New Local Search Strategy for the Minimum s-Club Cover Problem

Thanh Pham Dinh[1], Tuan Anh Do[2*], Son Nguyen Hung[3] and Thai Nguyen Duc[4]
[1]Faculty of Natural Science and Technology, Taybac University, Vietnam
[2]School of Information and Communication Technology, Hanoi University of Science and Technology, Vietnam
[3]Warsaw University, Poland
[4]Nanyang Technological University, Singapore
Email: thanhpd@utb.edu.vn, anhdt@soict.hust.edu.vn, hungson@gmail.com, ducthai001@e.ntu.edu.sg
*Corresponding author

Keywords: Evolutionary multitask optimization, minimum s-Club cover problem, evolutionary algorithms, greedy strategy, DIMACS

Received: Aug 8, 2025

The Minimum s-Club Cover problem presents significant challenges in social networks and group interactions analysis. Several studies have employed hybrid approaches to solve this problem, notably combining local search techniques with multifactorial evolutionary algorithms. To enhance the computational efficiency of such hybrid methodologies, this study proposes a novel local search method designed specifically for integration with a multifactorial evolutionary framework. The proposed local search algorithm is based on a combination of greedy and exhaustive strategies. The greedy strategy is applied when selecting clubs, while the exhaustive strategy is used when determining the appropriate clubs for vertex relocation. Unlike existing local search methods that operate at the vertex level, the proposed algorithm focuses on manipulating clubs directly. The effectiveness of the proposed approach is evaluated using benchmark datasets from the DIMACS library. Experimental results demonstrate that the algorithm achieves competitive performance, validating its potential in solving the Minimum s-Club Cover problem.

Povzetek: Raziskava obravnava problem prekrivanja grafa z najmanjšim številom s-klubov. Avtorji predlagajo novo lokalno iskalno strategijo, ki deluje na ravni klubov: klube izbirajo s pohlepnim pristopom, premike vozlišč med klubi pa odločajo z izčrpnim preverjanjem. Metoda je zasnovana za vključitev v večopravilni evolucijski okvir in na referenčnih grafih DIMACS izkaže konkurenčno učinkovitost.

1 Introduction

Graph covering problems are a fundamental and classical area of graph theory. This subject is also important in numerous mathematical models applied to various real-world scenarios. There are two distinct types of graph covering: edge covering and vertex covering. Both variants have received considerable research attention and remain active areas of investigation. and are potential research subjects.

The s-Club model, introduced by Mokken in 1979 [25], was designed to explore the coverage of vertex sets within a graph. Created as a fundamental mathematical model, the s-Club model was intended to facilitate research into information mining in graphs [15]. The s-Club model has numerous applications today, including analysing protein interactions by clustering networks with the minimal number of s-Clubs [26]. A comparable methodology has been examined in studies [5, 21, 24, 18] focused on social network analysis. Additionally, the s-Club model has been utilised to convert graphs into discrete clusters, referred to as s-Clubs [7].

The s-Club model exists in various forms, with one of the earliest studied models being the task of identifying the largest 2-Club, or, more broadly, the largest s-Club (maximum s-Club). The Maximum s-Club problem is classified as NP-Hard for s values greater than or equal to 1 [4]. Additionally, another challenge within the s-Club model [12] involves determining a collection of up to r non-overlapping s-Club subsets (each containing a minimum of 2 vertices) such that this collection covers the greatest number of vertices in the graph.

Recently, the approach of relaxing constraints in the s-Club model has been utilised to tackle the graph coverage issue. One of the suggested formulations is known as the Minimum s-Club cover problem [10]. This problem aims to identify a collection $\{C_1, C_2, \ldots, C_h\}$ of vertex subsets from the graph (which may not overlap) so that their combined union encompasses all vertices in the graph, and the subgraphs formed by each subset $C_i (1 \leq i \leq h)$ have a diameter that does not exceed s.

In the research conducted in [11], the researchers investigated the Minimum s-Club Cover problem, specifically for the cases where $s = 2$ and $s = 3$. They proved that for a given graph $G = (V, E)$, approximating the Minimum 3-Club Cover problem within a factor of $|V|^{1-\epsilon}$ for any $\epsilon > 0$ is infeasible. Additionally, it is impossible to achieve an approximate solution for the Minimum 2-Club Cover problem with a coefficient of $|V|^{1-\epsilon}$ for any $\epsilon > 0$.

In [29], the authors propose to apply a local search algorithm to the best individuals of each task in a multifactorial evolutionary algorithm. The local search algorithm will move a randomly selected vertex to the club with the most vertices satisfying the constraints of s-Club. The study also builds a formula to evaluate each club when multiple clubs have the same number of vertices.

Researchers have proposed various algorithms designed to tackle the Minimum s-Club Cover problem, encompassing a range from greedy approaches to memetic algorithms incorporating diverse algorithmic strategies. Among them, the algorithm that combines multifactorial evolution with local search algorithms is a potential direction, capable of obtaining good results. However, local search algorithms are currently focusing on processing vertices; this methodology may prove efficacious for problems of smaller dimensions, but will be less effective when applied to larger-scale issues. Consequently, this study proposes a local search algorithm that can be combined with multifactorial evolutionary algorithms. This local search algorithm has the following characteristics:

- A mechanism for processing clubs in local searches is being introduced. Local search to improve computational efficiency compared to vertex-based approaches.
- Introduce a mechanism for using a random greedy strategy to select clubs, and an exhaustive strategy for moving the vertex. While the random greedy strategy promotes exploration and maintains diversity within the population, the deterministic greedy strategy emphasizes exploitation, thereby enhancing the convergence toward high-quality solutions.

The continuation of this document is structured as follows: Section 2 covers the definitions and notations of the problem, while Section 3 presents the associated works. The suggested techniques are elaborated in Section 4. Section 5 contains the experimental settings, computational results on several test sets, and a performance comparison with other algorithms. Finally, Section 6 includes the conclusions and discussion of extensions.

2 Problem definition and notations

Given an undirected and simple graph $G = (V, E)$. For a vertex set $S \subseteq V$, let $G[S]$ denote the subgraph induced by S. $E(G)$ is the edges set of G. Given two vertices $u, v \in V$, the distance between u and v in G, denoted by $d_G(u, v)$, is the number of edges on the shortest path from u to v.

Definition 2.1 (s-Club) *Given a graph $G = (V, E)$, and a subset $U \subseteq V$, $G[U]$ is an s-Club if it has diameter at most s.*

Notice that an s-club must be a connected graph.

The Minimum s-Club Cover problem (Min s-Club Cover) is stated as follows:

Definition 2.2 *(Minimum s-Club Cover problem)*
Input: *a graph $G = (V, E)$ and an integer $s \geq 2$.*
Output: *a minimum cardinality collection $C = \{V_1, \ldots, V_h\}$ such that, for each i with $1 \leq i \leq h, V_i \subseteq V$, $G[V_i]$ is an s-Club, and for each vertex $v \in V$, there exists a set V_j, with $1 \leq j \leq h$, such that $v \in V_j$.*

The Min s-Club Cover problem in Definition 2.2 can also be expressed as in Table 1.

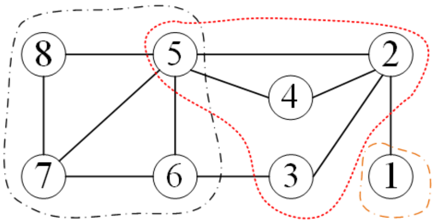


Figure 1: An example of an s-Club and the Minimum s-club Cover problem

Figure 1 depicts an example of a 2-Club and a solution to the Minimum 2-Club Cover problem. The subgraph induced by the vertex set $V' = \{2, 3, 4, 5\}$ is a 2-Club. A solution for the minimum 2-Club cover problem consists of three clubs, induced by the vertex sets $V_1 = \{1\}$, $V_2 = \{2, 3, 4, 5\}$, and $V_3 = \{5, 6, 7, 8\}$. Notice that vertex 5 is covered by both clubs V_2 and V_3.

3 Related works

Graph covering is a classical and extensively studied topic in theoretical computer science. One of the earliest problems explored in this domain is the clique problem. Numerous clique-related combinatorial problems have been investigated, such as the Minimum Clique Cover problem, the Maximum Clique problem [27], and the Minimum Clique Partition problem [6]. Among these, the Minimum Clique Partition problem is particularly well-known; it aims to partition the vertex set of a graph into the smallest possible number of cliques. This problem remains NP-hard even when restricted to specific graph classes. For instance, NP-hardness has been established for planar cubic graphs [6] and unit disk graphs [13]. Moreover, it has been shown that, for any $\epsilon > 0$, the Minimum Clique Partition problem cannot be approximated within a factor of $|V|^{1-\epsilon}$ unless $P = NP$.

However, in network analysis, the requirement of a complete subgraph is often too restrictive. In many cases, not every pair of vertices within a subgraph is connected; this may be due, for instance, to noise or missing data.

To address the limitations of the clique model, various alternative definitions of highly connected subgraphs have been proposed, leading to the concept of a relaxed clique. This work focuses on distance-based relaxations. In a traditional clique, all vertices must be at a distance of exactly

Table 1: Definition of minimum s-Club cover problem

Minimum s-Club Cover problem	
Input:	- An unweighted undirected graph $G = (V, E)$. - An integer $s \geq 2$.
Output:	A collection $C = \{V_1, \ldots, V_h\}, 1 \leq i \leq h, V_i \subseteq V$
Constraints:	- $G[V_i], \forall i = 1, \ldots, h$ is an s-Club. - For each vertex $v \in V$, there exist a set V_j, with $1 \leq j \leq h$, such that $v \in V_j$.
Objective:	$\lvert C \rvert \rightarrow$ min

one from each other. In contrast, this requirement is relaxed by allowing vertices to be at a distance of up to s, where s is an integer greater than one.

A subgraph where every vertex is at a maximum distance of s is called an s-Club (it is important to note that when $s = 1$, an s-Club corresponds precisely to a clique). s-Clubs in a network have been established for network analysis and have recently been employed in examining social networks and biological networks.

The objective of the Min s-Club Cover problem is to cover a graph with the minimum number of s-Clubs such that every vertex belongs to at least one s-Club. This problem has been previously studied [11], with particular focus on the cases $s = 2$ and $s = 3$. It has been shown that determining whether a graph can be covered by two 3-Clubs or three 2-Clubs is NP-complete.

In [30], the authors proposed a multifactorial evolutionary algorithm for solving the Minimum s-Club Cover problem. They introduced an individual representation, as well as crossover and mutation operators. To improve solution quality, a greedy strategy was applied during both the initial population generation and the crossover process. Additionally, the mutation operator was implemented as a combination of three simple mutation strategies.

In [29], a hybrid approach combining multitasking optimization and a heuristic method was introduced. In this approach, the heuristic serves as a local search algorithm applied at each generation. The local search focuses on determining effective criteria for selecting the best club to which a vertex should be moved. Furthermore, the study described a mechanism for applying the heuristic to individuals in the Unified Search Space (USS), specifically targeting the best individual in each task.

In recent years, researchers have shown growing interest in Multitasking Optimization (MTO), which focuses on addressing multiple tasks simultaneously. Inspiration from traditional Evolutionary Algorithm (EA), Evolutionary Multitasking Optimization (EMO) utilizes an evolutionary search strategy to solve multiple problems in parallel. This paradigm facilitates knowledge transfer between tasks, improving solution quality and faster convergence.

One prominent example of EMO is the Multifactorial Evolutionary Algorithm (MFEA) introduced by Gupta et al. [16], which employs a population-based framework known as the USS to enable the sharing of important genetic material among individuals from different tasks. Thanks to these capabilities, MFEA has demonstrated outstanding performance in various real-world applications [17], such as complex combinatorial optimization problems [14, 28].

The MFEA has also demonstrated promising results when applied to graph problems with clustering characteristics. Specifically, MFEA has been used to address the Clustered Shortest-Path Tree Problem (CluSPT) problem through various approaches, such as decomposing the problem into two levels [19], and employing a Cayley-based encoding scheme for individual representation in the USS [9]. Another NP-hard problem involving graph partitioning is the Inter-Domain Path Computation under Edge-defined Domain Uniqueness Constraint (IDPC-EDU) problem[23]. In [2], Binh et al. applied the MFEA to solve the IDPC-EDU problem by introducing a two-layer encoding technique.

While multitask evolutionary algorithms (MTEAs) have been successfully applied to various graph-related problems, including the s-Club cover problem, integrating local search strategies within these frameworks has received relatively limited attention. A key challenge lies in accurately identifying the corresponding task for each individual in the USS, which is necessary for applying task-specific local search methods effectively. Nevertheless, local search plays a vital role in refining candidate solutions and accelerating convergence in evolutionary computation. This creates a strong incentive to investigate more effective ways of incorporating local search into MTEAs, especially for complex combinatorial problems such as the Minimum s-Club Cover problem. In this study, we propose an enhanced local search mechanism for the Minimum s-Club Cover problem, and investigate how it can be incorporated into a multitask evolutionary framework.

4 Proposed algorithm

This section describes the proposed algorithms based on the combination of multitask optimisation and local search algorithms, focusing on describing the mechanism of the local algorithm. This study uses individual representation and evolutionary operators in [30].

Table 2: Results obtained by EMT-G, GA, EMT-DSE and SALO on instances.

Instances	**EMT-G**				**GA**				**EMT-DSE**				**SALO**			
	BF	Avg	STD	CV	BF	Avg	STD	CV	BF	Avg	STD	CV	BF	Avg	STD	CV
adjnoun	27	29.0	0.89	0.03	31	***32.0***	0.22	0.01	28	***29.3***	0.73	0.03	19	19.0	0.00	0.00
celegansneural	4	4.1	0.00	0.00	9	***12.8***	0.91	0.07	4	4.1	0.31	0.08	4.5	***4.5***	0.00	0.00
celegans_metabolic	87	88.3	0.73	0.01	88	***88.6***	0.50	0.01	88	***88.5***	0.60	0.01	32	32.0	0.00	0.00
chesapeake	3	3.0	0.00	0.00	3	3.0	0.00	0.00	3	3.0	0.00	0.00	3	3.0	0.00	0.00
dolphins	15	16.7	0.00	0.00	16	***17.0***	0.51	0.03	15	***16.8***	0.64	0.04	17	***17.0***	0.00	0.00
football	11	13.0	0.00	0.00	13	***13.3***	0.47	0.04	12	***13.2***	0.55	0.04	15	***15.0***	0.00	0.00
jazz	16	16.0	0.00	0.00	16	16.0	0.00	0.00	16	16.0	0.00	0.00	14	14.0	0.00	0.00
karate	4	4.0	0.00	0.00	4	4.0	0.00	0.00	4	4.0	0.00	0.00	4	4.0	0.00	0.00
lesmis	3	3.0	0.00	0.00	3	***3.1***	0.31	0.10	3	3.0	0.00	0.00	3	3.0	0.00	0.00
polbooks	14	15.4	0.00	0.00	15	***16.4***	1.04	0.06	14	15.2	0.81	0.05	15	15.0	0.00	0.00
johnson8-2-4	1	1.0	0.00	0.00	1	1.0	0.00	0.00	1	1.0	0.00	0.00	1	***1.6***	0.49	0.30
hamming6-4	1	1.0	0.00	0.00	1	1.0	0.00	0.00	1	1.0	0.00	0.00	4	***4.0***	0.00	0.00
MANN_a9	1	1.0	0.00	0.00	1	1.0	0.00	0.00	1	1.0	0.00	0.00	1	1.0	0.00	0.00
c-fat200-1	13	13.0	0.00	0.00	13	13.0	0.00	0.00	13	13.0	0.00	0.00	13	13.0	0.00	0.00
hamming6-2	1	1.0	0.00	0.00	1	1.0	0.00	0.00	1	1.0	0.00	0.00	1	1.0	0.21	0.21
johnson8-4-4	1	1.0	0.00	0.00	1	1.0	0.00	0.00	1	1.0	0.00	0.00	2	***2.0***	0.00	0.00
c-fat200-2	6	6.0	0.00	0.00	6	6.0	0.00	0.00	6	6.0	0.00	0.00	6	***6.6***	0.50	0.08
c-fat200-5	3	3.0	0.00	0.00	3	3.0	0.00	0.00	3	3.0	0.00	0.00	3	3.0	0.00	0.00
keller4	1	1.0	0.00	0.00	1	1.0	0.00	0.00	1	1.0	0.00	0.00	2	***2.0***	0.00	0.00
gen200_p0.9_44	1	1.0	0.00	0.00	1	1.0	0.00	0.00	1	1.0	0.00	0.00	2	***2.0***	0.00	0.00

4.1 Algorithm scheme

Incorporating local search operators into multitask evolutionary algorithms presents unique challenges compared to single-task optimization. In traditional evolutionary algorithms, local search can be directly applied to individuals within the population. However, in multitask settings, individuals in the USS encode solutions for multiple tasks simultaneously, complicating the direct application of task-specific local search operators.

Our approach addresses this challenge through a three-stage process applied to elite individuals from each task. First, we select the best-performing individual for each task from the combined parent-offspring population. Second, we decode the selected individual in USS to obtain a task-specific solution representation. Third, we apply the proposed local search operator to this decoded solution and subsequently update the corresponding individual in USS. This strategy ensures that local search improvements are propagated back to the shared population while maintaining the multitask optimization framework's integrity. Algorithm 1 presents the detailed implementation of this integration mechanism.

To apply the local search algorithm, first, the algorithm decodes the individual in USS to obtain the solution of the current task. Then, it applies the local search to the solution. Finally, the individual in the USS is updated based on the obtained solution.

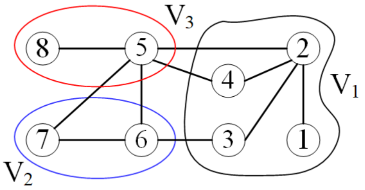
4.2 Encoding and decoding method

A chromosome consists of two sections: the first section, referred to as the club component, contains information about the clubs; the second section specifies the club assignment for each vertex. The direct vertex-to-cluster encoding maps each vertex to a specific cluster label. This encoding provides flexibility in handling a variable number of clusters and helps preserve meaningful structures throughout the evolutionary process, thereby enhancing convergence and search efficiency. As a result, this encoding scheme [8] is adopted for representing the second section.

Figure 2 shows an example of encoding a solution for a task, where Figure 2(a) shows a graph with three clubs $V_1 = \{1, 2, 3, 4\}$, $V_2 = \{6, 7\}$, and $V_3 = \{5, 8\}$; Since the graph has eight vertices, the individual has eight genes. In other words, the dimension of the individual is 8. Figure 2(b) illustrates an individual encoding the graph presented in Figure 2(a), with the clubs V_1, V_2, and V_3 labeled as 1, 2, and 3, respectively. Since vertices 1, 2, 3, and 4 belong to club 1, these vertices' labels are 1. Vertices 5 and 8 belong to club 2, so these two vertices have the label 3. Similarly, the label of vertices 6 and 7 belonging to club 2 is 2.

Algorithm 1: The main steps of the proposed algorithm

```
begin
    $N \leftarrow$ The population size;
    /* Initialize initial population */
    $P(0) \leftarrow$ Randomly generated individuals;
    Assign the skill factor for the individuals in $P(0)$;
    $t \leftarrow 0$;
    while stopping conditions are not satisfied do
        $P_c(t) \leftarrow \emptyset$        ▷ Offspring population;
        while $|P_c(t)| < N$ do
            $p_i\ (i = 1, 2) \leftarrow$ Select randomly two individuals from $P(t)$;
            /* Perform crossover and mutation operators */
            $o_i \leftarrow$ Perform crossover between the individuals $p_i\ (i = 1, 2)$ ;
            $o'_i \leftarrow$ Perform mutation on the individuals $o_i\ (i = 1, 2)$;
            Evaluate the individuals $o'_i\ (i = 1, 2)$;
            $P_c(t) \leftarrow P_c(t) \cup \{o'_i\}\ (i = 1, 2)$;
        $R(t) \leftarrow P_c(t) \cup P(t)$;
        Update scalar fitness of each individual in $R(t)$;
        foreach (task tsk) do
            $ind_{tsk} \leftarrow$ Select the best individual of the task $tsk$;
            $sol_{tsk} \leftarrow$ Decode the individual $ind_{tsk}$;
            $sol'_{tsk} \leftarrow$ Apply local search for $sol_{tsk}$;
            Update the solution $sol'_{tsk}$ to individual $ind_{tsk}$;
        $P(t + 1) \leftarrow$ Get $N$ fittest individuals from $R(t)$;
        $t \leftarrow t + 1$;
    return The best solution of Min s-Club Cover for each task.
```



(a)

clubs	1	2	3					
vertices	1	1	1	1	3	2	2	3

(b)

Figure 2: An example for the solution encoding method for a task

An individual in the USS encodes the solutions for all tasks; therefore, the algorithm must store sufficient information to reconstruct the solution of each task. This study adopts the following encoding strategy for individuals in the USS:

- The length of each section within an individual in the USS is set to the dimension of the largest task.
- For each task, if its dimension is m, then the first m genes from the corresponding section of the individual are used to construct its solution.

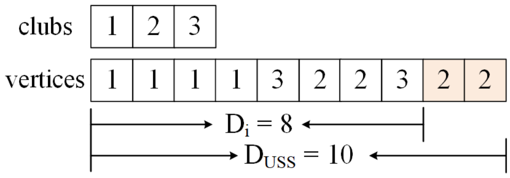


Figure 3: An individual in USS

Figure 3 illustrates the individual in the USS for two tasks, where both tasks have three clubs, and the number of vertices is 8 and 10, respectively. Because the number of vertices in the first task is 8, the first eight genes of the individual in the USS are used to construct the solution for the first task, i.e., 1–1–1–1–3–2–2–3. For the second task, which has 10 vertices, the first 10 genes from the corresponding section of the individual are used, resulting in the solution 1–1–1–1–3–2–2–3–2–2.

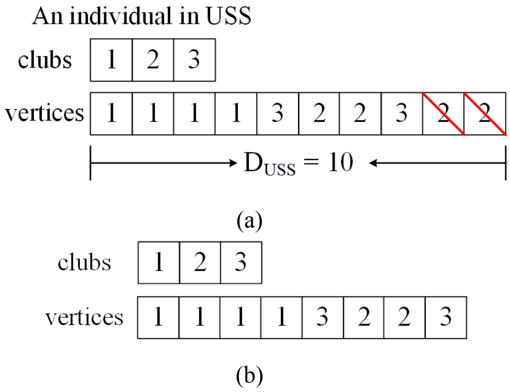


Figure 4: An example about decoding method

This encoding ensures that a single individual in the USS can be decoded to provide valid solutions for multiple tasks of varying dimensions.

The solution to a task is decoded from an individual in the USS by extracting the first genes in the vertex section. The club section is constructed by relabeling the gene values to ensure consistent and sequential club labels. Figure 4 illustrates the decoding process. In Figure 4(a), an individual in the USS is shown, where the last two genes are unselected during solution construction. Figure 4(b) presents the resulting solution for a task with eight vertices.

4.3 Evolutionary Operators

4.3.1 Crossover operator

The crossover operator utilised in this study is based on the method described in [30], and consists of the following main steps:

- Step 1: Randomly select two crossover points within the club sections of each parent.
- Step 2: Insert the elements from the selected clubs of the first parent into the corresponding positions in the offspring.
- Step 3: Add the elements from the selected clubs of the second parent to the offspring, ensuring that no duplicates are introduced from those already added by the first parent.
- Step 4: For the remaining unassigned elements, attempt to place them into existing clubs in ascending order of club size (i.e., the number of vertices in each club). If adding a vertex to a club does not violate the diameter constraint, assign it to that club; otherwise, proceed to the next one.
- Step 5: If there are still unassigned vertices that cannot be added to any existing club without violating constraints, create a new club and assign these vertices to it.
- Step 6: Renumber the club labels in the offspring sequentially, starting from 1 up to the total number of clubs.

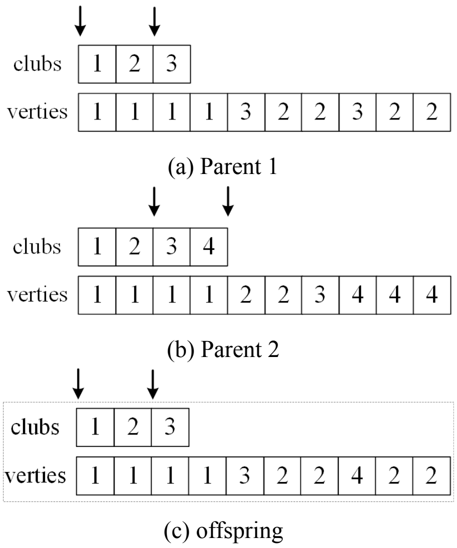


Figure 5: An illustration of crossover operator

Figure 5 depicts the crossover operator, where Figures 5(a) and 5(b) represent the two parent individuals, and Figure 5(c) illustrates the resulting offspring.

4.3.2 Mutation operator

The mutation operator comprises three types of mutation: move mutation, splitting mutation, and merging mutation. The main ideas of these mutations are as follows:

- Move mutation: randomly select a club containing at least two vertices, and then choose one vertex from that club to move to a different club.
- Splitting mutation: split a club into two clubs.
- Merging mutation: combine two clubs into a club.

4.4 Local search algorithm

This study employs a random greedy strategy to select a club, prioritising those with more vertices. It then sequentially transfers vertices from the selected club to other clubs, ensuring that, after the transfer, both the source club (from which vertices were moved) and the destination clubs (to which vertices were added) satisfy the s-club constraint. Since deleting vertices of degree 1 does not violate the s-club constraint, these vertices are given priority and are moved first.

The mean steps of the propose local search are presented in Algorithm!2.

Algorithm 2: Local seach algorithm

```
Input: - A connected graph $G = (V, E)$;
       - The number of clubs $s >= 2$;
       - The parameter of random greedy algorithm %priority and %restriction;
       - An individual $sol_p$;
Output: A solution of minimum s-Club cover;
begin
    $cl_i \leftarrow$ A club with the smallest number of vertices;
    if $(random < \%priority)$ then
        foreach (vetex $v$ in the $cl_i$ ) do
            if (The degree of vertex $v$ is either $1$ or $0$) then
                Remove vertex $v$ from club $cl_i$;
                foreach (club $cl_j$ in $sol_p(j \neq i)$) do
                    Add vertex $v$ to club $cl_j$;
                    if (club $cl_j$ is a s-Club) then
                        break;
                    else
                        Remove vertex $v$ from club $cl_j$;
                if (No suitable club is found for adding vertex $v$) then
                    Add vertex $v$ to club $cl_i$;
    else
        $N_i \leftarrow$ The number of vertices in the club $cl_i$;
        $max\_vertex \leftarrow N_i * (1 + \%restriction)$    ▷ Compute the maximum number of vertices for selecting a club.;
        $tList \leftarrow$ The list consists of clubs containing fewer than $max\_vertex$ vertices;
        $cl_r \leftarrow$ RandomSelect(tList)    ▷ Randomly selected a club from the list $tList$;
        foreach (vetex $v$ in the $cl_r$ ) do
            if (The degree of vertex $v$ is either $1$ or $0$) then
                Remove vertex $v$ from the club $cl_r$;
                foreach (club $cl_j$ in $sol_p(j \neq r)$) do
                    Add vertex $v$ to club $cl_j$;
                    if (club $cl_j$ is a s-Club) then
                        break;
                    else
                        Remove the vertex $v$ from the club $cl_r$;
                if (No suitable club is found for adding vertex $v$) then
                    Add vertex $v$ to club $cl_r$;
    return $sol_p$;
```

5 Computational results

5.1 Problem instances

To evaluate the performance of the proposed algorithm, Min s-Club Cover instances from the DIMACS benchmark suite [1, 20] are used. The selected instances contain fewer than 3,00 vertices, making them suitable for computational experimentation. Descriptive statistics for these instances are provided in Tables 3, where $|V|$ denotes the number of vertices, $|E|$ denotes the number of edges, and D_G denotes the graph density.

5.2 Experimental criteria

Criteria for assessing the quality of the output of the algorithms are presented in Table 4.

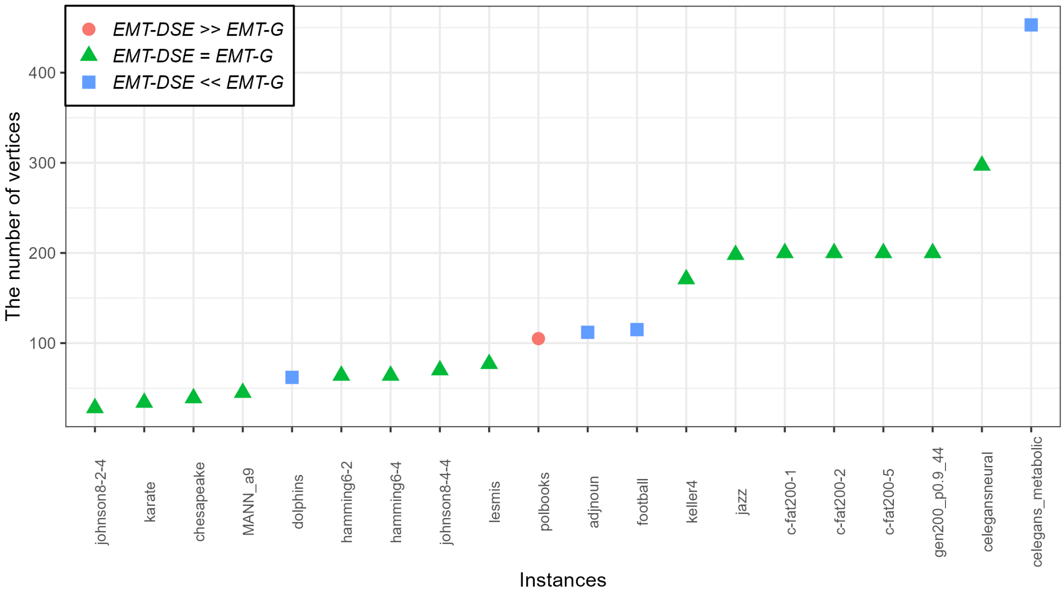
Table 4: Criterias for assessing the quality of the output of the algorithm

Average (Avg)	Average function value over all
Best-found (BF)	Best function value achieved over all runs
STD	Standard deviation
CV	Coefficient of Variation

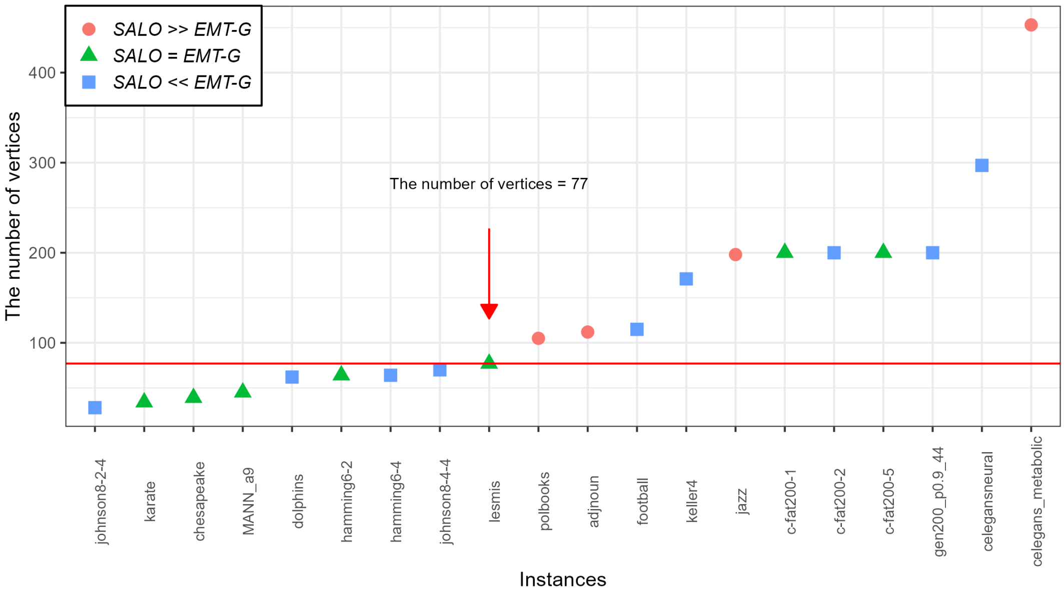
5.3 Experimental Settings

The proposed is compared with three algorithm:

- Genetic Algorithm (GA), representing a classical



(a) Comparison with EMT-DSE



(b) Comparison with SALO

Figure 6: Scatter plot illustrating the relationship between the number of vertices and the performance of the EMT-G algorithm in comparison with SALO and EMT-DSE

single-task optimization approach, was employed in [30] to address the problem.

- EMT-DSE [30, 11] is an evolutionary multitasking algorithm explicitly designed for the Min s-Club Cover. It leverages a dynamic solution encoding strategy to enable knowledge transfer across tasks.
- Simulated Annealing-based Local Optimization (SALO) [22] is a recently introduced heuristic method aimed at partitioning the vertex set of a graph into subsets. We adapt SALO by defining a neighborhood structure where a solution is modified by relocating a vertex from one club to another, thereby enhancing exploration in the solution space.

Since previous studies [10, 11, 30] only addresses the Min s-Club Cover problem with $s = 2$, the proposed algorithm is also implemented for this specific case. With respect to EMT-G and EMT-DSE, we adopt the same parameter settings as those used in the work of Cheng [16], specifically setting the random mating probability (rmp) to 0.3. These settings are widely used and validated in prior studies [3, 31]. The parameter configuration for the SALO algorithm follows the setup described by Zhi Lu et al. [22],

Table 3: Summary information of instances

Instances	$\lvert V\rvert$	$\lvert E\rvert$	D_G
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	0.020
polbooks	1490	16715	0.015
johnson8-2-4	28	210	0.56
hamming6-4	64	704	0.35
MANN_a9	45	918	0.93
c-fat200-1	200	1534	0.08
hamming6-2	64	1824	0.90
johnson8-4-4	70	1855	0.77
c-fat200-2	200	3235	0.16
c-fat200-5	200	8473	0.43
keller4	171	9435	0.65
gen200_p0.9_44	200	17910	0.90

$|V|$: The number of vertices; $|E|$: The number of edges; D_G: The density of a graph.

with $\theta_{size} = 8$, $\theta_{cool} = 0.96$, and $\theta_{minper} = 1\%$. For the proposed local search algorithm, the priority and restriction parameters are assigned values of 0.8 and 0.7, respectively. To ensure a fair comparison, all algorithms are independently executed 20 times on a machine with an Intel Core i7-12700K CPU and 32GB of RAM, running Microsoft Windows 10. The EMT-G, GA and EMT-DSE methods utilise a population of 100 individuals and perform 50,000 evaluations. The implementations were developed in the C# programming language.

5.4 Experimental results

5.4.1 A Comparative Analysis of Algorithms

The results obtained by the algorithms are presented in Table 2. In the table, bold and italic cells in a column indicate the instances where the EMT-G algorithm outperforms the corresponding algorithm in that column.

The table presents a comparative summary of EMT-G against GA, EMT-DSE, and SALO. The columns 'Worse', 'Better', and 'Equal' indicate the number of instances in which EMT-G performed worse than, better than, or equal to each respective algorithm.

The Table 5 presents a summary of comparisons of EMT-G against three other algorithms: GA, EMT-DSE, and SALO. The comparison metrics are the number of instances where the EMT-G algorithm performed *Worse*, *Better*, or *Equal* to the respective compared algorithms.

Table 5: Summary of the Comparison of Results Obtained by EMT-G, GA, EMT-DSE, and SALO

Algorithm	**EMT-G**		
	Better	Equal	Worse
GA	7	13	0
EMT-DSE	4	15	1
SALO	9	8	3

- **Comparison with GA:**
  - The EMT-G algorithm performed better than GA in 7 instances and equal to GA in 13 instances. There were no instances where EMT-G performed worse than GA.
  - This indicates that EMT-G consistently outperforms GA, with a strong lead in the number of better-performing cases and no cases of inferior performance.
- **Comparison with EMT-DSE:**
  - EMT-G performs worse than EMT-DSE in 1 instances, better in 4 instances, and equally in 15 instance.
  - Similar to the comparison with GA, EMT-G shows a strong advantage over EMT-DSE.
- **Comparison with SALO:**
  - EMT-G performs worse than SALO in 3 instances, better in 9 instances, and equally in 8 instances..
  - The results suggest a more balanced performance between EMT-G and SALO. Although EMT-G demonstrates a number of better outcomes, it also has more cases of worse performance compared to the other algorithms. The relatively high number of equal cases implies that SALO is a more competitive counterpart to EMT-G.

In summary, EMT-G generally demonstrates superior performance compared to GA and EMT-DSE, consistently achieving more favorable outcomes. However, when compared to SALO, its performance is more mixed, indicating that SALO presents a stronger challenge and, in some cases, may even outperform EMT-G.

5.4.2 Analysis of influential factors

In this subsection, we analyse the influence of the input graph's dimensions (number of vertices) and its density on the performance of EMT-G.

To examine the correlation between the number of vertices and graph density, scatter plots were generated showing the relationship between the number of vertices, graph density, and the performance comparison of EMT-G against EMT-DSE and SALO for the given instances. The correlation coefficient for this relationship was then calculated, as shown in Figure 6 and Figure 7. In these figures, circles indicate that EMT-G performs worse than the compared algorithms, squares indicate that EMT-G outperforms them, and triangles indicate equal performance between the algorithms.

As shown in Figure 6, when the number of vertices in an instance is less than 16.715, SALO does not outperform EMT-G. The figure also indicates that EMT-G outperforms SALO when the number of edges is relatively small.

Figure 7 shows that EMT-G is no worse than the compared algorithms for instances with a graph density greater than or equal to 0.16. When the graph density is greater than 0.09, EMT-G consistently outperforms EMT-DSE. This means that EMT-G tends to be more efficient than EMT-DSE as the graph density increases. When the graph density is greater than0.16, EMT-G consistently outperforms SALO.

6 Conclusion

The minimum s-club cover problem has attracted considerable attention from researchers in the analysis of social networks and group interactions. In this study, we propose a local search algorithm that utilizes a randomized greedy strategy to select clubs for evaluation, aiming to minimize the number of vertices required. The local search method is integrated into a multifactorial evolutionary algorithm framework, enhancing the quality of the best individual in each task at every generation. Experimental evaluations conducted on datasets from the DIMACS library demonstrate that the proposed algorithm outperforms existing approaches.

In future work, we aim to further improve the efficiency of the local search component by reducing the computational cost of verifying valid clubs.

Acknowledgements

This research was funded by Hanoi University of Science and Technology under project code T2024-PC-038.

References

[1] Graph Partitioning and Graph Clustering. American Mathematical Society (Jan 2013), `http://dx.doi.org/10.1090/conm/588`

[2] Binh, H.T.T., Thang, T.B., Long, N.B., Hoang, N.V., Thanh, P.D.: Multifactorial evolutionary algorithm for inter-domain path computation under domain uniqueness constraint. In: 2020 IEEE Congress on Evolutionary Computation (CEC). p. 1–8. IEEE (Jul 2020), `http://dx.doi.org/10.1109/cec48606.2020.9185701`

[3] Binh, H.T.T., Thang, T.B., Thai, N.D., Thanh, P.D.: A bi-level encoding scheme for the clustered shortest-path tree problem in multifactorial optimization. Engineering Applications of Artificial Intelligence **100**, 104187 (Apr 2021), `http://dx.doi.org/10.1016/j.engappai.2021.104187`

[4] Bourjolly, J.M., Laporte, G., Pesant, G.: An exact algorithm for the maximum k-club problem in an undirected graph. European Journal of Operational Research **138**(1), 21–28 (2002), `http://dx.doi.org/10.1016/s0377-2217(01)00133-3`

[5] Cavique, L., Mendes, A.B., Santos, J.M.A.: An Algorithm to Discover the k-Clique Cover in Networks, p. 363–373. Springer Berlin Heidelberg (2009), `http://dx.doi.org/10.1007/978-3-642-04686-5_30`

[6] Cerioli, M.R., Faria, L., Ferreira, T.O., Martinhon, C.A., Protti, F., Reed, B.: Partition into cliques for cubic graphs: Planar case, complexity and approximation. Discrete Applied Mathematics **156**(12), 2270–2278 (2008), `http://dx.doi.org/10.1016/j.dam.2007.10.015`

[7] Chakraborty, D., Chandran, L.S., Padinhatteeri, S., Pillai, R.R.: Algorithms and Complexity of s-Club Cluster Vertex Deletion, p. 152–164. Springer International Publishing (2021), `http://dx.doi.org/10.1007/978-3-030-79987-8_11`

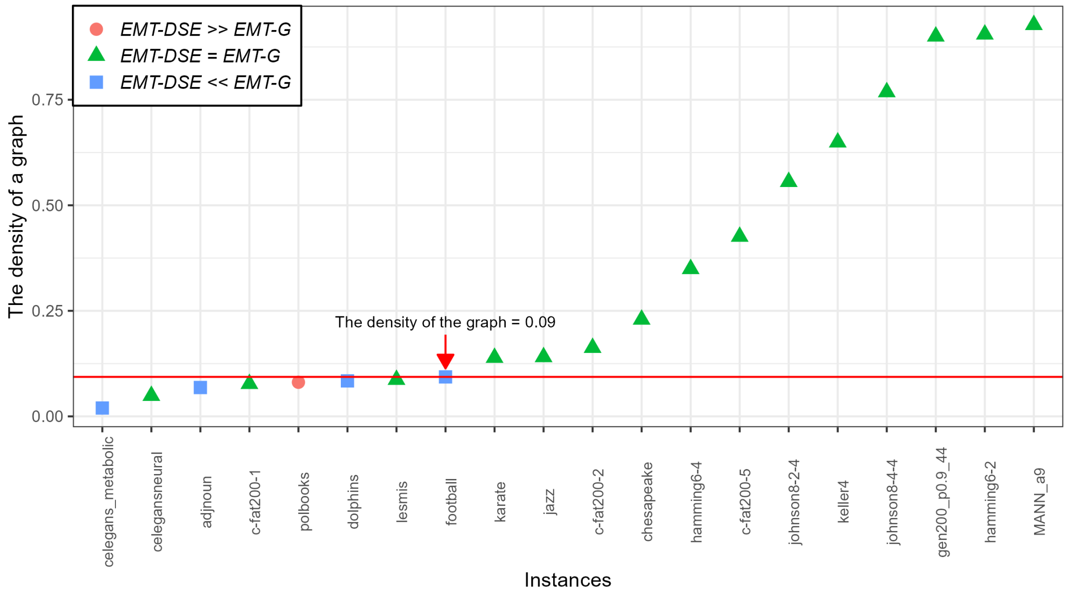
[8] Chaouche, A., Boulif, M.: Solving the unsupervised graph partitioning problem with genetic algorithms: Classical and new encoding representations. Computers & Industrial Engineering **137**, 106025 (Nov 2019), `https://linkinghub.elsevier.com/retrieve/pii/S036083521930484X`

[9] Dinh, T.P., Thanh, B.H.T., Ba, T.T., Binh, L.N.: Multifactorial evolutionary algorithm for solving clustered tree problems: competition among cayley codes: Case studies on the clustered shortest-path tree problem and the minimum inter-cluster routing cost clustered tree problem. Memetic Computing **12**(3), 185–217 (Aug 2020), `http://dx.doi.org/10.1007/s12293-020-00309-2`

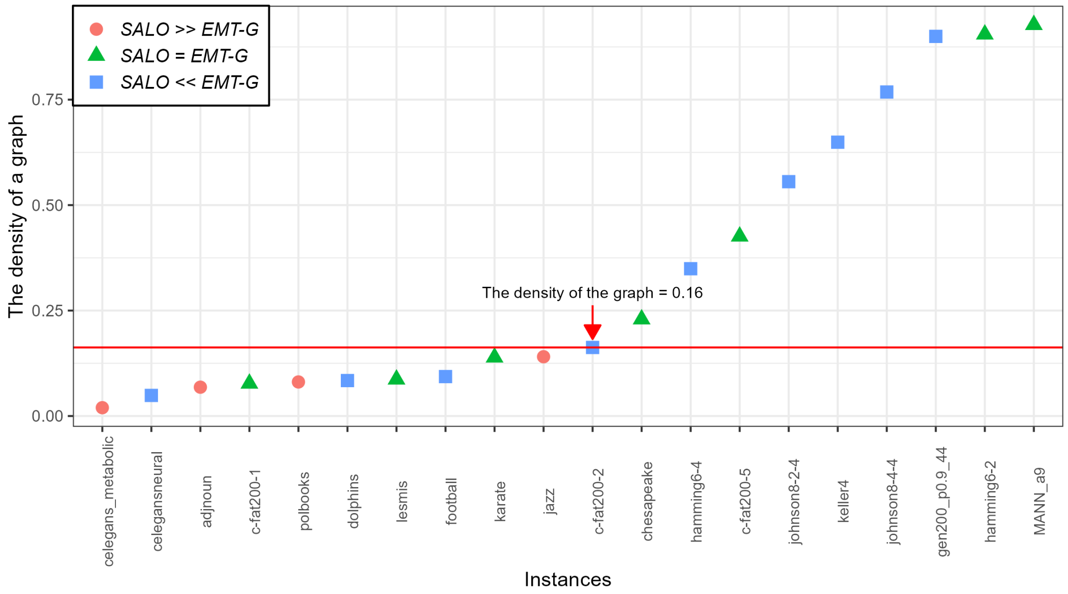
[10] Dondi, R., Lafond, M.: On the tractability of covering a graph with 2-clubs. Algorithmica **85**(4), 992–1028 (Nov 2022), `http://dx.doi.org/10.1007/s00453-022-01062-3`

[11] Dondi, R., Mauri, G., Sikora, F., Zoppis, I.: Covering a graph with clubs. Journal of Graph Algorithms and Applications **23**(2), 271–292 (Jan 2019), `http://dx.doi.org/10.7155/jgaa.00491`

[12] Dondi, R., Mauri, G., Zoppis, I.: On the tractability of finding disjoint clubs in a network. Theoretical



(a) Comparison with EMT-DSE



(b) Comparison with SALO

Figure 7: Scatter plot illustrating the relationship between graph density and the performance of the EMT-G algorithm in comparison with SALO and EMT-DSE

Computer Science **777**, 243–251 (Jul 2019), `http://dx.doi.org/10.1016/j.tcs.2019.03.045`

[13] Dumitrescu, A., Pach, J.: Minimum clique partition in unit disk graphs. Graphs and Combinatorics **27**(3), 399–411 (Mar 2011), `http://dx.doi.org/10.1007/s00373-011-1026-1`

[14] Feng, L., Huang, Y., Zhou, L., Zhong, J., Gupta, A., Tang, K., Tan, K.C.: Explicit evolutionary multitasking for combinatorial optimization: A case study on capacitated vehicle routing problem. IEEE Transactions on Cybernetics **51**(6), 3143–3156 (Jun 2021), `http://dx.doi.org/10.1109/tcyb.2019.2962865`

[15] Fortunato, S.: Community detection in graphs. Physics Reports **486**(3–5), 75–174 (Feb 2010), `http://dx.doi.org/10.1016/j.physrep.2009.11.002`

[16] Gupta, A., Ong, Y.S., Feng, L.: Multifactorial evolution: Toward evolutionary multitasking. IEEE Transactions on Evolutionary Computation **20**(3), 343–357 (Jun 2016), `http://dx.doi.org/10.1109/tevc.2015.2458037`

[17] Gupta, A., Zhou, L., Ong, Y.S., Chen, Z., Hou, Y.: Half a dozen real-world applications of evolutionary multitasking, and more. IEEE Computational Intelligence Magazine **17**(2), 49–66 (May 2022), `http://dx.doi.org/10.1109/mci.2022.3155332`

[18] Gupta, P., Arora, M., Thakur, H.K.: Community detection in social networks: A deep learning approach using autoencoders. Informatica **49**(5) (Jan 2025), `http://dx.doi.org/10.31449/inf.v49i5.7018`

[19] Huynh Thi Thanh, B., Pham Dinh, T.: Two levels approach based on multifactorial optimization to solve the clustered shortest path tree problem. Evolutionary Intelligence **15**(1), 185–213 (Oct 2020), `http://dx.doi.org/10.1007/s12065-020-00501-w`

[20] Johnson, D., Trick, M.: Introduction to the Second DIMACS Challenge: Cliques, coloring, and satisfiability, p. 1–7. American Mathematical Society (Oct 1996), `http://dx.doi.org/10.1090/dimacs/026/01`

[21] Laan, s., Marx, M., Mokken, R.J.: Close communities in social networks: Boroughs and 2-clubs. SSRN Electronic Journal (2015), `http://dx.doi.org/10.2139/ssrn.2686127`

[22] Lu, Z., Zhou, Y., Hao, J.K.: A hybrid evolutionary algorithm for the clique partitioning problem. IEEE Transactions on Cybernetics **52**(9), 9391–9403 (Sep 2022), `http://dx.doi.org/10.1109/tcyb.2021.3051243`

[23] Maggi, L., Leguay, J., Cohen, J., Medagliani, P.: Domain clustering for inter□domain path computation speed□up. Networks **71**(3), 252–270 (Dec 2017), `http://dx.doi.org/10.1002/net.21800`

[24] Mokken, R.J., Heemskerk, E.M., Laan, S.: Close communication and 2-clubs in corporate networks: Europe 2010. Social Network Analysis and Mining **6**(1) (Jun 2016), `http://dx.doi.org/10.1007/s13278-016-0345-x`

[25] Mokken, R.J., et al.: Cliques, clubs and clans. Quality and Quantity **13**(2), 161–173 (1979), `http://dx.doi.org/10.1007/bf00139635`

[26] Pasupuleti, S.: Detection of Protein Complexes in Protein Interaction Networks Using n-Clubs, p. 153–164. Springer Berlin Heidelberg (2008), `http://dx.doi.org/10.1007/978-3-540-78757-0_14`

[27] Szabó, S., Zaválnij, B.: Benchmark problems for exhaustive exact maximum clique search algorithms. Informatica **43**(2) (Jun 2019), `http://dx.doi.org/10.31449/inf.v43i2.2678`

[28] Thang, T.B., Long, N.B., Hoang, N.V., Binh, H.T.T.: Adaptive knowledge transfer in multifactorial evolutionary algorithm for the clustered minimum routing cost problem. Applied Soft Computing **105**, 107253 (Jul 2021), `http://dx.doi.org/10.1016/j.asoc.2021.107253`

[29] Thanh, P.D., Anh, D.T.: A Hybrid Multifactorial Evolutionary Algorithm for the Minimum s-Club Cover Problem, p. 232–242. Springer Nature Singapore (2025), `http://dx.doi.org/10.1007/978-981-96-4288-5_19`

[30] Thanh, P.D., Long, N.B., Vinh, L.S., Binh, H.T.T.: Evolutionary multitasking algorithm based on a dynamic solution encoding strategy for the minimum s-club cover problem. Evolutionary Intelligence **18**(1) (Nov 2024), `http://dx.doi.org/10.1007/s12065-024-00999-4`

[31] Wen, Y.W., Ting, C.K.: Parting ways and reallocating resources in evolutionary multitasking. In: 2017 IEEE Congress on Evolutionary Computation (CEC). p. 2404–2411. IEEE (Jun 2017), `http://dx.doi.org/10.1109/cec.2017.7969596`