Available online at http://scik.org

J. Math. Comput. Sci. 2022, 12:58

https://doi.org/10.28919/jmcs/6917

ISSN: 1927-5307

NOVEL MULTILEVEL PARTICLE SWARM OPTIMIZATION ALGORITHM FOR GRAPH PARTITIONING

NARESH GHORPADE[1,*], H. R. BHAPKAR[2]

[1]Department of Mathematics, MIT School of Engineering, MIT ADT University, Loni Kalbhor

Pune - 412201, India

[2]Department of Mathematics, MIT School of Engineering, MIT ADT University, Loni Kalbhor, Pune - 412201, India

Abstract: Graph partitioning is crucial step in resolving real time applications in the field of image analysis, smart city designing, wireless communications, data analysis etc. Though considerable research has been done for getting an optimal partitioning of graphs still it demands enhancement for diverse application problems. Hybrid graph partitioning approaches are promising and possess ability to partition graphs with large number of vertices. In our research we have developed multilevel particle swarm optimization algorithm for graph partitioning. Size of the graph is reduced by heavy edge matching algorithm and then greedy graph growing partitioning is used to divide the graph. Discrete particle swarm optimization used at the most important stage of refinement. Performance is evaluated by using Walshaw's Benchmark graphs and from analysis it has been observed that proposed algorithm generates optimal partitioning with reduced cut values and computational cost.

Keywords: multilevel particle swarm optimization algorithm; graph partitioning problems.

2010 AMS Subject Classification: 65K10.

[*]Corresponding author

E-mail address: nareshghorpade25051976@gmail.com

Received October 20, 2021

1. INTRODUCTION

Graph partitioning problems are extensively utilised in variety of application ranges such as optimal node deployments in IoT networks, Image Processing, Data Analysis, Corona Virus Graph Analysis etc. [1-6]. Optimal graph partitioning is NP-hard and hence for real time applications hybrid partitioning approaches are preferred. Multilevel graph partitioning is one of the most popular method to attain fast and optimum regions. In multilevel partitioning, original graph of larger size is reduced to the smaller graph by applying various schemes. This reduced graph is optimally partitioned and then those partitions are mapped revert with the input graphs. Multiple iterations are performed to get an optimal solution by using refinement method after each iteration. METIS [7], JOSTLE [8], Scotch [9], and DiBaP [10] are the examples of benchmark graphs in which multilevel partitioning scheme is implemented. From the methods reported in the literature it is observed that an appropriate combination of swarm intelligence-based optimization technique can improve the partitioning quality for all types of graphs. Multiple partitioning heuristics have been developed for the graphs involving influence regulation degrees dispersal [11]. Combinations of these algorithms can be useful for extensive range of partitioning applications. Machine learning methods can be explored for the choice of an appropriate partitioning techniques to solve particular application problem [12-13]. Particle Swarm Optimization (PSO) [14], a field of swarm intelligence-based computations, possess the capability of automatically producing [15] and enhancing heuristic techniques for a diverse set of applications [16]. Employing PSO to enhance multilevel partitioning technique can offer two different advantages. Primarily, the evolutionary progression will ponder heuristic which may be ignored all through the manual progress since they are not inbuilt. Subsequently, the outline designed for developing routine heuristic for the particular application can be swiftly utilised to solve numerous partitioning problems.

We have investigated the potentiality of discrete particle swarm optimization algorithm in automating the procedure of modifying multilevel partitioning algorithm, enhancing its performance in comparison with general state-of-the-art partitioning techniques.

Structure of the remaining paper is: Graph partitioning is described in depth in Section 2. Section 3 reviews of existing graph partitioning algorithms. Discrete Particle Swarm Optimization Technique is explained in Section 4. Multilevel Particle Swarm Optimization algorithm for graph partitioning is developed in Section 5. Performance evaluation of the developed algorithm using some benchmark graphs is presented and compared with existing techniques in Section 6. Lastly, Section 7 concludes the paper.

2. Graph Partitioning

If *G = (P, Q)* is weighted and graph in which P *and* Q represents set of nodes and edges respectively and every edge is assigned with positive weight, then the graph partitioning problem is to divide node set into m – blocks $(m \in N, M > 1)$, $P_1, P_2, \ldots, P_m$ such that $P_1 \cup P_2 \cup \ldots \cup P_m = P$ and $P_r \cap P_s = \emptyset, \forall\, r \neq s$. Balanced partition is the one in which each block has similar weight. Objective function for the graph partitioning problem is to minimize the cut value of partition between two sets P_r *and* P_s *in which* $P_r,\ P_s \subseteq P$

$$Cut\ (P_r, P_s) = \sum_{l \in P_r} \sum_{m \in P_s} w(l, m) \qquad (1)$$

An example of graph partitioning for undirected weighted graph is illustrated in Fig. 1.

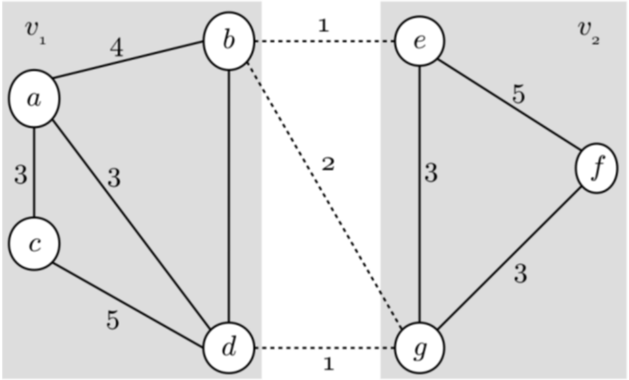


Figure 1. Graph Partitioning Example

2.1. Multi-level Graph Partitioning

Multilevel graph partitioning is most commonly utilised partition estimate method. Coarsening, partitioning and refinement are the stages involved in it. Multilevel graph partitioning is illustrated in Figure 2.

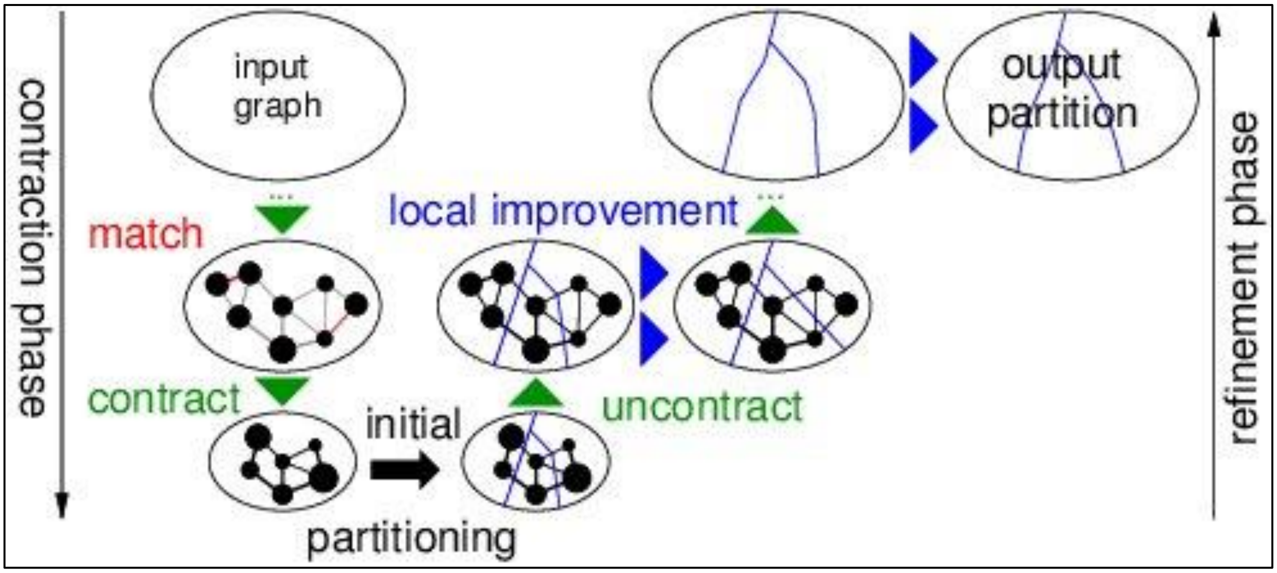


Fig. 2: Multilevel Graph Partitioning

2.1.1. Coarsening. In the coarsening stage, a reduced form of the input graph is generated. The process of coarsening generates a series of reduced and coarser graphs, till the dimensions of the coarsest graph are appropriately smaller. Maximal matching approach is used in it to select edges for contraction; which includes either random matching or heavy edge matching. In random edge matching vertices are visited in arbitrary order and edges between them are also arbitrarily chosen for contraction. Whereas in heavy edge matching an edge with maximal weight is chosen for contraction. Edge matching scheme during the coarsening is well suited for few application scenarios [6]. It has also been observed that coarsening of graphs only with edge matching is complex for graphs whose degrees are distributed by power law. For such type of graphs, enhanced coarsening can be attained by contraction of smaller and extremely connected subgraphs rather than contracting edges [11].

2.1.2. Partition. In this phase coarsened graphs are partitioned; smaller size of the graph is an added advantage for getting an optimal partitioning in very less time. Spectral partitioning [17], KL -partitioning [18] are the methods used for partitioning of coarsened graphs but they are computationally expensive. Karypis et al. [19] have proposed graph growing partition (GGP) and greedy graph growing partition (GGGP) which generates quality partitions. GGP starts by randomly visiting a vertex and then breadth first search approach is used to add it to the partition and the process is continued still the partition comprises the required vertex weight. GGGP is same as that of GGP but adjacent nodes

2.1.3. Uncoarsening and Refinement. In this phase coarsened partitioned graphs are mapped back to the coarsest graph and then the refinement is used for the quality improvement. This uncoarsening and refinement procedure is repetitive till an optimal partitioning is obtained. In KL-refinement coarsened partitioned graph is used as primary input, excluding individual run of an algorithm dismisses if a scalable number of node switches do not decline the cut value. On the contrary, in greedy refinement KL-refinement procedure is restricted to only one run.

3. Related Work

For the graphs with innovative stages of complexity, a novel range of partitioning methods on the basis of multilevel model have been proposed. Fiduccia et. al [20] proposed block gain approach and developed speedy heuristic method to bisect graph with edge weights. But weight among two nodes is not considered while calculating the gain. Multiple improvements in multilevel partitioning have been proposed but most of them fail to attain optimal partitioning of hypergraphs. These methods are widely clubbed with metaheuristics; viz. simulated annealing, tabu search, genetic algorithm (GA), ant colony optimization (ACO) and particle swarm optimization (PSO). Simulated annealing proposed for bipartition of graphs in [21] is used by C. Bichot at al. [22] to divide the graph into *k* – parts. Simulated annealing is flexible towards adaptability of distinct objective functions and partition conditions. However, its convergence speed is very low.

Genetic algorithm combined with multilevel method for *k*- partitioning of the graph in [23] uses jostle evolutionary and promising results are generated with minimal cut value. Ant Colony Optimization for partitioning of graphs uses the fundamental idea that the distinct colonies of ants in a region fights among them and every colony is considered as part of the graph. Multi-ant colony optimization approach proposed in [24] is based on the concept of grid shielded by region of ants in which every grid is associated by a vertex. Multilevel graph partitioning is united with ACO by S. Bach*et al.* [25], this approach produces optimal partitioning at the cost of higher time. Green *et al.* [26] proposed a graph partitioning technique for large size graphs by the combination of breadth first search (BFS) with particle swarm optimization (PSO). It uses the principle of communication among the partitions and inside the partition with the objective of minimizing communication among the partitions and maximizing communication inside the partition. It improves convergence rate of an algorithm. This approach is limited to canonical graphs.

Dynamic airspace configuration model proposed in [27] is defined as weighted graph partitioning problem with an objective of minimizing workload by appropriate sectorization. PSO is incorporated with multilevel method to partition the weighted graph of region to be sectorized. It has been observed that proposed approach has produced the sectors satisfying the balance of workload. Ghorpade et al. [28] proposed energy centered approach y characterized as graph cut. An objective function attained in this method bits optimal value after segmenting an image.

4. Discrete Particle Swarm Optimization (DPSO)

Kennedy *et al.* [29] have developed discrete particle swarm optimization (DPSO) technique for solving discrete optimization problems. In DPSO primarily, the number of particles and corresponding speed vectors are created arbitrarily. After fixed iterations, the objective of an algorithm is to get an optimal or close to optimal results with the help of predefined fitness function. Position of velocity vector is updated at each iteration by using best positions: personal best (p_{best}) and global best (g_{best}) after that velocity vector is used to determine particle positions. For a swarm of size K, location of kth particle is denoted using vector $\overrightarrow{A_k} = (a_{k1}, a_{k2}, \dots, a_{kl})^t;\ k \in \{1, 2, \dots, K\}$ and, a_{kn} corresponds to lth dimension of position vector $\vec{A}_k$ that possess the values zero and one. Velocity of these particles is represented by the vector, $\vec{B}_k = (b_{k1}, b_{k2}, \dots, b_{kl})^t$; where every element b_{kn} represents the chances that an element a_{kn} will possess the value 'one'. The best previously obtained location of kth particle is $\overrightarrow{C_k} = (c_{k1}, c_{k2}, \dots, c_{kl})^t$ and the best previously obtained location of the whole is $\overrightarrow{C_i} = (c_{m1}, c_{m2}, \dots, c_{ml})^t$ where m is the best particle in the swarm.

Velocity and positions after 't 'iterations are updated by using Eq. (2) and Eq. (3),

$$\overrightarrow{B_k^{t+1}} = c.\vec{B}_k^t + \vec{U}[0, \eta_1] * \left(\vec{C}_k^t - \vec{A}_k^t\right) + \vec{U}[0, \eta_2] * \left(\vec{C}_k^t - \vec{A}_k^t\right) \qquad (2)$$

$$a_k^t = \begin{cases} 1, & if\ Sig\ b_k^{t+1} > r_k \\ 0, & if\ Sig\ b_k^{t+1} \leq r_k \end{cases} \qquad (3)$$

Sigmoid function is defined in Eq. (3),

$$Sig(\ b_k^{t+1}) = \frac{1}{1 + e^{-(\ b_k^t)}} \qquad (4)$$

Function $\overrightarrow{U}$ is uniformly distributed and yields vectors whose locations are arbitrarily chosen, $*$ indicates point wise multiplication of vectors, η_1 and η_2 are perceptive and communal constraints. $r_k \in [0, 1]$. DPSO algorithm ends after the maximum number of iterations are attained.

5. Multilevel Discrete Particle Swarm Optimization for Graph Partitioning (MLPSO)

For minimal cut graph partitioning discrete particle swam optimization is not feasible for the applications in which graphs posse's larger vertex sets. Consequently, we have proposed to incorporate multilevel partitioning with DPSO and developed MLPSO algorithm for graph partitioning. MLPSO aims to find the optimal solution for the minimum cut partitioning problem of graph *G = (P, Q)*. Every particle of the swarm is selected as partition vector $\bar{A}$, and the fitness function is to minimize *Cut* $(\bar{A})$.

MLPSO Operates in three phases; primary partitioning in which population is initialized on smaller graphs, then the refinement projects back the particles to the succeeding level finer graph and lastly bisected graph is partitioned into *k* – parts by recursion.

In the stage of coarsening, a reduced form of the input graph is generated. The procedure of coarsening generates a series of reduced and coarser graph, till the dimensions of the coarsest graph are appropriately smaller. Maximal matching approach is used in it to select edges for contraction; which includes either random matching or heavy edge matching. In random edge matching vertices are visited in arbitrary order and edges between them are also arbitrarily chosen for contraction. Whereas in heavy edge matching an edge with maximal weight is chosen for contraction.

To partition the coarsened graphs; we have used GGGP method to create $G_r = (P_r, E_r)$. In this method, vertices of graph are separated in three groups X, Y, Z. Group X is set by arbitrarily choosing any vertex (say 'a') from set P to initialize set Y and Z. Select a nearest vertex from group Y to the group X with maximal gain and add it to X. Then every vertex in group Z that is incident on vertex 'a' and shifted to group Y by determining its gain. Similarly, recalculate the gain of each vertex in part Y that are incident on vertex 'a' and hence the subsequent iteration starts. This procedure is constant still the weight of part X grow into half of the overall weight. Process stops when $w(X) = \frac{1}{2}w(P)$.

The Position vector, velocity vector, and personal best vector on the graph $G_r = (V_r, E_r)$ for each k^{th} particle are $\overrightarrow{A_{rk}},\ \overrightarrow{B_{rk}}$, and $\vec{C}_{rk}$ respectively. MLPSO sets the swarm particles on G_r during the phase of partitioning and projects back sequentially entire swarm, $\overrightarrow{A_{rk}},\ \overrightarrow{B_{rk}}\ ,\ \vec{C}_{rk}$ to the succeeding level finer graph.

After this stage, internal and external weight of every particle is calculated. Internal weight of the k^{th} particle of the vertex u is represented by IW_u^k and it is the summation of weight of the arcs between vertex u and other vertices inside the block and external weight is represented by EW_u^k and it is the summation of weights of the between vertex u and other vertices outside the block.

$$IW_u^k = \sum_{\substack{(u,v)\in Q \\ a_{ku=a_{kv}}}} w(u,v) \tag{5}$$

$$EW_u^k = \sum_{\substack{(u,v)\in Q \\ a_{ku=a_{kv}}}} w(u,v) \tag{6}$$

Borderline vertices of every particle with nonnegative external weights are deposited in a border hash table. The framework of MLPSO entails a nested loop. Ending conditions are outlined by the outside loop, whether to run MLPSO for maximal number of cycles M_{max} or not. Internal and external weight of the particles are crucial in calculating gain and borderline vertex for the simple execution of MLPSO. At every iteration, the weights of all the adjacent vertices of the shifted vertex are restructured to maintain the uniformity in internal and external weights. The border hash table fluctuates corresponding to the variations in partitioning. Lastly, the recursive algorithm is applied to get 'k' parts of the bisected graph produced in earlier phase of MLPSO.

6. PERFORMANCE ANALYSIS

For the performance evaluation of MLPSO, Walshaw's GPT Benchmark [30] is used and the results are compared with multilevel mesh partitioning and domain decomposition technique (MMPDT) [31] and multilevel iterative tabu search (MLTS) [32]. Parameter setting for MLTS is $\alpha = 0.99, \gamma = 0.099|P|, p_{str=}0.0199|P|$ and for MLPSO is $c = 0.99, \eta_1 = \eta_2 = 0.499, B_{max} = 5, K = 40, M_{max} = 25.$ The termination time changes according to number of vertices and edges in a graph, which ranges from half second for the smaller graphs up to 2500 node to four minutes for the huge graphs. Comparison of cut values for partition of each graph for different values of k is shown in Fig. 3, Fig. 4, and Fig. 5.

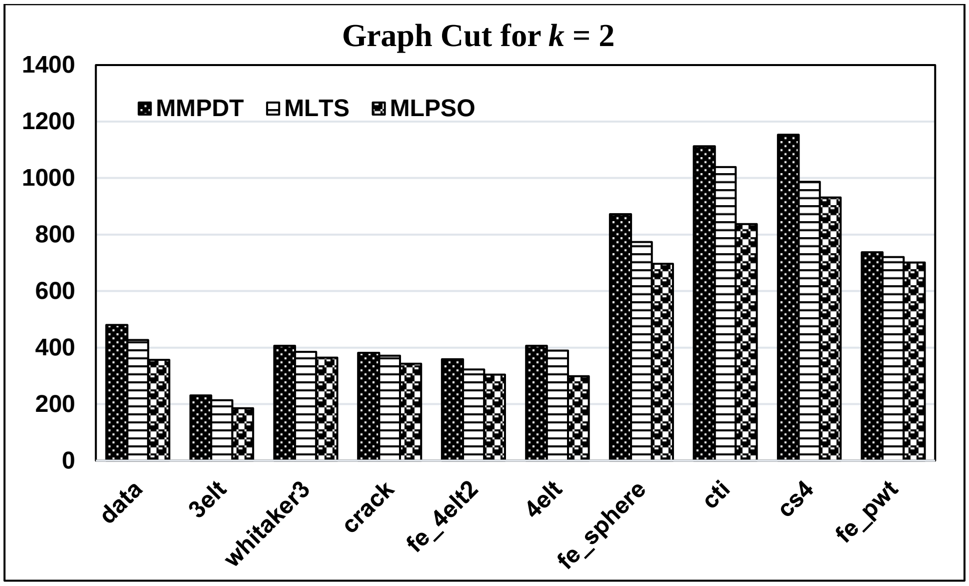


Fig. 3: Graph cut for *k* = 2

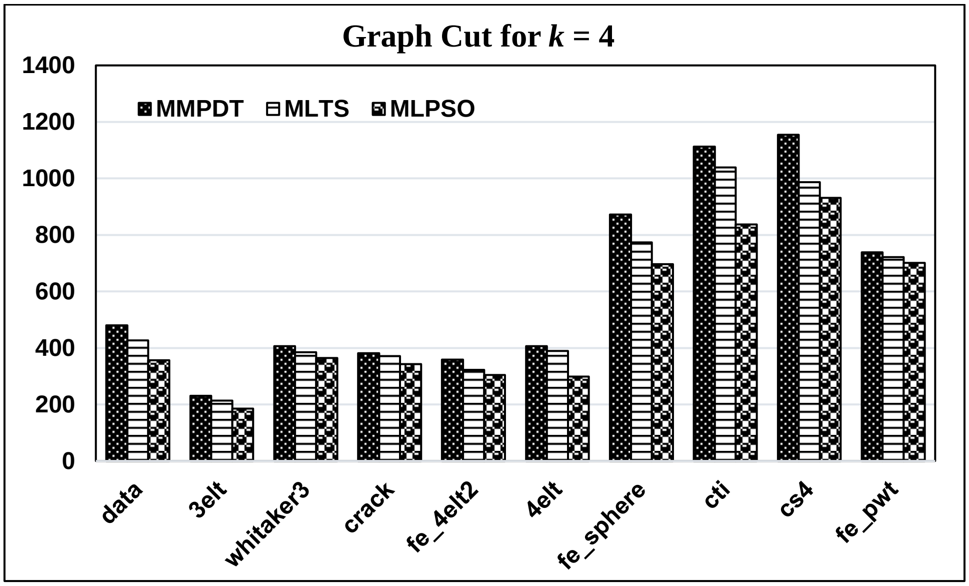


Fig. 4: Graph cut for *k* = 4

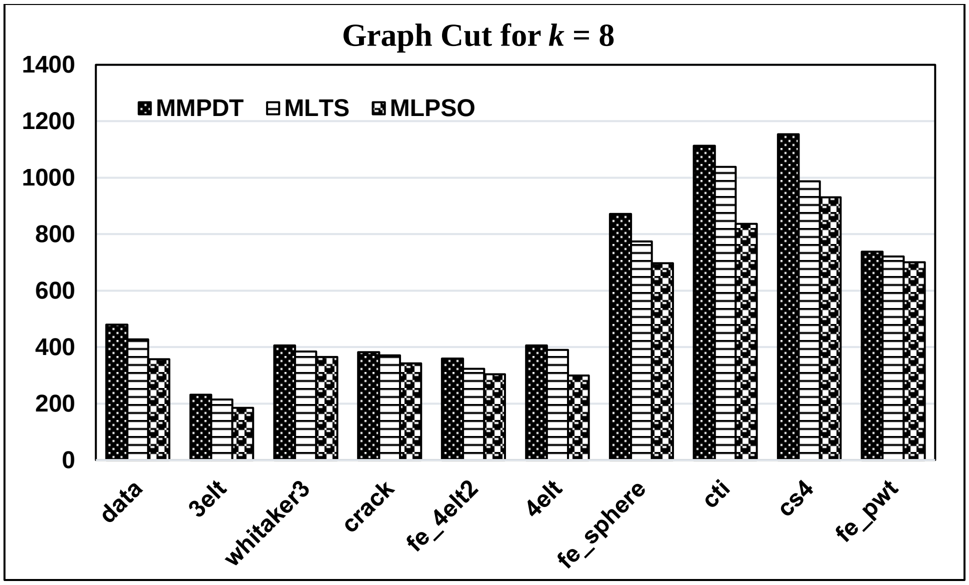
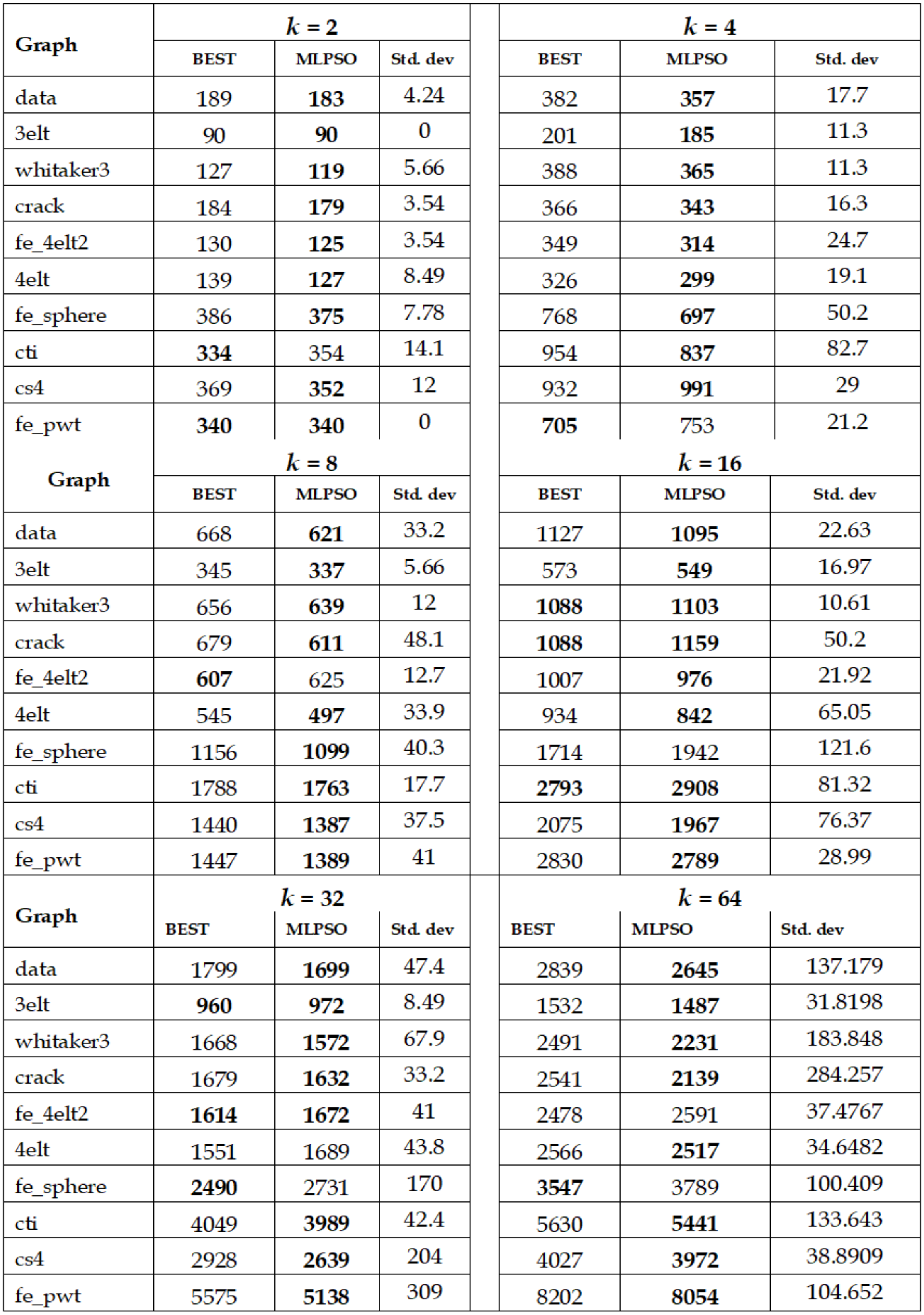


Fig. 5: Graph cut for *k* = 8

Heavy edge matching incorporated in MLPSO massively condenses the dimensions of the graph and assists greedy graph growing partitioning in producing improved stable partition. In addition to this discrete particle swarm optimization is implemented at the complicated and tedious refinement process it plays crucial role in reducing the algorithm run time while producing minimal cut and optimal partitioning. MLPSO is superior in comparison with other two algorithms. To evaluate the potential of MLPSO we have analyzed it in contrast to the best-balanced partitions that are deposited in the Graph Partitioning archive [33]. Maximum number of results in this archive are produced by the technique proposed by Schulz *et al* [34], by incorporating evolutionary technique with JOSTEL multilevel approach. Cut values produced by MLSO are highly improved than best partition cut values.



Graph	$k = 2$			$k = 4$		
	BEST	**MLPSO**	**Std. dev**	**BEST**	**MLPSO**	**Std. dev**
data	189	**183**	4.24	382	**357**	17.7
3elt	90	**90**	0	201	**185**	11.3
whitaker3	127	**119**	5.66	388	**365**	11.3
crack	184	**179**	3.54	366	**343**	16.3
fe_4elt2	130	**125**	3.54	349	**314**	24.7
4elt	139	**127**	8.49	326	**299**	19.1
fe_sphere	386	**375**	7.78	768	**697**	50.2
cti	**334**	354	14.1	954	**837**	82.7
cs4	369	**352**	12	932	**991**	29
fe_pwt	**340**	**340**	0	**705**	753	21.2
Graph	$k = 8$			$k = 16$		
	BEST	**MLPSO**	**Std. dev**	**BEST**	**MLPSO**	**Std. dev**
data	668	**621**	33.2	1127	**1095**	22.63
3elt	345	**337**	5.66	573	**549**	16.97
whitaker3	656	**639**	12	**1088**	**1103**	10.61
crack	679	**611**	48.1	**1088**	**1159**	50.2
fe_4elt2	**607**	625	12.7	1007	**976**	21.92
4elt	545	**497**	33.9	934	**842**	65.05
fe_sphere	1156	**1099**	40.3	1714	1942	121.6
cti	1788	**1763**	17.7	**2793**	**2908**	81.32
cs4	1440	**1387**	37.5	2075	**1967**	76.37
fe_pwt	1447	**1389**	41	2830	**2789**	28.99
Graph	$k = 32$			$k = 64$		
	BEST	**MLPSO**	**Std. dev**	**BEST**	**MLPSO**	**Std. dev**
data	1799	**1699**	47.4	2839	**2645**	137.179
3elt	**960**	**972**	8.49	1532	**1487**	31.8198
whitaker3	1668	**1572**	67.9	2491	**2231**	183.848
crack	1679	**1632**	33.2	2541	**2139**	284.257
fe_4elt2	**1614**	**1672**	41	2478	2591	37.4767
4elt	1551	1689	43.8	2566	**2517**	34.6482
fe_sphere	**2490**	2731	170	**3547**	3789	100.409
cti	4049	**3989**	42.4	5630	**5441**	133.643
cs4	2928	**2639**	204	4027	**3972**	38.8909
fe_pwt	5575	**5138**	309	8202	**8054**	104.652

Table1: Comparison of MLPSO Cut with Best Cut

7. CONCLUSION

Multilevel Particle Swarm Optimization algorithm for the stable *k* partitions of graph is developed in this paper. It uses the idea of edge matching, greedy graph partitioning and incorporates it with discrete particle swam optimization for the refinement process. We widely assessed performance on a group of graphs in benchmark dataset and partitioning archive. The outcomes demonstrates that the developed method performs improved than the other methods. Optimal cut values in lesser computation time span are the advantages of the proposed method. New hybrid approaches by combination of swarm intelligence-based optimization methods can be developed to enhance the performance.

CONFLICT OF INTERESTS

The author(s) declare that there is no conflict of interests.

REFERENCES

[1] H. N. Djidjev, G. Hahn, S. M. Mniszewski, C. F. A. Negre, A. M. N. Niklasson, Using graph partitioning for scalable distributed quantum molecular dynamics, Algorithms. 12 (2019), 187.

[2] D. Bader, H. Meyerhenke, D. Wanger, Challenge graph partitioning and graph clustering. DIMACS, 2003.

[3] S. D. Kapade, S. M. Khairnar, B. S. Chaudhari, Enhanced graph based normalized cut methods for image segmentation, ICTACT J. Image Video Proc. 5 (2014), 907 – 912.

[4] N. Ghorpade, H.R. Bhapkar, Brain MRI segmentation and tumor detection: challenges, techniques and applications, in: 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), IEEE, Madurai, India, 2021: pp. 1657–1664.

[5] H.R. Bhapkar, P.N. Mahalle, P.S. Dhotre, Virus graph and COVID-19 pandemic: a graph theory approach, in: A.-E. Hassanien, N. Dey, S. Elghamrawy (Eds.), Big Data Analytics and Artificial Intelligence Against COVID-19: Innovation Vision and Approach, Springer International Publishing, Cham, 2020: pp. 15–34.

[6] S.D. Kapade, S.M. Khairnar, B.S. Chaudhari, Evaluation and performance analysis of graph theoretical methods for image segmentation, in: International Conference on Information Communication and Embedded Systems (ICICES2014), IEEE, Chennai, India, 2014: pp. 1–7.

[7] G. Karypis, V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs, SIAM J. Sci. Comput. 20 (1998), 359–392.

[8] C. Walshaw, M. Cross, JOSTLE: parallel multilevel graph-partitioning software–an overview, Mesh Partition. Techn. Domain Decomp. Techn. 10 (2007), 27-58.

[9] C. Chevalier, F. Pellegrini, PT-Scotch: A tool for efficient parallel graph ordering, Parallel Comput. 34 (2008), 318–331.

[10] H. Meyerhenke, B. Monien, T. Sauerwald, A new diffusion-based multilevel algorithm for computing graph partitions of very high quality, in: 2008 IEEE International Symposium on Parallel and Distributed Processing, IEEE, Miami, FL, USA, 2008: pp. 1–13.

[11] A. Abou-Rjeili, G. Karypis, Multilevel algorithms for partitioning power-law graphs, in: Proceedings 20th IEEE International Parallel & Distributed Processing Symposium, IEEE, Rhodes Island, Greece, 2006: p. 10.

[12] P. D. Hough, P. J. Williams, Modern machine learning for automatic optimization algorithm selection, in Proceedings of the INFORMS Artificial Intelligence and Data Mining Workshop, 2006, pp. 1–6.

[13] S. M. Khairnar, S. Kapade, N. Ghorpade, Vedic mathematics-the cosmic software for implementation of fast algorithms, Int. J. Comput. Sci. Appl. 1 (2012), 1–7.

[14] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95 - International Conference on Neural Networks, IEEE, Perth, WA, Australia, 1995: pp. 1942–1948.

[15] E.K. Burke, M. Gendreau, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, R. Qu, Hyper-heuristics: a survey of the state of the art, J. Oper. Res. Soc. 64 (2013), 1695–1724.

[16] S.N. Ghorpade, M. Zennaro, B.S. Chaudhari, R.A. Saeed, H. Alhumyani, S. Abdel-Khalek, Enhanced differential crossover and quantum particle swarm optimization for IoT applications, IEEE Access. 9 (2021), 93831–93846.

[17] W.E. Donath, A.J. Hoffman, Algorithms for partitioning of graphs and computer logic based on eigenvectors of connections matrices, IBM Technical Disclosure Bulletin, vol. 15, 1972.

[18] S.D. Kapade, S.M. Khairnar, B.S. Chaudhari, Recent trends in metaheuristic graph partitioning techniques. Int. J. Interdiscip. Res. Adv. Eng. 7.0 (2015), 71.0–90.0.

[19] S.D. Kapade, Swarm intelligence-based graph partitioning for image segmentation, in Chapter 3: Review of image segmentation methods. Thesis, Suresh Gyan Vihar University. (2015).

[20] S. Ghorpade, M. Zennaro, B. Chaudhari, Survey of localization for internet of things nodes: approaches, challenges and open issues, Future Internet. 13 (2021), 210.

[21] D.S. Johnson, C.R. Aragon, L.A. McGeoch, C. Schevon, Optimization by simulated annealing: an experimental evaluation; Part I, Graph Partitioning, Oper. Res. 37 (1989), 865–892.

[22] C.-E. Bichot, Metaheuristics versus spectral and multilevel methods applied on an air traffic control problem, IFAC Proc. 39 (2006), 493–498.

[23] A.J. Soper, C. Walshaw, M. Cross, A combined evolutionary search and multilevel optimization approach to graph partitioning, J. Glob. Optim. 29 (2014), 225-241.

[24] P. Kuntz, P. Layzell, D. Snyers, A colony of ant-like agents for partitioning in VLSI technology, in: Proceedings of the Fourth European Conference on Artificial Life, pp. 417-424, 1997.

[25] S. Bach, B. Huang, B. London, L. Getoor, Hingeloss Markov random fields: Convex inference for structured prediction, in Uncertainty in Artificial Intelligence, 2018.

[26] R. Green, S. Gadde, On the use of particle swarm optimization with breadth first search for partitioning large graphs, in North American Power Symposium 2011, Boston, Massachusetts, pp. 1-6, August 2011.

[27] S. Ghorpade, Airspace configuration model using swarm intelligence based graph partitioning, in: 2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), IEEE, Vancouver, BC, Canada, 2016: pp. 1–5.

[28] N. Ghorpade, H.R. Bhapkar, Enhanced N-Cut and watershed based model for brain MRI segmentation, Int. J. Eng. Adv. Technol. 9 (2020), 346-351.

[29] J. Kennedy, R.C. Eberhart, A discrete binary version of the particle swarm algorithm, in: 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, IEEE, Orlando, FL, USA, 1997: pp. 4104–4108.

[30] C. Walshaw, Multilevel refinement for combinatorial optimisation problems, Ann. Oper. Res. 131 (2004), 325–372.

[31] C. Walshaw and M. Cross, Jostle parallel multilevel graph partitioning software – an overview, in Mesh Partitioning Techniques and Domain Decomposition Techniques, pp. 27 – 58, 2017.

[32] U. Benlic, J.-K. Hao, An effective multilevel tabu search approach for balanced graph partitioning, Computers Oper. Res. 38 (2011), 1066–1075.

[33] C. Walshaw and M. Cross, Jostle parallel multilevel graph partitioning software – an overview, in Mesh Partitioning Techniques and Domain Decomposition Techniques, pp. 27 – 58, 2017

[34] M. Holtgrewe, P. Sanders, C. Schulz, Engineering a scalable high quality graph partitioner, Technical Report, 2009.