Finding a maximum clique in social networks using a modified differential evolution algorithm

JUTAMARD WONGPEN, JEERAYUT WETWEERAPONG, PIKUL PUPHASUK
Department of Mathematics, Faculty of Science
Khon Kaen University, Khon Kaen, 40002
THAILAND
jutamard.w@kkumail.com wjeera@kku.ac.th ppikul@kku.ac.th

Abstract: The relationships of the individuals in social networks give rise to interesting and important features in various fields, such as graph mining and communication networks. Among those useful features are clique structures which represent fully connected relations between members, and the maximum cliques which identify the highest-connected subgroups. In this work, we propose the modified differential evolution algorithm (moDE) for finding a maximum clique in social networks. The moDE solves the constrained continuous optimization problem which is transformed from the discrete maximum clique problem. It uses a new mutation strategy to generate and adjust mutant vectors, mixes two important crossover rates in crossover, and incorporates the extracting and extending clique procedure to increase the performance of clique finding. The algorithm is tested on several social network problems and compared with the previously developed method. The results show that moDE is effective for finding a maximum clique and outperforms the compared method.

Key-Words: Maximum clique problem, social network, differential evolution algorithm, constrained continuous optimization problem

1 Introduction

Let $G=(V,E)$ be an undirected graph where V and E are the vertex set and edge set, respectively. The maximum clique problem (MCP) is to find a largest complete subgraph of G. This subgraph is called a maximum clique and its cardinality is called a clique number denoted by $\omega(G)$. The MCP arises in many applications, such as telecommunications [1], bioinformatics [2], and social network analysis [3]. Finding the maximum clique in an arbitrary graph is difficult since it is an NP-hard problem [4]. Thus, the use of heuristic or metaheuristic methods is necessary. Various local methods, for instance, branch and bound algorithm [5], tabu search [6], sequential greedy algorithm [7], and variable neighborhood search [8] have been proposed for solving MCP. There are also population-based methods, such as genetic algorithm (GA) [9] and ant colony optimization (ACO) [10] designed for MCP.

Due to the discrete nature of the problem, all those population-based methods for MCP are discrete optimization methods. In contrast to this, the differential evolution algorithm (DE), an efficient population-based methods for continuous optimization problems [11-12], is aimed in this research to solve the MCP which is equivalently transformed to the continuous problem using the following theoretical result of Motzkin and Straus [13]:

$$\text{maximize} \quad f(x)=\sum_{\{i,j\}\in E,\, i<j} x_i x_j \tag{1}$$

$$\text{subject to} \sum_{i=1}^{n} x_i = 1$$

$$\text{where} \quad 0 \le x_i \le 1,\ i=1,2,\ldots,\mathrm{n} \ \text{ and } \ n=|V|.$$

The optimal solution x^* of (1) forms a maximum clique S of G of size k where $k=\dfrac{1}{1-2f(x^*)}$ and x^* can be attained by setting $x^*=1/k$ if the vertex $i\in S$ and $x^*=0$ otherwise.

Recently, MCP has been applied to social network analysis for identifying the highly-connected subgroups of members. Soleimani-pouri et al. presented the ACO-PSO algorithm, a hybrid of ant colony optimization and particle swarm optimization, for finding a maximum clique in social networks [3]. The performances of ACO-PSO are compared to those of standard ACO by testing on social network problems. The results show that ACO-PSO outperforms ACO. In this work, we propose the modified differential evolution algorithm called moDE for solving MCP. The moDE solves the constrained continuous MCP (1) by integrating several improvements to the basic DE. The algorithm is tested on several social network problems and compared with the ACO-PSO.

The remainder of the paper is organized as follows. The next Section 2 summarizes the basic DE algorithm and describes in detail the modified DE (moDE) algorithm for solving the transformed continuous MCP. Section 3 explains the experimental design and lists all the test problems. In Section 4, the results of applying the moDE are presented and compared with those of the ACO-PSO method [3]. Then, the conclusion is given in the last section.

2 Algorithm description

2.1 Basic differential evolution algorithm

DE algorithm is a simple and robust population-based optimization method proposed by Storn and Price in 1997 [12]. For more than two decades, it has been shown to be one of the most efficient methods for continuous optimization problems [14-16]. DE algorithm consists of three basic population operations: mutation, crossover and selection. The pseudocode of a basic DE is illustrated in Table 1. The main distinguished features of DE are the self-referential mutation to generate mutant vectors using the scaled difference of two vectors to adjust the other one, and the combined binomial crossover to each target vector to obtain a trial vector for the selection. First, the initial population of real vectors are generated uniformly in the feasible region. For each generation and each target vector x_i, a mutant vector $v_i = x_{r_1} + F(x_{r_2} - x_{r_3})$ is constructed from three different random population vectors x_{r_1}, x_{r_2} and x_{r_3} which are also different from x_i where F is a scaling factor. Then the components of v_i are exchanged with those of x_i according to the crossover rate C to construct a trial vector u_i and the vector u_i will replace x_i in selection if u_i is fitter. DE population vectors will evolve iteratively and move toward an optimal solution. It is well recognized that DE's performances depend on the control parameters np, F and C [17-18].

2.2 Modified differential evolution algorithm

Since the transformed continuous MCP (1) has one equality constraint, the moDE needs to modify and improve a basic unconstrained DE in order to handle the constrained problem. All population vectors are normalized in initialization, mutation and crossover processes. The new mutation strategy, the special bound adjustment, and the mix of two important crossover rates are introduced. The extracting & extending clique procedure is also

Table 1. Pseudocode of the basic DE algorithm.

The DE algorithm:

1. **Input**: The objective function f to be maximized, dimension n (of the domain of f), and population size np .
2. **Initialization**: Randomly generate the population of np vectors of dimension n in the domain of f . Evaluate all population vectors and find the best vector $xbest$ and its best value $fbest$.
3. **Mutation**: Generate a mutant vector v_i for each target (population) vector x_i by
   $v_i = x_{r_1} + F(x_{r_2} - x_{r_3})$
   where x_{r_1}, x_{r_2} and x_{r_3} are different random population vectors which are also different from x_i , and F is a scaling factor.
4. **Crossover**: Construct a trial vector u_i by replacing some of components of x_i with the corresponding components of v_i using the crossover rate C and one randomly fixed index to guarantee a change of at least one component as follows.
   $$u_{ij} = \begin{cases} v_{ij}\,;\ q_j < C \ \ or\ \ j = IC \\ x_{ij}\,;\ otherwise \end{cases}$$
   where $j = 1, 2, ..., n$; IC is a random integer in the set $\{1, 2, ..., n\}$ and q_j is a random number generated for each j in range of (0,1).
5. **Selection**: Replace the target vector x_i with the trial vector u_i if $f(u_i)$ is greater than $f(x_i)$. Also update the $xbest$ vector if $f(u_i)$ is greater than $fbest$.
6. Repeat all the steps 3-5 until reaching the stopping condition. Then report the obtained best solution.

incorporated to increase the performance of clique finding. The flowchart of moDE is shown in Fig. 1 and its algorithm is described as follows.

1. **Initialization** : Initialize the population of real vectors $X = [x_i]$ by randomly generating $x_i = [x_{i1}, x_{i2}, ..., x_{in}]$ with $0 \le x_{ij} \le 1$ where $i = 1, 2, ..., np; j = 1, 2, ..., n$; np is the population size and n is the number of vertices. Then, normalize each x_i to satisfy the constraint $\sum_{j=1}^{n} x_{ij} = 1$. Calculate the fitness for each population vector and record the best vector $xbest$ and the best value $fbest$.

2. **Mutation:** Random p_m between 0 and 1 to select one of two mutation equations to generate the mutant vector v_i for each i according to p_m as follows:

$$v_i = \begin{cases} x_{r_1} + F_1(x_{r_2} - x_{r_3}) + F_2(x_{r_4} - x_{r_5});\ p_m < 0.7 \\ x_{r_1} + F_1(x_{best} - x_{r_2}) + F_2(x_{best} - x_{r_3});\ otherwise \end{cases}$$

where r_1, r_2, r_3, r_4 and r_5 are randomly chosen in the set $\{1, 2, ..., np\}$ such that $r_1 \ne i$, x_{best} is the current best solution; and F_1 and F_2 are the scaling factors. If $v_{ij} < 0$ or $v_{ij} > 1$, then adjust each component v_{ij} of v_i by

$$v_{ij} = p_j \cdot \max_{1 \le k \le n}\{x_{r_1}(k)\}$$

where $j = 1, 2, ..., n$ and p_j is a random number generated for each j in range of (0,1). Then normalize v_i by $v_{ij} := \frac{v_{ij}}{\sum_{j=1}^{n} v_{ij}}$.

3. **Crossover**: Random p_c between 0 and 1 to select a crossover rate C_i for each i by

$$C_i = \begin{cases} 0.1;\ p_c < 0.5 \\ 0.9;\ otherwise \end{cases}.$$

Then construct the trial vector u_i by exchanging the components of x_i and v_i as follows:

$$u_{ij} = \begin{cases} v_{ij};\ q_j < C_i \ or \ j = IC \\ x_{ij};\ otherwise \end{cases}$$

where $j = 1, 2, ..., n$; IC is a random integer in the set $\{1, 2, ..., n\}$ and q_j is a random number generated for each j in range of (0,1). Then normalize u_i by

$$u_{ij} := \frac{u_{ij}}{\sum_{j=1}^{n} u_{ij}}.$$

4. **Selection** : Apply the greedy selection by comparing the function values of u_i and x_i. Retain or update x_i for the next generation as follows:

$$x_i = \begin{cases} u_i;\ f(u_i) > f(x_i) \\ x_i;\ otherwise \end{cases}.$$

Update the best current solution x_{best} and its best value f_{best}.

5. **Extracting & extending clique procedure**: Apply the procedure every nm generations. Let g be the current generation. If $\text{modulo}(g, nm) = 0$ then extract a set S of components of x_{best}, as large as possible, such that the induced graph by S is a clique. The extracting procedure varies the values $h = 0.1, 0.2, ..., 1$ to find the smallest h such that $S = \{i \mid x_i \ge h \cdot \max_{1 \le k \le n}\{x_{best}(k)\}\}$ induces a clique. Then the extending procedure extends S (when possible) using a random permutation Q of all vertices with each of the following three strategies:
(E1) Extend S by iteratively adding each vertex of $Q - S$ to S based on the order of Q to form the larger clique as possible.
(E2) Randomly select a proper subset T of S and extend T in the same manner as in E1.
(E3) Randomly select the vertex xs outside of S and extend $\{xs\}$ as in E1.
Update the current largest clique.

6. Repeat steps 2-5 until the maximum number of generations (ng) is reached. Then report the largest clique obtained.

3 Experimental design

To evaluate the performance of the moDE algorithm, two experiments are performed on 20 social network problems taken from the Network Repository [19]. The problems are also divided into 2 groups for the first and the second experiments, respectively. The first group contains 10 problems (A1-A10) which are used in [3] to test the performance of ACO-PSO. Thus, they are used in the first experiment to compare the performances of moDE with those of ACO-PSO. The second group of another 10 problems (B1-B10) are chosen to further verify the performances of the proposed algorithm in the second experiment.

The experiments are carried out on an Intel® core i5 processor 2.0 GHz and 4 GB RAM. The moDE algorithm is coded in Scilab version 6.0.2, an open source software available at http://www.scilab.org/.

3.1 Social network benchmark problems

The test problems for the two experiments are listed in Table 2 and Table 3, respectively. For each social network problem, the data set name, numbers of vertices and edges, and the lower bound of the size of the maximum clique (as report in [19]) are described. The example graphs of 6 problems of the first groups (A1-A6) are illustrated in Fig. 2 by using Social Network Visualizer Software [20].

3.2 Experiment 1: Performance comparison of the proposed moDE and ACO-PSO

The first experiment compares the performances of the moDE with those of ACO-PSO using the problems A1-A10. The following parameters are set as in [3]: the population size $np = 30$, the maximum number of generations $ng = 1000$, and 10 independent runs for each problem. In addition, the other parameters of moDE are set as follows: the scaling factors $F_1 = 0.5$ and $F_2 = 0.5$, and the period of generations to apply the extracting & extending clique procedure $nm = 10$. For each problem, the values "Best", "Mean" and "SD" of the sizes of the largest cliques found by each method are reported.

3.3 Experiment 2: Additional performance test of the moDE

The second experiment further verifies the performances of the moDE using the problems B1-B10. The settings of moDE are the same as in the first experiment. the values "Best", "Mean" and "SD" of the sizes of the largest cliques found by moDE are reported.

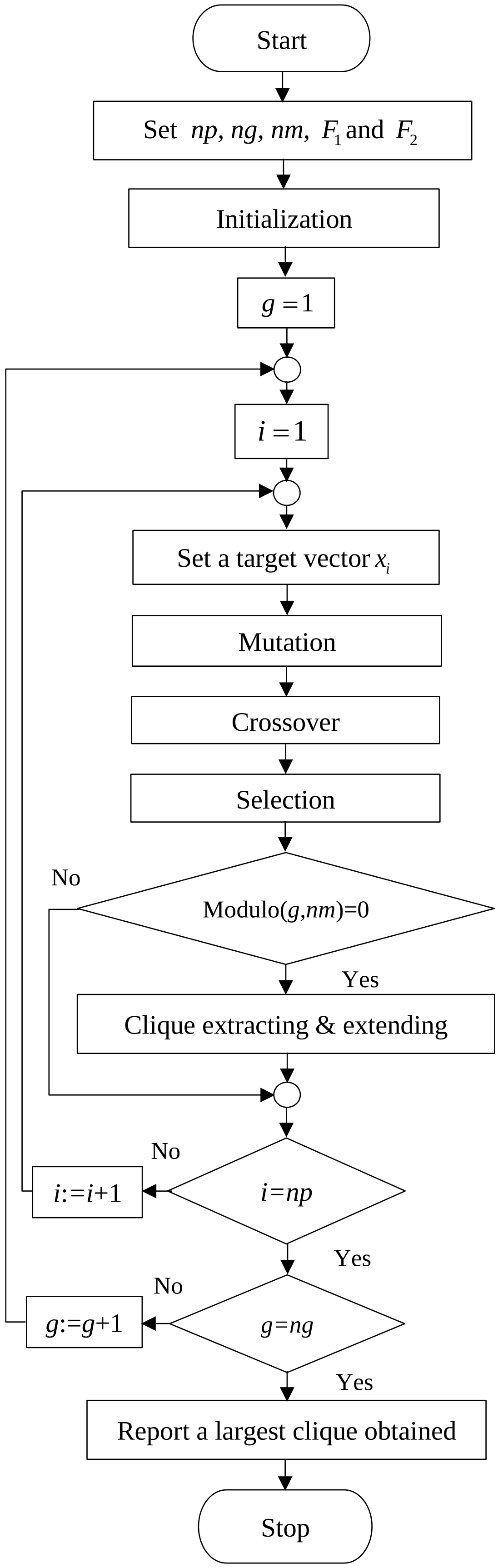

Fig. 1. Flowchart of the proposed moDE method.

Table 2. The first group of test problems [3,19].

Problem	Data set	$\lvert V \rvert$	$\lvert E \rvert$	$\omega(G)$
A1	Zachary's karate club	34	78	≥ 5
A2	Common adjective and nouns in "David Copperfield"	112	425	≥ 5
A3	Social network of dolphins, Doubtful Sound, New Zealand	62	159	≥ 5
A4	Pajek network: Erdos collaboration network 971	472	1314	≥ 7
A5	Pajek network: Erdos collaboration network 991	492	1417	≥ 7
A6	Pajek network: World Soccer, Paris 1998	35	295	≥ 6
A7	Pajek network: graph and digraph glossary	72	118	≥ 4
A8	Pajek network: Slovenian journals 1999-2000	124	823168	≥ 65
A9	Pajek network: SmaGri Citation network	1059	4919	≥ 8
A10	Email interchange network, Univ. Rovira i Virgili, Tarragona	1133	5451	≥ 12

Table 3. The second group of test problems [19].

Problem	Data set	$\lvert V \rvert$	$\lvert E \rvert$	$\omega(G)$
B1	Les Miserables Co-appearance network	77	254	≥ 10
B2	American football games between Div IA colleges, Fall 2000	115	613	≥ 9
B3	Pajek network: US Air	332	2126	≥ 22
B4	Pajek network: Erdos collaboration network 981	485	1381	≥ 7
B5	Wildbird network	202	4574	≥ 45
B6	Collaboration network between Jazz musicians	198	2742	≥ 9
B7	Co-authorship of scientists in network theory & experiments	379	914	≥ 9
B8	Co-authorship of scientists in network theory & experiments	1589	2742	≥ 9
B9	Facebook networks	769	16656	≥ 12
B10	Wikipedia who-votes-on-whom network	889	2914	≥ 6

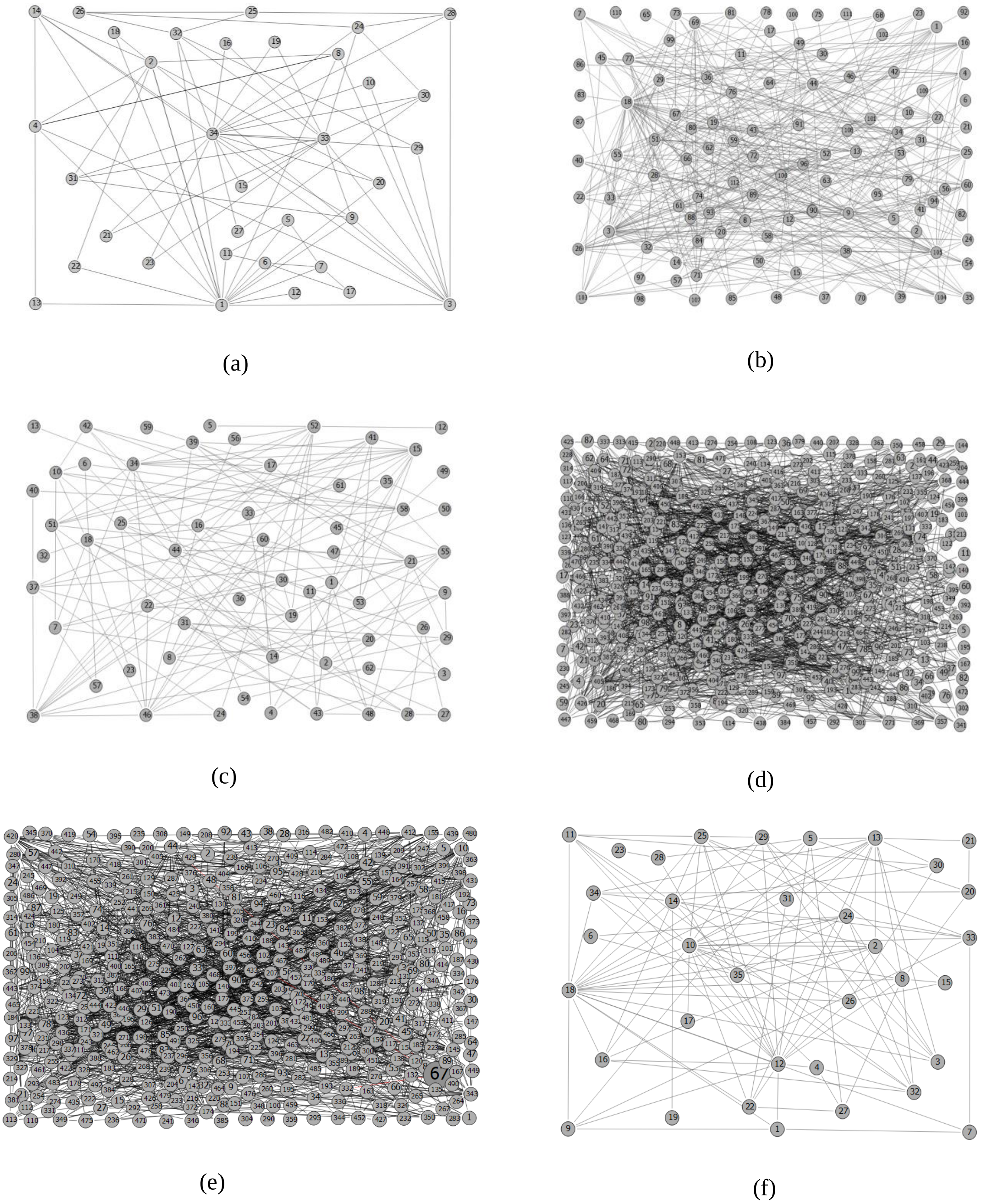

Fig. 2. Graphs of some test problems: (a) Problem A1, (b) Problem A2, (c) Problem A3, (d) Problem A4, (e) Problem A5, (f) Problem A6.

4 Results and discussion

The experimental results of experments 1 and 2 are shown in Table 4 and Table 5, respectively. In each table, the best values of Best, Mean, SD are indicated in bold. Moreover, we present the largest cliques obtained by moDE of the first and second groups of test problems in Table 6 and Table 7 where the vertex numbers are the same as in the reference.

4.1 Performance comparison of moDE and ACO-PSO

The performance comparison of moDE and ACO-PSO on problems A1-A10 is presented in Table 4. The results show that moDE can find the cliques with the maximum sizes as reported in the reference for all test problems whereas ACO-PSO can find 8 out of 10 cases.

The moDE gives the better Mean values and also gives the stable results with SD=0.0 for all cases. Moreover, it finds a larger clique than that reported in the reference for problem A8. This clearly shows that moDE outperforms ACO-PSO.

4.2 Performance of moDE on additional test problems

For the additional problems B1-B10, the performances of moDE are presented in Table 5. It shows the same trend of the performances of moDE. The proposed algorithm can find the cliques with the maximum sizes larger or equal to those reported in the reference for all test problems. The larger cliques are found for problems B5, B6 and B8. This indicates the effectiveness of moDE.

Table 4. Performance comparison of moDE and ACO-PSO on the first group of test problems.

Problem	$\omega(G)$	ACO-PSO[3]			moDE		
		Best	Mean	SD	Best	Mean	SD
A1	≥ 5	5	4.995	0.070	5	**5**	0.0
A2	≥ 5	5	4.783	0.412	5	**5**	0.0
A3	≥ 5	5	4.998	0.044	5	**5**	0.0
A4	≥ 7	7	5.848	0.667	7	**7**	0.0
A5	≥ 7	7	6.011	0.819	7	**7**	0.0
A6	≥ 6	5	4.118	0.322	**6**	**6**	0.0
A7	≥ 4	4	3.985	0.121	4	**4**	0.0
A8	≥ 65	4	3.185	0.299	**66**	**66**	0.0
A9	≥ 8	8	6.409	0.765	8	**8**	0.0
A10	≥ 12	12	7.997	2.241	12	**12**	0.0

Table 5. Performance of moDE on the second group of test problems.

Problem	$\omega(G)$	moDE		
		Best	Mean	SD
B1	≥ 10	10	10	0.0
B2	≥ 9	9	9	0.0
B3	≥ 22	22	22	0.0
B4	≥ 7	7	7	0.0
B5	≥ 45	**46**	**46**	0.0
B6	≥ 9	**30**	**30**	0.0
B7	≥ 9	9	9	0.0
B8	≥ 9	**20**	**20**	0.0
B9	≥ 12	12	12	0.0
B10	≥ 6	6	6	0.0

Table 6. The largest cliques obtained by moDE algorithm for the first group of test problems.

Problem	The largest cliques represented by sets of vertices
A1	{1, 2, 3, 4, 8}, {1, 2, 3, 4, 14}
A2	{3, 18, 22, 51, 52}, {3, 18, 35, 52, 55}
A3	{7, 10, 14, 18, 58}, {19, 22, 30, 46, 52}, {19, 25, 30, 46, 52}
A4	{51, 117, 152, 165, 214, 356, 370}
A5	{58, 126, 161, 174, 223, 370, 386}, {71, 126, 161, 174, 223, 269, 386}, {71, 161, 174, 223, 269, 386, 479}
A6	{10, 12, 18, 24, 25, 35}
A7	{18, 28, 30, 41}, {18, 30, 41, 71}, {22, 39, 52, 71}, {26, 30, 41, 71}
A8	{1, 2, 3, 4, 5, 6, 8,9,10, 11, 12, 14, 16, 17, 21,22, 24, 26, 27, 29, 30, 32, 34, 35, 36, 37, 38, 39, 40, 42, 56, 57, 62, 66, 68, 71, 74, 77, 78, 79, 80, 85, 86, 87, 88, 90, 92, 99, 102, 103, 104, 106, 107, 109, 111, 112, 113, 114, 115, 116, 118,119, 120, 121, 122, 123}, {1, 2, 3, 4, 6, 8, 9, 10, 11, 12, 14, 16, 18, 21, 22, 24, 26, 27, 29, 30, 32, 34, 35, 36, 37, 38, 39, 40, 42, 56, 57, 62, 66, 68, 71, 72, 73, 74, 77, 78, 79, 80, 81, 85, 86, 87, 88, 92, 97, 102, 103, 104, 106, 107, 109, 111, 112, 113, 114, 116, 118, 119, 120, 121, 122, 123}, {1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 14, 16, 17, 21, 22, 24, 26, 27, 29, 30, 32, 34, 35, 36, 37, 38, 39, 40, 42, 56, 57, 62, 66, 68, 71,72, 74, 77, 78, 79, 80, 85, 86, 87, 88, 92, 99, 102, 103, 104, 106, 107, 109, 111, 112, 113, 114, 115, 116, 118, 119, 120, 121,122,123}
A9	{1, 10, 11, 176, 177, 260, 379, 470}
A10	{299, 389, 434, 552, 571, 726, 756, 788, 885, 886, 887, 888}

Table 7. The largest cliques obtained by moDE algorithm for the second group of test problems.

Problem	The largest cliques represented by sets of vertices
B1	{49, 56, 58, 59, 60, 62, 63, 64, 65, 66}, {49, 59, 60, 61, 62, 63, 64, 65, 66, 67}
B2	{2, 26, 34, 38, 46, 90, 104, 106, 110}, {47, 50, 54, 68, 74, 84, 89, 111, 115}
B3	{67, 109, 112,118, 131, 147, 150, 152, 162, 166, 167, 174, 176, 182, 201, 219,230, 248, 255, 258, 261, 299}, {67,112,118,131,147, 152, 162, 166, 167,172,174,176, 182, 201, 217, 219, 230, 248, 255, 258, 261, 299}
B4	{56, 124, 159, 172, 221, 366, 381}, {69,124,159, 172, 221, 366, 381}
B5	{2, 6, 7, 9, 10, 11, 12, 15, 22, 23, 25, 26, 27, 35, 41, 55, 56, 58, 59, 60, 61, 64, 73, 74, 77, 79, 80, 82, 86, 87, 89, 90, 91, 92, 97, 132, 133, 136, 142, 153, 155, 157, 164, 166, 172, 173}, {2, 6, 9, 10, 11, 12, 15, 22, 23, 25, 26, 27, 35, 41, 55, 56, 58, 59, 60, 61, 64, 73, 74, 77, 79, 80, 82, 86, 87, 89, 90, 91, 92, 97, 99, 132, 133, 136, 142,153, 155, 157, 164,166,172,173 }, {6, 9, 10, 11, 12, 15, 22, 23, 25, 26, 27, 35, 41, 55, 56, 58, 59, 60, 61, 73, 74, 76, 77, 79, 80, 82, 84, 86, 87, 89, 90, 91, 92, 97, 99, 132, 133,136, 142, 153, 155, 157,164, 166,172, 173}
B6	{4, 7, 12, 13, 14, 15, 18, 19, 20, 21, 23, 101, 121, 128, 133, 137, 149, 150, 152, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174 }
B7	{4, 5,15,16,45,46, 47,176,177}
B8	{646, 1430, 1431, 1432,1433,1434, 1435, 1436,1437, 1438, 1439, 1440, 1441, 1442, 1443, 1444, 1445, 1446, 1447, 1448}
B9	{90,132,154,249,276,279,377, 433, 455, 582, 615,669}
B10	{132, 204, 273, 399, 416, 536}, {204, 273, 399, 416, 431, 448}, {399, 416, 431, 466, 504, 536}, {416, 448, 482, 504, 523, 584}, {536,538, 562, 568, 575, 619}

5 Conclusion

In this research, a modified DE algorithm called moDE is presented for finding a maximum clique in social networks. It solves the constrained continuous optimization problem which is transformed from the discrete maximum clique problem by integrating several improvements to the basic DE: a new mutation strategy to generate and adjust mutant vectors, the mix of two important crossover rates in crossover, and the extracting and extending clique procedure. The experimental results show that the moDE algorithm is efficient for finding maximum cliques for all test problems and outperforms the compared method. Moreover, it can find the larger cliques than those reported in the reference for 4 test problems. This indicates that the proposed algorithm is effective for finding a maximum clique in social networks.

This research work also presents a promising approach of modifying and applying the efficient DE algorithm for continuous optimization problems to solve graph and network problems which are discrete (combinatorial) optimization problems. It is done through the equivalent continuous versions of the discrete problems. Future research can investigate the possibility of solving other graph problems such as graph coloring problems and the maximum independent set problems using the same approach.

Acknowledgements: The authors would like to thank the research capability enhancement program through graduate student scholarship, Faculty of Science, Khon Kaen University, for the financial support.

References:

[1] B. Balasundaram and S. Butenko, *Graph domination, coloring and cliques in telecommunications*, Handbook of Optimization in Telecommunications, pp. 865-890, Springer, 2006.

[2] E. Tomita, T. Akutsu and T. Matsunaga, *Efficient algorithms for finding maximum and maximal cliques: Effective tools for bioinformatics*, in A. N. Laskovski (Eds.), Biomedical Engineering Trends in Electronics, Communications and Software, pp. 625-640, InTech, 2011.

[3] M. Soleimani-Pouri, A. Rezvanian and M. R. Meybodi, *An ant based particle swarm optimization algorithm for maximum clique problem in social network*, in F. Can, T. Ozyer and F. Polat (eds.), State of the art applications of social network analysis, Lecture notes in social networks, pp. 295-304, Springer International Publishing, 2014.

[4] R. M. Karp, *Reducibility among combinatorial problems*, in R. E. Miller and J. W. Thatcher (Eds.), Complexity of Computer Computations, pp. 85-103, Plenum Press, 1972.

[5] P. M. Pardalos and G. P. Rodgers, A branch and bound algorithm for the maximum clique problem, *Computers Ops Res*, 19(5), 1992, pp. 363-375.

[6] M. Gendreau, P. Soriano and L. Salvail, Solving the maximum clique problem using a tabu search approach, *Annals of Operations Research*, 41(4), 1993, pp. 385-403.

[7] I. M. Bomze, M. Budinich, P. M. Pardalos and M. Pelillo, *The Maximum Clique Problem*, in D. Z. Du and P. M. Pardalos (Eds.), Handbook of Combinatorial Optimization, pp. 1-74. Kluwer Academic Publishers, 1999.

[8] P. Hansen, N. Mladenovic and D. Urosevic, Variable neighborhood search for the maximum clique, *Discrete Applied Mathematics*, 145, 2004, pp. 117-125.

[9] E. Marchiori, Genetic, iterated and multistart local search for the maximum clique problem, *Proceedings of the Applications of Evolutionary Computing on EvoWorkshops 2002: EvoCOP, EvoIASP, EvoSTIM/EvoPLAN* ,2002, pp. 112-121.

[10] C. Solnon and S. Fenet, A study of ACO capabilities for solving the maximum clique problem, *J Heuristics*, 12, 2006, pp. 155-180.

[11] S. Das, S. S. Mullick and P. N. Suganthan, Recent advances in differential evolution - An updated survey, *Swarm and Evolutionary Computation*, 27, 2016, pp. 1-30.

[12] R. Storn and K. Price, Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces, *J Glob Optim*, 11(4), 1997, pp. 341-359.

[13] T. S. Motzkin and E. G. Straus, Maxima for Graphs and a new proof of a theorem of Turán, *Canadian Journal of Mathematics*, 17, 1965, pp. 533-540.

[14] R. Storn, *Differential evolution research-trends and open questions*, in U. K. Chakraborty (ed.), Advances in Differential Evolution, pp. 1-31, Springer, 2008.

[15] F. Neri and V. Tirronen, Recent advances in differential evolution: a survey and

experimental analysis, *Artif Intell Rev*, 33, 2010, pp. 61-106.

[16] S. Das and P. N. Suganthan, Differential evolution: a survey of the state-of-the-art, *IEEE Trans Evol Comput*, 15(1), 2011, pp. 4-31.

[17] J. Lampinen and I. Zelinka, On stagnation of the differential evolution algorithm, *Proceeding of the 6th international Mendel conference on soft computing*, 2002, pp. 76-83.

[18] R. Gamperle, S. D. Muller and P. Koumoutsakos, A parameter study for differential evolution, *Proceeding of the conference in neural networks and applications (NNA), fuzzy sets and fuzzy systems (FSFS) and evolutionary computation (EC), WSEAS,* 2002, pp. 293-298.

[19] Network repository: A scientific network data repository with interactive visualization and mining tools. http://networkrepository.com/ index.php.

[20] SocNetV: Social Network Analysis and Visualization Software. https://socnetv.org.