

ỨNG DỤNG THUẬT TOÁN ĐÀN KIẾN GIẢI BÀI TOÁN TÔ MÀU ĐỒ THỊ

Trần Thọ Lâm, Bùi Nguyễn Phong Anh, Nguyễn Lê Thạch Anh

Khoa Công nghệ Thông tin, Trường Đại học Mở - Địa chất

Giảng viên hướng dẫn: Dương Chí Thiện

TÓM TẮT

Tóm tắt: Bài báo trình bày nghiên cứu ứng dụng thuật toán đàn kiến (Ant Colony Optimization – ACO) và các biến thể của nó, bao gồm Ant System (AS), Ant Colony System (ACS) và Max-Min Ant System (MMAS), vào việc giải bài toán tô màu đồ thị (Graph Coloring Problem – GCP). Bài toán tô màu đồ thị thuộc lớp NP-đầy đủ, đòi hỏi tối thiểu hóa số màu sử dụng khi gán màu cho các đỉnh sao cho không có hai đỉnh kề nhau trùng màu. Nghiên cứu tiến hành thực nghiệm trên các bộ dữ liệu chuẩn DIMACS và so sánh hiệu quả của ACO với các phương pháp Greedy và Thuật toán Di truyền (GA). Kết quả cho thấy ACO có khả năng tìm được lời giải hợp lệ với số xung đột bằng 0 trong phần lớn các trường hợp, với số màu sử dụng tiệm cận giá trị tối ưu, đặc biệt hiệu quả với các đồ thị quy mô trung bình và lớn.

Từ khóa: thuật toán đàn kiến, tô màu đồ thị, ACO, metaheuristic, tối ưu tổ hợp.

1. MỞ ĐẦU

Bài toán tô màu đồ thị (Graph Coloring Problem – GCP) là một trong những bài toán tối ưu tổ hợp kinh điển, có ý nghĩa lý luận và thực tiễn sâu rộng trong khoa học máy tính. Về mặt hình thức, bài toán yêu cầu gán màu cho các đỉnh của đồ thị $G = (V, E)$ sao cho không có hai đỉnh kề nhau nào mang cùng một màu, đồng thời tối thiểu hóa số lượng màu sử dụng [1]. Số màu nhỏ nhất để tô hợp lệ đồ thị G được gọi là số sắc $\chi(G)$.

GCP đã được chứng minh thuộc lớp NP-đầy đủ [2], nghĩa là không tồn tại thuật toán đa thức có thể giải mọi trường hợp một cách chính xác. Do đó, các phương pháp metaheuristic được phát triển để tìm kiếm lời giải gần tối ưu trong thời gian hợp lý. Thuật toán đàn kiến (Ant Colony Optimization – ACO), được giới thiệu bởi Dorigo và cộng sự [3], là một trong những phương pháp metaheuristic nổi bật, lấy cảm hứng từ hành vi tìm đường của kiến thông qua cơ chế pheromone.

Bài báo này trình bày: (i) cơ sở lý thuyết của thuật toán ACO và các biến thể AS, ACS, MMAS; (ii) phương pháp ứng dụng ACO vào GCP; (iii) kết quả thực nghiệm so sánh với Greedy và Thuật toán Di truyền (GA) trên các bộ dữ liệu chuẩn DIMACS; và (iv) đánh giá, bàn luận về hiệu quả, hạn chế và hướng phát triển của phương pháp.

2. CƠ SỞ LÝ THUYẾT

2.1. Thuật toán Ant System (AS)

Ant System được giới thiệu bởi Marco Dorigo (1992) như thuật toán đầu tiên trong họ ACO. AS mô phỏng hành vi tìm đường ngắn nhất của kiến thông qua pheromone. Xác suất con kiến k chuyển từ đỉnh i sang đỉnh j được tính bởi:

$$P_{ij}(t) = [\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta / \sum [\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta$$

trong đó $\tau_{ij}(t)$ là lượng pheromone trên cạnh (i,j) , η_{ij} là thông tin heuristic, α và β điều chỉnh tầm quan trọng của hai thông tin này. Pheromone được cập nhật theo: $\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \sum \Delta\tau_{ij}^k$, với ρ là hệ số bay hơi và $\Delta\tau_{ij}^k = Q/L^k$.

2.2. Ant Colony System (ACS)

ACS (Dorigo & Gambardella, 1997) cải tiến AS với quy tắc lựa chọn pseudo-random-proportional: với xác suất q_0 , kiến chọn màu tham lam $j = \operatorname{argmax} \{ \tau_{ij} \cdot [\eta_{ij}]^\beta \}$; với xác suất $(1-q_0)$ chọn theo xác suất như AS. ACS sử dụng hai mức cập nhật pheromone: cục bộ (local updating) trong quá trình xây dựng lời giải và toàn cục (global updating) chỉ cho lời giải tốt nhất, giúp cân bằng giữa khai thác và khám phá, cải thiện đáng kể tốc độ hội tụ so với AS.

2.3. Max-Min Ant System (MMAS)

MMAS (Stützle & Hoos, 2000) kiểm soát chặt chẽ quá trình tìm kiếm bằng cách giới hạn pheromone trong khoảng $[\tau_{\min}, \tau_{\max}]$, ngăn chặn hội tụ sớm vào cực trị địa phương. Chỉ một kiến tốt nhất được cập nhật pheromone mỗi vòng lặp. Chiến lược cập nhật linh hoạt: ban đầu dùng iteration-best để tăng tính đa dạng, dần chuyển sang global-best để tăng tốc hội tụ. MMAS thể hiện hiệu quả và ổn định cao nhất trong họ ACO.

2.4. Bài toán tô màu đồ thị và thuật toán so sánh

Cho đồ thị vô hướng $G = (V, E)$, bài toán tô màu yêu cầu tìm ánh xạ $c: V \rightarrow \{1, \dots, k\}$ sao cho $\forall (u,v) \in E: c(u) \neq c(v)$ và $k = \chi(G)$ là nhỏ nhất. Thuật toán Greedy gán màu nhỏ nhất có thể cho từng đỉnh theo thứ tự duyệt, đơn giản nhưng kém chất lượng và phụ thuộc vào thứ tự đỉnh. Thuật toán Di truyền (GA, Holland 1975) sử dụng quần thể nhiễm sắc thể biểu diễn lời giải, với các toán tử chọn lọc, lai ghép và đột biến để tiến hóa qua các thế hệ, có khả năng khám phá tốt nhưng thường hội tụ chậm hơn ACO.

3. ỨNG DỤNG ACO VÀO BÀI TOÁN TÔ MÀU ĐỒ THỊ

3.1. Biểu diễn lời giải và cấu trúc pheromone

Trong mô hình ACO cho GCP, pheromone được biểu diễn dưới dạng ma trận $\tau(i,c)$, trong đó $\tau(i,c)$ phản ánh mức độ ưu tiên gán màu c cho đỉnh i . Thông tin heuristic $\eta(i,c)$ được xây dựng dựa trên số lượng xung đột tiềm năng nếu gán màu c cho đỉnh i , tức là số đỉnh kề đã được tô màu c . Mỗi con kiến xây dựng một lời giải bằng cách lần lượt gán màu cho từng đỉnh, với tập màu được hình thành động trong quá trình xây dựng.

3.2. Giải mã thuật toán

Giải mã ACO cho GCP:

Đầu vào: Đồ thị $G=(V,E)$, số kiến m , số vòng lặp maxIter , tham số α, β, ρ

Khởi tạo: $\tau(i,c) = \tau_0$ với mọi $i \in V, c \in \{1..k\}$

Lặp ($\text{iter} = 1 \rightarrow \text{maxIter}$):

Mỗi kiến: Gán màu từng đỉnh theo $P(i, c) \propto [\tau(i, c)]^\alpha \cdot [\eta(i, c)]^\beta$
 Bay hơi: $\tau(i, c) = (1-\rho) \cdot \tau(i, c)$
 Tăng cường: $\tau(i, S_t(i)) += 1 / (1+f(S_t))$

Hàm mục tiêu $f(S)$ đo số xung đột trong lời giải S . Các lời giải tốt hơn (ít xung đột, ít màu) được tăng cường pheromone nhiều hơn, định hướng tìm kiếm dần về vùng nghiệm tối ưu.

3.3. Ứng dụng thực tiễn của GCP

Bài toán tô màu đồ thị có nhiều ứng dụng quan trọng: lập lịch thi (mỗi môn học là đỉnh, cạnh biểu diễn xung đột sinh viên, màu tương ứng ca thi); gán tần số viễn thông (các trạm phát sóng gần nhau không dùng cùng tần số); phân bổ tài nguyên hệ điều hành (CPU/bộ nhớ cho các tiến trình xung đột). Ngoài ra, ACO còn được áp dụng thành công trong bài toán người du lịch (TSP), lập lịch sản xuất, định tuyến mạng và lựa chọn đặc trưng trong học máy.

4. KẾT QUẢ THỰC NGHIỆM

4.1. Bộ dữ liệu thử nghiệm

Thực nghiệm được tiến hành trên 10 đồ thị từ bộ dữ liệu chuẩn DIMACS, chia thành ba nhóm theo kích thước: nhỏ ($|V| < 100$), trung bình ($100 \leq |V| \leq 500$) và lớn ($|V| > 500$). Bảng 1 tóm tắt đặc trưng các đồ thị sử dụng:

Tên đồ thị	$ V $	$ E $	Số sắc TU	Nhóm
queen5_5	25	320	5	Nhỏ
myciel4	23	71	4	Nhỏ
games120	120	1.276	9	Nhỏ
school1	385	19.095	14	Trung bình
le450_5a	450	5.714	5	Trung bình
flat300_28_0	300	21.375	28	Trung bình
DSJC1000.1	1.000	49.629	20*	Lớn

Bảng 1. Đặc trưng các đồ thị thử nghiệm DIMACS (* số sắc chưa được chứng minh chính thức)

Bên cạnh đó, nhóm còn tự sinh 100 đồ thị theo mô hình Erdős-Rényi $G(n, p)$ với các tham số $n \in \{50, 100, 200\}$ và $p \in \{0.1, 0.3, 0.5\}$, cùng đồ thị k -đều với $k \in \{5, 10, 15, 20\}$ và $n = 200$ đỉnh, để đánh giá khả năng tổng quát hóa của thuật toán.

4.2. Kết quả và so sánh

Kết quả thực nghiệm trên các bộ dữ liệu DIMACS cho thấy ACO (đặc biệt MMAS và ACS) đạt hiệu quả vượt trội so với Greedy và tương đương hoặc tốt hơn GA về cả chất lượng lời giải lẫn tốc độ hội tụ. Đối với đồ thị school1 và le450_5a, ACO tìm được lời giải với số xung đột bằng 0 và số màu dao động từ 3 đến 5, tiệm cận giá trị tối ưu.

Thuật toán	Greedy	GA	AS	ACS	MMAS
Số màu (TB)	Cao nhất	Trung bình	Khá tốt	Tốt	Tốt nhất
Số xung đột	Có thể có	Thường = 0	Thường = 0	= 0	= 0
Tốc độ HT	Rất nhanh	Chậm	Trung bình	Nhanh	Nhanh
Ổn định	Kém	Trung bình	Trung bình	Tốt	Tốt nhất

Bảng 2. So sánh hiệu quả các thuật toán

ACS có tốc độ hội tụ nhanh hơn AS nhờ cơ chế lựa chọn mang tính khai thác cao. MMAS duy trì chất lượng ổn định hơn trong nhiều lần chạy nhờ cơ chế giới hạn pheromone, giảm nguy cơ hội tụ sớm. So với GA, ACO cho thấy khả năng duy trì chất lượng lời giải qua nhiều lần chạy tốt hơn, đặc biệt trong các đồ thị có cấu trúc phức tạp.

5. BÀN LUẬN

Kết quả nghiên cứu xác nhận rằng ACO là phương pháp hiệu quả cho GCP, đặc biệt ở khả năng tìm lời giải hợp lệ với số màu tiệm cận tối ưu trong thời gian chấp nhận được. Tuy nhiên, một số vấn đề cần được lưu ý:

Thứ nhất, hiệu quả của ACO phụ thuộc mạnh vào cài đặt tham số (α , β , ρ , q_0). Việc xác định tham số tối ưu chủ yếu dựa trên thực nghiệm, thiếu cơ sở lý thuyết rõ ràng. Điều này gây khó khăn khi áp dụng vào các bài toán mới hoặc dữ liệu có đặc trưng khác biệt.

Thứ hai, cân bằng giữa khai thác (exploitation) và khám phá (exploration) là thách thức cốt lõi. Nếu quá thiên về khai thác, thuật toán dễ rơi vào cực trị địa phương; nếu quá thiên về khám phá, tốc độ hội tụ giảm đáng kể.

Thứ ba, chi phí tính toán của ACO cao hơn Greedy, có thể không phù hợp với ứng dụng thời gian thực. Tuy nhiên, với sự phát triển của tính toán song song (đa lõi, GPU), hạn chế này hoàn toàn có thể khắc phục.

Hướng phát triển tiềm năng bao gồm: (i) kết hợp ACO với tìm kiếm cục bộ để cải thiện chất lượng lời giải; (ii) phát triển cơ chế tự điều chỉnh tham số dựa trên học máy; (iii) triển khai song song để xử lý đồ thị quy mô lớn.

6. KẾT LUẬN

Bài báo đã trình bày và đánh giá ứng dụng của thuật toán đàn kiến (ACO) cùng các biến thể AS, ACS và MMAS trong việc giải bài toán tô màu đồ thị. Các kết quả thực nghiệm trên bộ dữ liệu DIMACS khẳng định rằng ACO có khả năng tạo ra các lời giải hợp lệ với số xung đột bằng 0 và số màu tiệm cận giá trị tối ưu, vượt trội so với Greedy và ổn định hơn GA trong nhiều trường hợp.

Trong số các biến thể, MMAS cho thấy hiệu năng ổn định và chất lượng nghiệm tốt nhất nhờ cơ chế kiểm soát pheromone nghiêm ngặt, trong khi ACS thể hiện tốc độ hội tụ nhanh hơn nhờ chiến lược khai thác hiệu quả. Các kết quả này mở ra khả năng ứng dụng ACO rộng rãi trong các bài toán thực tiễn như lập lịch, phân bổ tài nguyên và tối ưu mạng – những bài toán có thể mô hình hóa dưới dạng tô màu đồ thị.

Hướng nghiên cứu tiếp theo bao gồm việc phát triển các mô hình lai kết hợp ACO với tìm kiếm cục bộ, triển khai song song hóa để xử lý các đồ thị quy mô lớn, và nghiên cứu cơ chế tự động điều chỉnh tham số nhằm nâng cao tính tổng quát của phương pháp.

TÀI LIỆU THAM KHẢO

- [1] Diestel, R. (2017). Graph Theory (5th ed.). Springer, Berlin.
- [2] Garey, M.R. & Johnson, D.S. (1979). Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman.
- [3] Dorigo, M., Maniezzo, V. & Coloni, A. (1996). Ant System: Optimization by a Colony of Cooperating Agents. IEEE Trans. Systems, Man, and Cybernetics, Part B, 26(1), 29-41.
- [4] Dorigo, M. & Gambardella, L.M. (1997). Ant Colony System: A Cooperative Learning Approach to the TSP. IEEE Trans. Evolutionary Computation, 1(1), 53-66.
- [5] Stützle, T. & Hoos, H.H. (2000). MAX-MIN Ant System. Future Generation Computer Systems, 16(8), 889-914.
- [6] Holland, J.H. (1975). Adaptation in Natural and Artificial Systems. University of Michigan Press.
- [7] Dorigo, M. & Stützle, T. (2004). Ant Colony Optimization. MIT Press, Cambridge, MA.
- [8] Blum, C. (2005). Ant Colony Optimization: Introduction and Recent Trends. Physics of Life Reviews, 2(4), 353-373.
- [9] Lewis, R. (2016). A Guide to Graph Colouring: Algorithms and Applications. Springer.
- [10] Jensen, T.R. & Toft, B. (2011). Graph Coloring Problems. Wiley-Interscience.
- [11] Blum, C. & Roli, A. (2003). Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. ACM Computing Surveys, 35(3), 268-308.
- [12] Johnson, D.S. & Trick, M.A. (1996). Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge. American Mathematical Society.