APPROXIMATION ALGORITHM BASED ON MULTI-START METHOD TO SOLVE THE MINIMUM S-CLUB COVER

Pham Dinh Thanh
Tay Bac University

ARTICLE INFO

Received: 23/11/2023
Revised: 27/12/2023
Published: 27/12/2023

KEYWORDS

Minimum s-club cover
Multi-start method
Covering graph
Greedy algorithm
Neighbor

ABSTRACT

Covering graph is one of the classical topics in theoretical research in computer science. For research on the vertex cover of graphs, the *s*-club model is widely used in social network analysis, protein interaction analysis, etc., where the Minimum *s*-club cover problem has recently received attention. Although there is an approximate algorithm to solve the Minimum *s*-club cover problem, this algorithm can only be applied to the case $s = 2$. In addition, the previous algorithm uses a greedy strategy, so the quality of the obtained solution is not good and depends on the input graph. Therefore, this study proposes an approximate algorithm based on multi-start method to solve the Minimum *s*-club cover problem. To improve the quality of the received solutions, this study also proposes a greedy strategy to find the best club at each step, as well as a method for creating and evaluating neighbors of the current solution. The effectiveness of the proposed algorithm is demonstrated through comparison with a previously published approximation algorithm on two different data sets from the DIMACS library. Experimental results show that the proposed algorithm finds better solutions in one-third of the tested cases, and the two algorithms are equal in two-thirds of the tested cases.

THUẬT TOÁN GẦN ĐÚNG DỰA TRÊN PHƯƠNG THỨC KHỞI TẠO LỜI GIẢI NHIỀU LẦN GIẢI BÀI TOÁN MINIMUM S-CLUB COVER

Phạm Đình Thành
Trường Đại học Tây Bắc

THÔNG TIN BÀI BÁO

Ngày nhận bài: 23/11/2023
Ngày hoàn thiện: 27/12/2023
Ngày đăng: 27/12/2023

TỪ KHÓA

Minimum s-club cover
Phương thức đa khởi tạo
Phủ đồ thị
Thuật toán tham lam
Lân cận

TÓM TẮT

Phủ đồ thị là một trong các chủ đề cơ bản trong nghiên cứu lý thuyết về khoa học máy tính. Đối với hướng nghiên cứu về phủ tập đỉnh của đồ thị, mô hình *s*-club được ứng dụng nhiều trong phân tích mạng xã hội, phân tích tương tác protein,... trong đó bài toán Minimum *s*-club cover nhận được sự quan tâm nghiên cứu trong thời gian gần đây. Mặc dù đã có thuật toán gần đúng để giải bài toán Minimum *s*-club cover, tuy nhiên, thuật toán này chỉ được áp dụng cho trường hợp $s = 2$ và do sử dụng chiến lược tham nên chất lượng lời giải của thuật toán phụ thuộc nhiều vào đồ thị đầu vào. Do đó, nghiên cứu này đề xuất thuật toán gần đúng dựa trên việc khởi tạo lời giải nhiều lần để giải bài toán Minimum *s*-club cover. Để nâng cao chất lượng lời giải tìm được, nghiên cứu còn đề xuất chiến lược tham lam để tìm club tốt nhất tại mỗi bước, cũng như phương pháp tạo lân cận và đánh giá lân cận của lời giải hiện tại. Hiệu quả của thuật toán đề xuất được chứng minh qua việc so sánh với thuật toán gần đúng được công bố trước đây trên hai tập dữ liệu khác nhau từ thư viện DIMACS. Kết quả thực nghiệm cho thấy, thuật toán đề xuất tìm được lời giải tốt hơn trên một phần ba bộ dữ liệu và tìm được lời giải bằng nhau trên hai phần ba bộ dữ liệu.

DOI: https://doi.org/10.34238/tnu-jst.9280

Email: thanhpd05@gmail.com, thanhpd@utb.edu.vn

1. Giới thiệu

Các bài toán bao phủ đồ thị là một trong những chủ đề cổ điển và cơ bản của lý thuyết đồ thị. Chủ đề này cũng đóng một vai trò quan trọng trong nhiều mô hình toán học cho các ứng dụng thực tế khác nhau. Có hai biến thể khác nhau liên quan đến việc bao phủ một đồ thị là bao phủ các cạnh và bao phủ các đỉnh của đồ thị. Cả hai biến thể đều thu hút rất nhiều sự chú ý của giới khoa học và là đối tượng nghiên cứu phong phú.

Mô hình *s*-club được giới thiệu bởi Mokken năm 1979 [1] nghiên cứu về việc bao phủ tập các đỉnh của đồ thị. Mô hình *s*-club là một mô hình toán học cơ bản, ban đầu được thiết lập phục vụ việc nghiên cứu khai phá thông tin trong đồ thị [2]. Hiện nay, mô hình *s*-club có nhiều ứng dụng như: trong việc phân tích các tương tác protein dựa trên việc phân cụm một mạng với số *s*-club ít nhất [3]. Một cách tiếp cận tương tự đã được xem xét trong nghiên cứu [4] để phân tích mạng xã hội. Mô hình *s*-club cũng đã được áp dụng để biến đổi các đồ thị thành các cụm rời rạc (*s*-club) [5]–[7].

Mô hình *s*-club có nhiều dạng khác nhau, một trong những mô hình được nghiên cứu sớm nhất là bài toán tìm 2-club có kích thước lớn nhất, hay tổng quát hơn là bài toán tìm *s*-club có kích thước lớn nhất (Maximum *s*-club). Bài toán Maximum *s*-club là NP-Khó khi $s \geq 1$ [8]. Nghiên cứu [9] cũng chỉ ra rằng với đồ thị đầu vào G=(V, E) có thể tìm được lời giải gần đúng của bài toán Maximum *s*-club với hệ số là $|V|^{1/2}$ với $s \geq 2$ và không thể tìm được lời giải gần đúng với hệ số $|V|^{1/2-\epsilon}$ với $\epsilon > 0$ và $s \geq 2$ trừ khi P = NP trong đó |V| là số đỉnh của đồ thị đầu vào.

Một bài toán khác của mô hình *s*-club có yêu cầu tìm một tập nhiều nhất *r* tập con *s*-club không giao nhau (mỗi tập *s*-club có ít nhất 2 đỉnh) sao cho tập này phủ nhiều nhất số đỉnh của đồ thị. Điểm khác của bài toán này so với Maximum *s*-club là các *s*-club phải không giao nhau. Các nghiên cứu [10], [11] đã chứng minh bài toán này là NP-Khó.

Thời gian gần đây, hướng tiếp cận nới lỏng ràng buộc của mô hình *s*-club được áp dụng vào bài toán phủ đồ thị. Một trong những mô hình được đề xuất nghiên cứu là bài toán Minimum *s*-club cover. Bài toán Minimum *s*-club cover tìm một tập $\{C_1, C_2, \dots, C_h\}$ các tập con của các đỉnh của đồ thị (các tập này có thể không giao nhau) sao cho hợp của các tập này chứa tất cả các đỉnh của đồ thị và đồ thị con cảm sinh bởi mỗi tập con C_i ($1 \leq i \leq h$) có đường kính không lớn hơn *s*. Trong nghiên cứu [12], các tác giả đã xét các trường hợp *s* = 2, 3 của bài toán Minimum *s*-club cover và đã chỉ ra rằng, bài toán quyết định tương ứng với các trường hợp nay là NP-Đầy đủ khi xác định câu trả lời của câu hỏi "có thể phủ một đồ thị với hai 3-club hay không?", "có thể phủ một đồ thị với ba 2-club hay không?". Cũng trong nghiên cứu này các tác giả đã chỉ ra rằng, trên đồ thị *G=(V, E)* không thể tìm được lời giải gần đúng của bài toán Minimum 3-club cover có hệ số $|V|^{1-\epsilon}$ với $\varepsilon > 0$; bài toán Minimum 2-club cover không thể tìm được lời giải gần đúng có hệ số $|V|^{1/2-\epsilon}$.

Trong những năm gần đây, nhiều phương pháp gần đúng được đề xuất để giải các bài toán tối ưu tổ hợp. Trong đó, một vài phương pháp sử dụng các chiến lược chỉ giải được một số bài toán nhất định và rất khó để áp dụng cho các bài toán khác; một số phương pháp dựa trên cấu trúc được thiết kế để có thể sử dụng các phương pháp đã có để giải các bài toán khác nhau [13]. Phương pháp khởi tạo lời giải nhiều lần (Multi-Start Methods - *MSM*) được thiết kế để có thể sử dụng kết hợp nhiều phương pháp khác nhau để tìm lời giải bài toán. Do kết hợp nhiều phương pháp nên *MSM* có thể khai thác được thế mạnh của các thuật toán khác nhau [13]. Thuật toán *MSM* lặp lại hai bước chính: trong bước 1 thuật toán sẽ khởi tạo lời giải; trong bước 2 thuật toán sẽ cải thiện lời giải được tạo trong bước 1.

Mặc dù trong nghiên cứu [12], các tác giả đã đề xuất thuật toán gần đúng Club-Cover-Approx (ký hiệu *CCA*) tìm được lời giải bài toán Minimum 2-club cover với hệ số $2|V|^{1/2}log^{3/2}|V|$, trong đó *V* là tập đỉnh của đồ thị, |*V*| là số đỉnh của tập *V*. Tại mỗi bước, thuật toán *CCA* sẽ tìm một 2-club là tập lớn nhất các đỉnh được chọn. Mặc dù thuật toán *CCA* có ưu điểm về thời gian

thực hiện và đơn giản về ý tưởng lẫn cài đặt thuật toán, tuy nhiên do dựa trên chiến lược tham lam và chất lượng lời giải phụ thuộc nhiều vào bậc của các đỉnh của đồ thị đầu vào nên chất lượng lời giải mà thuật toán chưa thực sự tốt. Bên cạnh đó, thuật toán *CCA* chỉ giải bài toán Minimum 2-club cover, không áp dụng để giải được các bài toán Minimum s-club cover với $s > 2$. Nhằm sử dụng ưu điểm của phương pháp *MSM* để cải thiện chất lượng lời giải tìm được, nghiên cứu này đề xuất thuật toán gần đúng (ký hiệu là *I-MSM*) để giải bài toán Minimum *s*-club cover với $s \geq 2$ dựa trên trên phương pháp *MSM* [14], [15]. Để áp dụng phương pháp *MSM* nghiên cứu đã đề xuất:

- Chiến lược tham lam để tìm club tốt nhất tại mỗi bước.
- Đề xuất phương pháp tạo lân cận của lời giải hiện tại.
- Định nghĩa hàm đánh giá cá thể khác hàm mục tiêu của bài toán Minimum *s*-club cover để so sánh các lân cận của lời giải đang xét.

Các phần còn lại của nghiên cứu được tổ chức như sau: phần 2 trình bày về Thuật toán đề xuất; phần 3 trình bày các Kết quả thực nghiệm và đánh giá thuật toán; phần Kết luận của nghiên cứu được trình bày trong phần 4.

2. Thuật toán đề xuất

Phần này sẽ trình bày về bài toán Minimum *s*-club cover và thuật toán đề xuất *I-MSM* giải bài toán Minimum *s*-club cover.

2.1. Bài toán nghiên cứu

Cho đơn đồ thị vô hướng *G=(V, E)* với tập đỉnh *V*, tập cạnh *E*. Mỗi tập đỉnh $S \subseteq V$, ký hiệu *G[S]* là đồ thị con cảm sinh bởi tập *S*. Cho hai đỉnh $u, v \in V$, khoảng cách giữa *u* và *v* trên đồ thị *G*, ký hiệu bởi $d_G(u, v)$ là số cạnh trên đường ngắn nhất nối từ *u* tới *v*.

Định nghĩa 1 (đường kính của đồ thị)

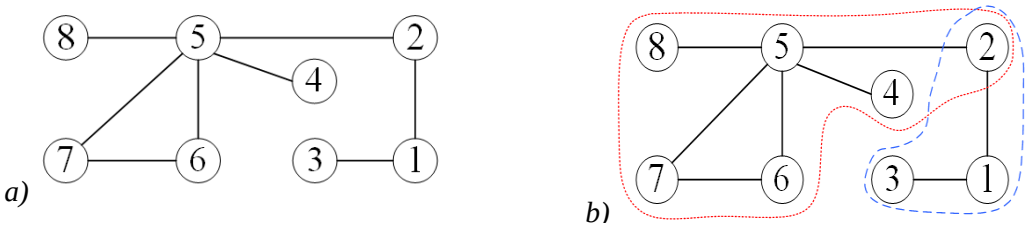
Đường kính của đơn đồ thị vô hướng *G=(V, E)* (ký hiệu *diam(G)*) là khoảng cách lớn nhất giữa hai đỉnh bất kỳ của tập *V*.

Định nghĩa 2 (*s*-club)

Cho đơn đồ thị vô hướng *G=(V, E)* và tập con $U \subseteq V$, đồ thị con *G[U]* là một *s*-club nếu đường kính của *G[U]* lớn nhất là *s* ($diam(G[U]) \leq s$).

Định nghĩa 3 (bài toán Minimum *s*-club cover)

Cho đơn đồ thị vô hướng *G=(V, E)* và một số nguyên $s \geq 2$, hãy tìm tập $C = \{V_1, V_2, \ldots, V_h\}$ nhỏ nhất sao cho với mỗi i ($1 \leq i \leq h$) và tập $V_i \subseteq V$ thì $G[V_i]$ là một *s*-club và với mỗi đỉnh $v \in V$, tồn tại một tập V_j ($1 \leq j \leq h$) sao cho $v \in V_j$.



Hình 1. *Minh hoạ lời giải bài toán Minium s-club cover*

(a) đồ thị đầu vào và (b)một lời giải bài toán Minium 2-club cover

0 minh hoạ ví dụ về lời giải của bài toán Minimum *s*-club cover với 0(a) là đồ thị đầu vào. Tập các đỉnh $V_1 = \{4, 5, 6, 7, 8\}$ là một 2-club do đường đi ngắn nhất nối hai đỉnh bất kỳ của tập V_1 có độ dài lớn nhất là 2. 0(b) minh hoạ một lời giải của bài toán Minimum 2-club cover với hai tập $V_1 = \{4, 5, 6, 7, 8\}$ và $V_2 = \{1, 2, 3\}$.

2.2. Lược đồ thuật toán

Thuật toán đề xuất thực hiện lặp lại các bước chính sau đây:

- Bước 1: Sử dụng thuật toán *CCA* để khởi tạo lời giải của bài toán Minimum *s*-club cover, lời giải này được đặt là lời giải hiện thời.

- Bước 2: Lặp lại các bước sau:

+ Bước 2.1. Tạo danh sách các lân cận tốt nhất của lời giải hiện thời (mô tả chi tiết trong phần 2.3). Nếu không tìm được lân cận nào tốt hơn lời giải hiện thời thì thoát khỏi vòng lặp.

+ Bước 2.2. Chọn ngẫu nhiên một trong các lời giải từ Bước 2.1 đặt làm lời giải hiện thời.

+ Bước 2.3. Nếu sau một số lần xác định trước lời giải tốt nhất tìm được không được cải thiện thì thoát khỏi vòng lặp.

Input: - Đồ thị $G = (V, E)$;
- Số nguyên dương $s >= 2$;
- Hệ số thoát khỏi vòng lặp Δ_{minper};
- Hệ số lặp tìm lân cận Δ_{size};
- Hệ số số tối đa số lần lời giải không cải thiện được $P_{unchange}$;

Output: Lời giải của bài toán min s-club cover;

begin
  $sol_{bn} \leftarrow \emptyset$ ▷ Lời giải tốt nhất tìm được;
  repeat
    $sol_p \leftarrow$ Tạo lời giải bằng thuật toán CCA;
    $f(sol_p) \leftarrow$ Đánh giá lời giải sol_p;
    $sol_{bn} \leftarrow sol_p$ ▷ Lưu lân cận tốt nhất là lời giải đang xét;
    $move \leftarrow 1$;
    $count\_exit \leftarrow 1$;
    for $i \leftarrow 0$ **to** $|V| * s * \Delta_{size}$ **by** 1 **do**
      $v \leftarrow$ Chọn ngẫu nhiên một đỉnh thuộc tập V;
      $sol \leftarrow sol_p$;
      $idx \leftarrow$ find_best_club(sol, v) ▷ Tìm club tốt nhất để chuyển v tới;
      if *(không tìm được club tốt nhất)* **then** break;
      $sol \leftarrow$ Move(sol, v, idx) ▷ Chuyển đỉnh v sang club thứ idx;
      $f(sol) \leftarrow$ Đánh giá lời giải sol;
      `/* Đặt sol là lời giải hiện thời */`
      if $(f(sol) < f(sol_p))$ **then**
        $sol_p \leftarrow sol$;
        move++;
      `/* Cập nhật lân cận tốt nhất */`
      if $(f(sol_p) < f(sol_{bn}))$ **then** $sol_{bn} \leftarrow sol_p$;
      `/* Cập nhật lời giải tốt nhất tìm được */`
      if *(số club của sol_{bn} nhỏ hơn số club của sol_{best})* **then** $sol_{best} \leftarrow sol_{bn}$;
      $probalility\_exit \leftarrow move/(|V| * s * \Delta_{size})$;
      if *(probalility_exit $< \Delta_{minper}$)* **then**
        $sol_{bn} \leftarrow sol_p$;
        $count\_exit++$;
        if *(count_exit $> P_{unchange} * |V| * s * \Delta_{size}$)* **then** break;
      else count_exit $\leftarrow 1$;
  until *(điều kiện dừng chưa thỏa mãn)*;
  return sol_{best};

Thuật toán 1. *Mã giả các bước chính của thuật toán I-MSM*

Thuật toán được cài đặt để dừng khi số lần đánh giá được sử dụng lớn hơn số lần đánh giá được xác định trước. Nghiên cứu này tiến hành thực nghiệm với số lần đánh giá lớn nhất là 50.000 lần.

Chi tiết hơn về các bước của thuật toán đề xuất được minh hoạ bằng mã giả trong Thuật toán 1 với dòng lệnh thứ 4 tạo lời giải bằng thuật toán *CCA*. Thuật toán sẽ lặp lại nhiều nhất $|V|*s*\triangle_{size}$ lần để xét lân cận của lời giải hiện thời. Một lân cận được tạo thành bằng cách chọn ngẫu nhiên một đỉnh của lời giải đang xét và di chuyển sang một club khác. Dòng lệnh thứ 12 thực hiện việc tìm club tốt nhất bằng cách đánh giá các lời giải trung gian nhận được tương ứng khi thử chuyển đỉnh *v* sang club khác; nếu tìm được lời giải trung gian là hợp lệ thì lời giải này được đánh giá bằng số đỉnh trong club mà đỉnh *v* được chuyển tới.

Điểm lưu ý trong thuật toán *I-MSM* là việc so sánh một lời giải với một lân cận mới sẽ được thực hiện thông qua sử dụng hàm đánh giá f(.) tại các dòng lệnh thứ 5, 15 (trình bày chi tiết trong phần 0); còn việc so sánh lân cận mới với lời giải tốt nhất tìm được tại dòng lệnh thứ 20 được thực hiện thông qua số club của mỗi lời giải (hàm mục tiêu của bài toán Minimum *s*-club cover).

2.3. Xác định lân cận của lời giải đang xét

Với mỗi lời giải đang xét, thuật toán *I-MSM* sẽ lặp lại việc tìm lân cận mới và cập nhật lời giải đang xét nếu lân cận tìm được tốt hơn. Một lân cận của lời giải đang xét được tạo bằng cách di chuyển một đỉnh sang một club khác. Phương thức *find_best_club(sol, v)* tại dòng lệnh thứ 12 của Lược đồ thuật toán *I-MSM* sẽ xác định club tốt nhất được tạo thành từ lời giải *sol* khi chuyển đỉnh *v* của lời giải này sang club khác. Các lân cận được đánh giá thông qua hàm đánh giá được xác định như sau:

- Nếu lân cận là lời giải hợp lệ thì hàm đánh giá bằng số lượng đỉnh trong club mà đỉnh *v* chuyển tới.
- Nếu lân cận là lời giải không hợp lệ thì hàm đánh giá bằng vô cùng.
- Trong trường hợp có nhiều lân cận là lời giải hợp lệ và có cùng giá trị hàm đánh giá thì thuật toán sẽ lựa chọn ngẫu nhiên một lân cận.

Thuật toán 2 minh hoạ các bước chính của thuật toán *find_best_club*.

Input: - Đồ thị $G = (V, E)$;
- Số nguyên dương $s >= 2$;
- Đỉnh $v \in V$;
- Lời giải đang xét *solution*;

Output: Chỉ số của club tốt nhất khi chuyển đỉnh v sang;

```
1  begin
2      cost_max ← −∞;
3      foreach (club cl không chứa đỉnh v) do
4          Thêm đỉnh v vào club cl;
5          radius ← Tính đường kính của đồ thị con G[cl] cảm sinh tập cl;
6          if (radius < s) then
7              cost_cl ← |cl|          ▷ Chi phí của club cl khi thêm đỉnh v hợp lệ;
8              if (cost_max < cost_cl) then
9                  cost_max ← cost_cl;
10         else cost_cl ← −∞      ▷ Chi phí của club cl sau khi thêm đỉnh v không hợp lệ ;
11         Xoá đỉnh v khỏi club cl;
12     if (Không tìm được club hợp lệ sau khi thêm đỉnh v) then
13         return-1;
14     else
15         lstBestClub ← Tạo danh sách club có chi phí bằng cost_max;
16         return club được chọn ngẫu nhiên từ lstBestClub;
```

Thuật toán 2. *Các bước chính của thuật toán find_best_club*

Thuật toán *find_best_club* đánh giá mỗi club có đường nhỏ hơn *s* bằng số lượng đỉnh trong club đó sẽ dẫn tới việc ưu tiên lựa chọn club có số lượng đỉnh lớn để chuyển đỉnh *v* tới.

2.4. Đánh giá lời giải

Khác với việc đánh giá lời giải thường sử dụng để giải các bài toán là tính trực tiếp bằng giá trị hàm mục tiêu, nghiên cứu xây dựng cách đánh giá lời giải *sol* thông qua hai yếu tố: số club của lời giải *sol* và số đỉnh của club có số lượng đỉnh nhỏ nhất. Cá thể *sol* được đánh giá qua công thức sau:

$$f(sol) = num\_club + min\_vertex/|V| \quad (1)$$

trong đó: *num_club* là số club của lời giải *sol*, nếu *sol* là lời giải không hợp lệ thì *num_club* sẽ bằng số đỉnh của đồ thị đầu vào (khi ấy mỗi đỉnh sẽ là một club); *min_vertex* là số lượng đỉnh trong club có ít đỉnh nhất; $|V|$ là số đỉnh của đồ thị đầu vào. Do tỉ lệ $\frac{min\_vertex}{|V|} < 1$ nên khi so sánh hai lời giải thông qua hàm đánh giá *f(sol)*, thuật toán sẽ so sánh số club của hai lời giải trước, nếu hai lời giải có cùng số club thì lời giải nào có tỉ lệ *min_vertex*/$|V|$ nhỏ hơn sẽ được coi là tốt hơn.

Việc đánh giá thông qua hai yếu tố có ưu điểm là cá thể nào có cùng số club nhưng có club nhỏ nhất có ít đỉnh hơn sẽ được đánh giá tốt hơn. Nguyên nhân dẫn tới đề xuất cách đánh giá này là do nếu hai lời giải có cùng số club thì lời giải nào có club có số lượng đỉnh ít nhất nhỏ hơn sẽ có khả năng giảm được số club nhanh hơn bằng cách chuyển các đỉnh thuộc club bé nhất sang các club khác. Hay nói cách khác, lời giải nào có club nhỏ nhất với số đỉnh ít hơn sẽ có khả năng cải thiện hàm mục tiêu nhanh hơn nên được đánh giá cao hơn.

3. Kết quả thực nghiệm

3.1. Dữ liệu thực nghiệm và tiêu chí đánh giá

Nghiên cứu sử dụng hai tập dữ liệu (ký hiệu là Type 1 và Type 2) từ thư viện DIMACS [16], [17] để đánh giá thuật toán *I-MSM*. Các bộ dữ liệu được lựa chọn để tiến hành thực nghiệm có số đỉnh nhỏ hơn 500. Thông tin chính về các tập dữ liệu được mô tả trong các 0 và 0 với ký hiệu $|V|$, $|E|$ và *GD (Graph density)* lần lượt là số đỉnh, số cạnh và mật độ (tỉ lệ giữa số cạnh và số cạnh nhiều nhất mà một đồ thị có thể có) của đồ thị.

Nghiên cứu tập trung phân tích các tiêu chí chất lượng lời giải tìm được của các thuật toán theo giá trị trung bình cộng (*Avg*) chi phí của hàm mục tiêu và giá trị tốt nhất tìm được trong các lần thực hiện (*BF*). Giá trị *BF* thể hiện số miền *s*-club ít nhất cần thiết để phủ hết các đỉnh của đồ thị.

Bảng 1. *Thông tin chính về các bộ dữ liệu thuộc tập Type 1*

Bộ dữ liệu	|V|	|E|	DG
karate	34	78	0,139
chesapeake	39	170	0,229
dolphins	62	159	0,084
lesmis	77	254	0,087
adjnoun	112	425	0,068
football	115	613	0,094
jazz	198	2742	0,141
celegansneural	297	2148	0,049
celegans metabolic	453	2025	0,02
email	1133	5451	0,009
polblogs	1490	16715	0,015
polbooks	1490	16715	0,015
netscience	1589	2742	0,002
power	4941	6594	0,001

Bảng 2. *Thông tin chính về các bộ dữ liệu thuộc tập Type 2*

Bộ dữ liệu	|V|	|E|	DG	Bộ dữ liệu	|V|	|E|	DG
johnson8-2-4	28	210	0,56	san400_0.7_2	400	55860	0,7
MANN_a9	45	918	0,93	sanr400_0.7	400	55869	0,70
hamming6-4	64	704	0,35	sanr400_0.5	400	39984	0,50
hamming6-2	64	1824	0,9	san400_0.7_1	400	55860	0,7
johnson8-4-4	70	1855	0,77	p_hat500-2	500	62946	0,50
johnson16-2-4	120	5460	0,76	p_hat500-1	500	31569	0,25
C125.9	125	6963	0,9	p_hat300-1	300	10933	0,24
keller4	171	9435	0,65	p_hat300-2	300	21928	0,49
c-fat200-5	200	8473	0,43	p_hat300-3	300	33390	0,75
brock200_1	200	14834	0,75	san400_0.5_1	400	39900	0,5
brock200_2	200	9876	0,50	gen200_p0.9_55	200	17910	0,9
brock200_3	200	12048	0,61	san200_0.9_3	200	17910	0,9
brock200_4	200	13089	0,66	san200_0.9_2	200	17910	0,9
gen200_p0.9_44	200	17910	0,9	C250.9	250	27984	0,9
san200_0.9_1	200	17910	0,9	gen400_p0.9_55	400	71820	0,9
hamming8-2	256	31616	0,97				

3.2. Môi trường và tham số thực nghiệm

Thuật toán *I-MSM* và *CCA* được lập trình bằng ngôn ngữ lập trình C#. Với mỗi bộ dữ liệu, thuật toán *I-MSM* sẽ được thực nghiệm 20 lần trên máy tính cài đặt hệ điều hành Microsoft Windows 10 với cấu hình: CPU - Intel Xeon E5620, RAM - 8GB. Các tham số sử dụng trong thuật toán *I-MSM* được mô tả như trong 0.

Bảng 3. *Bảng tóm tắt các tham số thực nghiệm thuật toán I-MSM*

Tham số	Giá trị	Tham số	Giá trị
Số lần đánh giá:	50.000	Δ_{minper}	0,01
Δ_{size}	8	$P_{unchange}$	0,03

3.3. Kết quả thực nghiệm

0 và 0 trình bày kết quả giữa thuật toán *I-MSM* và thuật toán *CCA* trên các bộ dữ liệu thuộc tập Type và Type 2 với *BF* và *Avg* lần lượt ký hiệu giá trị tốt nhất và giá trị trung bình của lời giải tìm được. Dữ liệu trong cột *CCA* trong hai bảng kết quả trên được định dạng đậm nghĩa là trên bộ dữ liệu đó thuật toán *CCA* có kết quả kém hơn kết quả trung bình của thuật toán *I-MSM*; Tương tự, dữ liệu tại mỗi dòng trong cột *BF* của thuật toán *I-MSM* được định dạng nghiêng đậm nghĩa là giá trị tốt nhất của lời giải của bộ dữ liệu cùng dòng tương ứng tìm được bởi thuật toán *I-MSM* tốt hơn lời giải tìm được bởi thuật toán *CCA*.

Bảng 4. *Kết quả so sánh giữa thuật toán I-MSM và CCA trên các bộ dữ liệu thuộc tập Type 1*

Bộ dữ liệu	I-MSM		CCA
	BF	Avg	
adjnoun	19	19,0	19
celegansneural	***5***	5,0	**32**
celegans_metabolic	32	32,0	32
chesapeake	3	3,0	3
dolphins	17	17,0	17
email	229	229,0	229
football	15	15,0	15
jazz	14	14,0	14
karate	4	4,0	4
lesmis	***3***	3,0	**10**
netscience	***322***	322,0	**544**

Kết quả so sánh trong 0 cho thấy thuật toán *I-MSM* tìm được lời giải tốt hơn *CCA* trên 3 bộ dữ liệu; hai thuật toán tìm được lời giải có giá trị hàm mục tiêu bằng nhau trên 8 bộ dữ liệu.

Kết quả so sánh giữa hai thuật toán trong 0 cho thấy thuật toán *I-MSM* tìm được lời giải tốt hơn thuật toán *CCA* trên 11 bộ dữ liệu; hai thuật toán tìm được lời giải có giá trị hàm mục tiêu bằng nhau trên 21 bộ dữ liệu.

Bảng 5. *Kết quả so sánh giữa thuật toán I-MSM và CCA trên các bộ dữ liệu thuộc tập Type 2*

Bộ dữ liệu	I-MSM		CCA	Bộ dữ liệu	I-MSM		CCA
	BF	Avg			BF	Avg	
johnson8-2-4	***1***	1,7	**3**	san200_0.9_1	2	2,0	2
hamming6-4	4	4,0	4	gen200_p0.9_55	2	2,0	2
MANN_a9	***1***	1,0	**2**	san200_0.9_3	2	2,0	2
c-fat200-1	13	13,0	13	san200_0.9_2	2	2,0	2
hamming6-2	***1***	1,1	**2**	p_hat300-2	4	4,0	4
johnson8-4-4	2	2,0	2	C250.9	2	2,0	2
johnson16-2-4	***2***	2,6	**3**	hamming8-2	2	2,0	2
C125.9	***1***	1,0	**2**	p_hat500-1	9	9,0	9
c-fat200-5	3	3,0	3	p_hat300-3	***2***	2,0	**3**
keller4	2	2,0	2	san400_0.5_1	4	4,0	4
brock200_2	***4***	4,0	**5**	sanr400_0.5	***4***	4,0	**5**
p_hat300-1	8	8,0	8	san400_0.7_1	***2***	2,0	**3**
brock200_3	***3***	3,0	**4**	san400_0.7_2	3	3,0	3
brock200_4	3	3,0	3	sanr400_0.7	3	3,0	3
brock200_1	***2***	2,0	**3**	p_hat500-2	4	4,0	4
gen200_p0.9_44	2	2,0	2	gen400_p0.9_55	2	2,0	2

4. Kết luận

Bài toán Minimum *s*-club cover được xuất hiện trong các nghiên cứu về lý thuyết đồ thị dưới dạng các bài toán phân tích tương tác protein, phân tích mạng xã hội... Nghiên cứu này đề xuất thuật toán gần đúng để giải bài toán Minimum *s*-club cover dựa trên chiến lược tham lam với định nghĩa lân cận và hàm đánh giá lời giải mới. Bên cạnh ưu điểm về khả năng có thể tìm lời giải của bài toán Minimum *s*-club cover, thuật toán đề xuất còn tìm được lời giải tốt với thuật toán gần đúng được nghiên cứu trước đây trên các bộ dữ liệu của hai tập dữ liệu lấy từ thư viện DIMACS. Trong thời gian tới, tác giả sẽ nghiên cứu kết hợp thuật toán đề xuất kết hợp với thuật toán dựa trên quần thể để nâng cao chất lượng lời giải của bài toán Minimum *s*-club cover tìm được.

TÀI LIỆU THAM KHẢO/ REFERENCES

[1] R. J. Mokken, "Cliques, clubs and clans," *Qual. Quant.*, vol. 13, no. 2, pp. 161–173, 1979.

[2] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, no. 3–5, pp. 75–174, 2010.

[3] S. Pasupuleti, "Detection of protein complexes in protein interaction networks using n-clubs," presented at t*he European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, Springer, 2008, pp. 153–164.

[4] L. Cavique, A. B. Mendes, and J. M. Santos, "An algorithm to discover the k-clique cover in networks," presented at *the Progress in Artificial Intelligence: 14th Portuguese Conference on Artificial Intelligence*, EPIA 2009, Aveiro, Portugal, October 12-15, 2009. Proceedings 14, Springer, 2009, pp. 363–373.

[5] D. Chakraborty, L. S. Chandran, S. Padinhatteeri, and R. R. Pillai, "Algorithms and complexity of s-club cluster vertex deletion," presented at *the International Workshop on Combinatorial Algorithms*, Springer, 2021, pp. 152–164.

[6] A. Figiel, A.-S. Himmel, A. Nichterlein, and R. Niedermeier, "On 2-Clubs in Graph-Based Data Clustering: Theory and Algorithm Engineering.," presented at *the CIAC*, 2021, pp. 216–230.

[7] H. Liu, P. Zhang, and D. Zhu, "On editing graphs into 2-club clusters," presented at *the Frontiers in Algorithmics and Algorithmic Aspects in Information and Management: Joint International Conference, FAW-AAIM 2012,* Beijing, China, May 14-16, 2012. Proceedings, Springer, 2012, pp. 235–246.

[8] J.-M. Bourjolly, G. Laporte, and G. Pesant, "An exact algorithm for the maximum k-club problem in an undirected graph," *Eur. J. Oper. Res.*, vol. 138, no. 1, pp. 21–28, 2002.

[9] Y. Asahiro, E. Miyano, K. Samizo, and H. Shimizu, "Optimal approximation algorithms for maximum distance-bounded subgraph problems," *Algorithmica*, vol. 80, no. 6, pp. 1834–1856, 2018.

[10] R. Dondi, G. Mauri, and I. Zoppis, "On the tractability of finding disjoint clubs in a network," *Theor. Comput. Sci.*, vol. 777, pp. 243–251, 2019.

[11] P. Zou, H. Li, W. Wang, C. Xin, and B. Zhu, "Finding disjoint dense clubs in a social network," *Theor. Comput. Sci.*, vol. 734, pp. 15–23, 2018.

[12] R. Dondi, G. Mauri, F. Sikora, and Z. Italo, "Covering a graph with clubs," *J. Graph Algorithms Appl.*, vol. 23, no. 2, pp. 271–292, 2019.

[13] M. Gendreau and J.-Y. Potvin, *Handbook of metaheuristics*, vol. 2. Springer, 2010.

[14] R. Martí, J. A. Lozano, A. Mendiburu, and L. Hernando, "Multi-start methods," in *Handbook of heuristics*, Springer, 2018, pp. 155–175.

[15] R. Martí, M. G. Resende, and C. C. Ribeiro, "Multi-start methods for combinatorial optimization," *Eur. J. Oper. Res.*, vol. 226, no. 1, pp. 1–8, 2013.

[16] D. S. Johnson and M. A. Trick, *Cliques, coloring, and satisfiability: second DIMACS implementation challenge*, October 11-13, 1993, vol. 26. American Mathematical Soc., 1996.

[17] D. A. Bader, H. Meyerhenke, P. Sanders, and D. Wagner, *Graph partitioning and graph clustering*, vol. 588. American Mathematical Society Providence, RI, 2013.