik, TU Darmstadt, 1999.
- [11] T. Stützle. Iterated local search for the quadratic assignment problem. *European Journal of Operational Research*, 174:1519–1539, 2006.
- [12] T. Stützle and H. Hoos. $MAX - MIN$ ant system. *Future Generation Computer Systems*, 16(8):889–914, 2000.
- [13] É. D. Taillard. Robust taboo search for the quadratic assignment problem. *Parallel Computing*, 17:443–455, 1991.
- [14] É. D. Taillard. Comparison of iterative searches for the quadratic assignment problem. *Location Science*, 3(2):87–105, 1995.
- [15] É. D. Taillard and L. M. Gambardella. Adaptive memories for the quadratic assignment problem. Technical report, IDSIA-87-97, IDSIA, Lugano, Switzerland, 1997.
- [16] S. Tsutsui. *cAS*: Ant colony optimization with cunning ants. In T. P. Runarsson *et al.*, editor, *Proc. of the 9th Int. Conf. on Parallel Problem Solving from Nature (PPSN IX)*, volume 4193 of *LNCS*, pages 162–171. Springer, Heidelberg, 2006.
- [17] W. Wiesemann and T. Stützle. Iterated ants: An experimental study for the quadratic assignment problem. In M. Dorigo, L. M. Gambardella, M. Birattari, A. Martinoli, R. Poli, and T. Stützle, editors, *ANTS 2006*, volume 4150 of *LNCS*, pages 179–190. Springer, Heidelberg, 2006.