Benchmarking of Graph Partitioning Tools and Techniques

Anuja Bokhare, P S Metkewar

Abstract: ***In this paper the authors have used a systematic literature review to provide benchmarking on influencing parameters for graph partitioning tools, which is the principal contribution of the present paper. Tools are compared on the basis of parameters which will impact the performance of tool. The paper elucidates about the tools and techniques along with their features, merits and demerits and also highlighted on influencing parameters which is missing in other reviews. These techniques are analysed by identifying merits and demerits of each technique. This research paper can help the researchers to choose the appropriate tool or technique for their own partitioning problems. Also authors have suggested future research directions and anomalies for improvement in tools and techniques for Graph Partitioning.***

Keywords : Graph Partitioning tools, graph partitioning techniques, benchmarking and performance influencing parameters.

I. INTRODUCTION

Graph is a conceptual idea of representing any objects which are connected to each other in a form of relation i.e., nodes are connected via edges i.e., relationship. Graphs are often used as a construct for demonstrating an application problem. Dividing the graph into parts suggested as a best way to work with large graphs. Many real world problems with respect to graph are difficult to handle, partitioning them into smaller graphs would reduce the complexity of problem [1]. Complexity of graph concept studied and explained in [2] with the help of mathematical model. Graph partitioning can be viewed as a parallelization of graph where larger graphs are divided into smaller parts. Graph partitioning is used in many applications including parallel processing [3], road networks [4], image processing [5], VLSI design [6], social networks [7], and bio-informatics [8]. Graph partitioning is a technique to allocate the total graph data as a disjoint subset to a different device. The need of allocate huge graph data set is to process data efficiently and fast. Good graph partitioning algorithms always make an effort to reduce the communication between machines in their distributed environment and distribute vertices roughly equal to all the machines. Graph partitioning has been studying in the discipline between computer sciences and applied mathematics. Graph partitioning applications are complex networks such as biological network, social network, PPI network.

Revised Manuscript Received on November 15, 2019
***** Correspondence Author
Anuja Bokhare*, Symbiosis Institute of Computer Studies and Research, Symbiosis International (Deemed University), Pune, India.
Email: anuja.bokhare@gmail.com
P S Metkewar, Symbiosis Institute of Computer Studies and Research, Symbiosis International (Deemed University), Pune, India.
Email: pravin.metkewar@sicsr.ac.in

Other applications are Web Page Ranking, Road Networks, and VLSI Design.
The purpose of this paper is to give a structured overview of the existing advance literature. The work gives emphasis on recently used tools and techniques used during graph partitioning which is missing in other reviews. Our survey structured as follows. Section 2 introduces about the Graph Partitioning criteria. Section 3 discusses typical and popular Graph partitioning tools along with their features and also summarizes by providing benchmarking on various performance metrics of each tool. Section 4 discusses classic techniques used during Graph partitioning. These techniques are analysed by identifying merits and demerits of each technique. Finally, Section 5 points to conclusion and future direction for research with respect to tools and techniques improvement.

II. GRAPH PARTITIONING CRITERIA

Graph partitioning problem is from NP-hard problem category. Determinations to these problems can be is a given using heuristic or approximation algorithms. Graph partitioning performed on the data, which is in n graphical format. If G= (v, e) where v is the number of vertices and e is the number of edges in the given graph. Graph partitioning problem is to divide the given graph into small components with some criteria. Generally, these components should be small and there should be minimum number of connections or edges between these components. It shows a good partition. For an illustration, in k-way partitioning, graph G is partitioned into k equal components with minimum cut. A good quality partition is the one where the numbers of cuts are minimum between two partitions or there should be few connections between the partitions, which indicate the uniform graph partition [9, 10]. Various criteria's are followed to partition the graph. Some of them are minimum cut, normalized cut, average cut, partition size and between-ness of edges etc.

Tools and techniques assist in completing the any process. Similarly, graph partitioning can also be benefited with the use of fast as well efficient tools and techniques. Further section provides investigation of the same.

III. GRAPH PARTITIONING TOOLS

This section provides the details about the typical and popular Graph partitioning tools along with their features. This section ends with a benchmarking on various performance parameters of each tool. Further, couple of tools have been discussed thoroughly and analysed individually at possible extent:

A. JOSTLE

JOSTLE is a software package. It uses a parallel multilevel graph partitioning. It is written at University of Greenwich. JOSTLE is freely available for research and academic purposes. It is designed to partition unstructured meshes, which make use of distributed memory on parallel computers [12].

Features:

- It supports repartition and load-balance in existing partitions.
- It accomplishes repartition by displaying the mesh by means of an undirected graph and then applying related graph partitioning techniques.
- It uses a multilevel refinement strategy. It recursively iterate till the graph size decreases below threshold to define new graph. This new graph gets partitioned and again projected back through iteration to original graph.
- It uses a greedy refinement scheme.
- It can enhance an existing partition.
- It can find a high quality partition independent of the existing partition.
- It integrates load-balancing methods.
- It implements refinement strategies [12] that are interface optimisation, alternating optimisation and relative gain optimisation [11].

B. Gephi

Gephi is used for network and graph analysis. It is open source software. Large networks are displayed immediately and it uses 3D render engine to speed up the analysis. This is software for Exploratory Data Analysis. It supports multi-task architecture with flexibility to analyse and explore complex data sets and generates useful visual results. It offers simple and extensive access to network data and supports for different stages such as specializing, filtering, navigating, manipulating and clustering [13]. Figure.1 shows file format and matching encoding data patterns supported by Gephi.

Features:

- Mainly used by data analysts and scientists intense to discover and distinguish graphs.
- It is similar Photoshop tool but used for graph data through representation, manipulation of data, use of the structures, shapes and colours to investigate the hidden patterns.
- The aim of Gephi is to assist data analysts to build assumption, automatically investigate patterns, separate structure individualities or mistakes during data tracking [14].
- It is mainly used for visualization.
- Gephi runs on any operating system [15].
- It supports file formats like GEXF, Pajek NET, UCINET DL,GML,GDF, Netdraw VNA, GraphML, GraphViz DOT, CSV, Tulip TPL and Spreadsheet.
- It supports maximum number of nodes as compare to other tools [16].

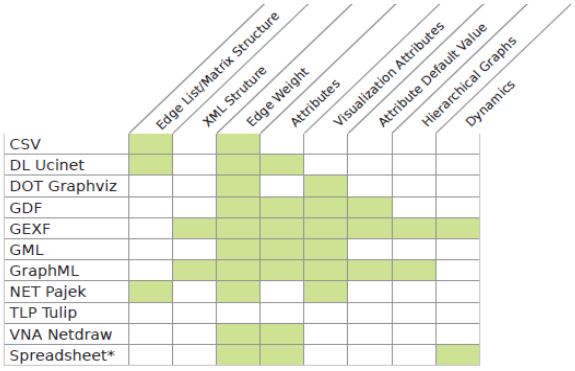

Figure 1: Source:- https://gephi.org/users/supported-graph-formats/

C. KaHIP (Karlsruhe High Quality Partitioning)

KaHIP is a software package. It is based on algorithms and algorithms KaFFPa (Karlsruhe Fast Flow Partitioner) that implements a multilevel graph partitioning [17] approach [18], KaFFPaE (KaFFPaEvolutionary) that implements a parallel evolutionary approach and KaBaPE algorithm uses a strict balance constraint that is suitable for small values while solving graph-partitioning problem. Balancing variants of these procedures supported by KaBaPE which can make infeasible partitions to become feasible. Applications related to partition the road network and social network [19], they uses same file format as of METIS and Chaco. *In* order to read files efficiently in parallel, may also use a binary file format. Tools are provided to convert the input file format into the binary file format.

Features:

- It is a standalone program [20].
- Provides operations like a mutation and combine. These are supported through KaBaPE algorithm that is evolutionary algorithm.
- It implements low-based methods.
- Half a billion edges from a web graph can be partitioned in a minute [20].
- It implements local, parallel as well as sequential meta-heuristics for searching.
- In future, algorithms are required for improving the maximum communication volume of a partition.

D. Scotch

It is used for parallel and sequential graph partitioning, static mapping, mesh partitioning, hypergraph partitioning and sparse matrix ordering. It's a software package with libraries defined for partitioning graphs [21]. For static mapping it uses dual recursive bi-partitioning algorithm. It uses methods like Fiduccia-Mattheyses, Greedy graph growing, Multi-level, Thinner, Vertex cover etc for graph separation. It accepts graphs in the form of adjacency lists [22][23].

Features:

- It is written in C and highly portable.
- It gives sequential graph partitioning with fixed vertices.
- It implements recursive multilevel bisection.
- It consists of parallel and sequential partitioning methods [24].

- It provides novel, quick, direct k-way partitioning and mapping algorithms
- In the sequential part of the library consist of multi-threaded, shared memory algorithms.
- Makes good memory utilization for large graph.
- Algorithms to build, check, display graphs / meshes and matrix patterns are supported.
- PT-SCOTCH uses the MPI interface [22].

E. Parkway 2.0

Pakway is a partitioning tool used for parallel multilevel hypergraph partitioning. Also used for k-way hypergraph partitioning problem [24]. Implementation of multilevel approach works well when coarsen graph is small, it not always scale well for hypergraph. Refinement methods need to be implemented carefully otherwise lead to scalability issues. Parkway does not provide guarantee on communication complexity between machines or processors [25].

Features:

- Supports distributed hypergraph partitions [25].
- Focus on hypergraph partitioning.
- It avoids single machine bottlenecks and communication overhead [26] [27].

F. Mondriaan

It is written in C. It follows sequential programming. It partitions a hypergraph and rectangular sparse matrix. The program is built on a recursive bi-partitioning algorithm that cuts the matrix vertically and horizontally similar to some of the well-known mondriaan pictures [28]. This is a multilevel algorithm. It decreases the communication costs. It distributes the computation and communication uniformly over the processors [29].

Features:

- It recommends both better quality and quicker partitioning using medium-grain partitioning method compared to the local best method.
- It uses Path Growing algorithm for matching vertices in coarsening phase which assures best matching's pictures [28].
- Counts the number of partitions where a hyperedge (edge connected to any number of vertices) is connected [30].
- Provide heuristic to calculate feasible solution [30].
- It gives low communication volume but no guarantee of quality of solution.
- It uses methods like mondriaan, mondriaan TB, Stairway, StairwayM for neighbour-finding and greedy, PGA, GPA and ROMA method as matching function.

G. METIS

METIS tool implements partitioning of meshes/graphs and generates sparse matrices [31]. METIS supports multilevel *k*-way multi-constraint partitioning and multilevel recursive-bisection schemes for partitioning [32-34]. It provides mpmetis program for partitioning meshes. It accepts graph file and mesh file as input file format. Writing coarser graph involves writing massive amounts of data to memory. There is no provision to change the imbalance for Metis easily [35].

Features:

- It generates partition with high quality [31].
- It is very fast [31]. Several millions of vertices take few seconds to partition in 256 parts.
- 95% of runtime is spent on coarsening and refinement [36].
- Metis is a standalone software and library.
- As compared to spectral partitioning algorithms METIS generates 10% to 50% better results [36].
- Produces low fill orderings.
- Reduced computational and storage requirement of sparse matrix factorization.

H. KMETIS

KMETIS is software programme written in C used to partition nodes in graph by using library from METIS [37] [38]. KMETIS gives partitions with minimum edge cut. Multilevel recursive bisection used by related program of KMETIS i.e., PMETIS. PMETIS gives good result when the number of parts is 8 or less.

Features:-

- It uses k-way partitioning. Works well when k>=8.
- KMETIS can be executed on a single processor as well in parallel too.
- Kmetis make use of a heuristic heavy edge matching.
- KMetis aims at greater partitioning speed [24].
- O(n + m + klog(k)) is the complexity of kmetis with k is number of partitions, n is the number of nodes and m is the number of edges [37].

I. Parallel Graph Partitioning and Fill-reducing Matrix Ordering (ParMETIS)

ParMETIS is an MPI-based parallel library. It supports different algorithms for partitioning which compute orderings with fill reducing of sparse matrices, unstructured graphs and meshes. It extends features provided by METIS and supports procedures for large scale problems and parallel computation [39].

Algorithms like adaptive repartitioning, parallel multilevel *k*-way graph-partitioning and multi-constrained partitioning are implemented in ParMETIS [33]. As compared to JOSTLE, ParMETIS gives fast execution with average 33% of total partition time [12]. The structure of the code is not well described in the manual or in the code [36] restricts to improve it further.

Features:

- It is parallelization of KMETIS. Ultimately parallel implementation of METIS algorithm [24].
- Performs mesh/graph partitioning, partitioning refinement and matrix reordering [36].
- Uses recursive bisection and geometric partitioning.
- Only distributed as software library that supports various partitioning algorithms.
- It produces partitions very fast for very large graph.
- To avoid loss of quality during partition, it gain benefit from geometry information of graph if presented.

- For multi-physics and multi-phase computation able to produce partitions.

J. hKMETIS

hMETIS mainly used to partition hypergraphs related to VLSI circuits. It works on hypergraph and circuit partitioning. It is a set of program which supports algorithms that follows multilevel hypergraph routines.

Features:

- It generates partitions with good quality.
- It is very quick [31]. In a couple of minute it partition large circuit with 100,000 vertices.
- It is faster as compared to algorithms like CLIP, KL and FM.
- It can able to generate good quality partitions due to its average cut feature with very few execution run.
- It is very well applicable for hypergraphs partitioning with good quality of partition [24].

K. Diffusion-based Partitioning (DibaP)

It is used for partitioning as well repartitioning. It is a multilevel algorithm based on diffusion for graph partitions. It's sequential and threads parallel variation while its MPI parallel version focuses on repartitioning, the sequential and thread-parallel edition performs graph partitioning and MPI parallel edition used for repartitioning [40-41]

Current partitioning libraries use versions of Kernighan–Lin (KL) heuristic during the process. Apart from fast processing the output generated by these libraries do not satisfy the user requirement. However, DibaP maintain the good features of slow algorithm along with high speed up in generating partitioning.

DibaP generates constant results which are better than METIS and JOSTLE during experiment conducted on standard benchmark graph. For a considerable number of partitions, it gives superior edge-cut values [42-43].

Features:

- DibaP also called as Bubble-FOS/C, essentially parallel algorithm which implements the Bubble framework using diffusion [24].
- It does adjustment in KL algorithm and performs load balancing in parallel applications.
- It implements diffusive approach for linear system.
- It employees AMG based schemes during coarsening.

L. PaToH

Umit V. Catalyurek has proposed a new hypergraph partitioning tool as an outcome of his PhD work i.e., Partitioning Tools for Hypergraph (PaToH) [44]. Multilevel hypergraph partitioning implemented in PaToH.

PaToH uses recursive bisection approach for k way partitioning. It implements hierarchical clustering and agglomerative clustering based on randomized matching and heuristics schemes. Heavy Connectivity Matching (HCM) scheme is used in PaToH [45] for partitioning [46] [30].

Features:

- Supports multi-constraint partitioning,
- Partitioning with fixed vertices [44].
- PaToH works on weighted nets.
- Fast, stable multilevel hypergraph partitions [47].
- PaToH involves 18 KLFM-based refinement algorithms [46].
- Re-allocate of memory happens only when hypergraph is changed otherwise no need to do it for each partition.
- New memory allocation has been carried out only when hypergraph is changed. Sometimes reallocation done for change in coarsening algorithm.

M. Chaco

Chaco [48] is named in honor of Chaco Canyon. Chaco is a software package for graph partitioning. It uses recursive approach for partitioning. The Kernighan & Lin method used to get good quality partitions produced as compared to other algorithms [49]. It supports wide variety of algorithms. It supports multilevel graph partitioning and spectral partitioning. It also uses KL-FM method for handling weighted graph. During parallel software development the performance depends on parameters i.e., how division or decomposition of work, node and process has been done. It is difficult to find which one is most optimal decomposition. To overcome this problem Chaco has been developed [50], which is a suite of algorithms for decomposition and further assign the task to parallel processor [48].

Features:

- It follows multilevel approach along with spectral partitioning technique [24].
- Calculation of eigenvalues for spectral partitioning made efficiently in Chaco [51].
- It is flexible i.e., supplementary method used if one of the methods fails to give result. Actually, the results of different methods can be compared and the best possible one selected even if a method does not fail [51].
- Simple, Inertial, Spectral, Kemighan-Lin and Multilevel Kemighan-Lin algorithms are implemented in chaco.
- Weighted graph, lazy initiation and arbitrary number of sets handled by using generalized Kernighan-Lin/Fiduccia-Mattheyses algorithm.
- Mapping of different graphs onto targeted parallel architecture is improved with the development of skewed partitioning.
- Output partitions are improved by processing in different ways [52].
- It generates partitions by applying iterative and recursive approach. While applying this it confirms load is equally balanced among processors by minimizing edge cut value [51].

N. Zoltan

Among available partitioning software packages like ParMetis, PT-Scotch and Zoltan, [53] Zoltan used for verifying performance of parallel and hypergraph (PHG) partitions. Recursive bisection method is used for partitioning in Zoltan.

Zoltan uses randomization because of which result may vary for each run. Authors have worked on edge cut and communication volume parameter for partitioning. Evaluation done on graph selected from DIMACS collection. The sample for testing consist only symmetric graph. However Zoltan can work for all types of data.

Features:

- Focus on hypergraph partitioning.
- Used for load balancing and parallel hypergraph partitioning which are mostly required for dynamic and large problems.
- It performs data migration during partitioning.
- Its supports parallel coloring algorithms.
- It supports interfaces for vertex ordering of graphs through parallel ordering algorithms such as PT-Scotch and ParMETIS.
- Communication volume in traditional graph partitioning reduced by 10-20% by the use of hypergraph partitioning
- As compare to Parkway, it gives partitions much faster.
- It uses Recursive Coordinate Bisection and Recursive Inertial Bisection algorithms [54].

O. Party

The PARTY partitioning tool supports different partitioning methods divided in global and local methods [55]. Global methods called "constructional heuristics, “which generate a balanced partition [35] and local methods called "improvement heuristics," try to improve the quality of partition accepted from global method. Global methods are -Optimal Method, linear method, Scattered Method, Random Method, Gain Method, Farhat Method, Coordinate Sorting Method, Multilevel Method, Spectral Method, Inertial Method. Local methods are -Keraighan-Lin Method, Helpful-Set Method [35]. PARTY did not work correctly for any given data set. It takes more time than pmetis and Jostle for partitioning [57].

Features:

- It is a well-known sequential and parallel graph partitioning technique.
- Implemented algorithms are Bubble / Sharp – optimizatio and Helpful Sets.
- It follows LAM matching algorithm during coarsening phase [56].
- Party produces 5% better result quality than pmetis [57] but performs slower.
- Performs best for small numbers.

After reviewing the literature on tools for graph partitioning; Table I recapitulate the benchmarking of reviewed tool on the basis of performance parameters namely; name of the tool, licensing, standalone or network based, its software and hardware requirement, tool best suited or applicable for, limitations, input and output format used by tool, capacity to handle graph size, techniques used for graph partitioning, approach used during graph partitioning and refinement methods used by the tool. On the basis of benchmarking performed so far we may conclude that; merits and demerits of each tool have been reported and researcher would be able to choose the appropriate tool as per the requirement.

Table I Benchmarking of Graph partitioning tools based on influencing performance parameters

Sr No.	Name	License	Software Requirement	Hardware Requirement	Best for	Limitations	Input file format	Output file format	Technique used	Approach used	Refinement scheme	References
1	JOSTLE	NA	UNIX operating system. ANSI C compiler	NA	Unstructured meshes, sequential and parallel graph partitioner	Lack of expressibility	.graph	.ptn	Multilevel partitioning	Diffusive load-balancing, sequential and parallel	Greedy refinement, interface optimization, alternating optimization and relative gain optimization.	[11] [58]
2	Gephi	GPL	Java JRE, Windows, Mac OS X and Linux,	500 Megahertz CPU and 128 megabytes of RAM, OpenGL	Exploratory Data Analysis ,visualizing network and complex systems	Visualization size limitation for large networks, Gephi has a limit based on amount of memory allocated to it in JVM	GEXF, GDF, Pajek, DOT, GraphML, UCINET, CSV	SVG,PNG or PDF	Multilevel	Multilevel	Local refinement	[13] [19][14] [15][59][20] [60]
3	KaHIP	GPL	Software packages-Scons, g++, Argtable , OpenMPI ,Linux (32/64 bit) Mac OS	NA	Balanced graph partitioning	Computation overhead	.graph, binary format	.txt	Iterative multi-leve, parallel and sequential meta-heuristics	Flow-based methods, more-localized local searches	Several parallel and sequential meta-heuristics	[18][20] [53]
4	Scotch	CeCILL-C	POSIX operating system. ANSI C compiler	NA	Sequential and parallel graph partitioning,graph and mesh/hypergraph partitioning, graph clustering, and sparse matrix ordering	File compression issues, machine word size issues	Matrix Market format, the Harwell-Boeing collection format , the Chaco/MeTiS graph format , and the Scotch format	Matrix Market format, the Chaco/MeTiS graph format and the Scotch source graph and geometry data format.	Multilevel recursive bisection ,band ,diffusion techniques, sequential and parallel	Multilevel	KL-FM, Gibbs-Poole-Stockmeyer ,simulated annealing, quadratic assignment, genetic algorithms	[61][21][37]
5	Parkway	NA	NA	NA	Parallel multilevel hypergraph partitioning	Refinement methods need to be implemented carefully otherwise lead to scalability issues	.graph	.graph	Parallel multilevel hypergraph	Multilevel	Greedy graph growing method	[33] [25][29][59]

NA-Not Available , GPL- General Public License

Table I Benchmarking of Graph partitioning tools based on influencing performance parameters

Sr No.	Name	License	Software Requirement	Hardware Requirement	Best for	Limitations	Input file format	Output file format	Technique used	Approach used	Refinement scheme	References
6	Mondriaan	GPL	ANSI C compiler, Linux	NA	Partitions a hypergraph and rectangular sparse matrix.	No guarantee of quality of solution	.graph	.graph	Recursive bipartitioning , mondriaan, mondriaan TB, Stairway, StairwayM ,greedy, PGA, GPA and ROMA method	Recursive multilevel	Local refinement method	[28] [62][30]
7	METIS	Apache	ANSI C compiler ,Linux, SunOS, and OSX	Support for 64 bit architectures	For large irregular graphs, partitioning large meshes, and computing fill-reducing orderings of sparse matrices	Migration issues	.graph, Mesh file	Partition file, Ordering file	Multilevel recursive bisection or the multilevel k-way partitioning	Multilevel	KL-FM	[38][33] [35][31][36]
8	KMETIS	Apache 2.0	ANSI C compiler ,Linux	Support for 64 bit architectures	Gives partitions with minimum edge cut	NA	.graph	.graph	Multilevel recursive bisection	Multilevel	k-way local search	[37] [38]
9	ParMETIS	Apache	ANSI C and uses MPI for inter-processor, Linux, SunOS, and OSX	Support for 64 bit architectures	Parallel multilevel k-way graph-partitioning, adaptive repartitioning, and parallel multi-constrained partitioning	Cannot be used on a single processor, need to be distributed among the processors initially	.graph, adjacency structure of the graph, mesh file	Dual graph	Multilevel k -way multi-constraint partitioning algorithm,coordinate-based space-filling curves method	Multilevel	Multilevel k -way refinement algorithm.	[11] [33] [39] [36]
10	hMETIS	Apache	Sun, SGI, Linux	IBM	For large hypergraphs	Computational problems	.graph	.graph	Multilevel hypergraph partitioning , multilevel recursive bisection	Multilevel	KL-FM	[36][59]
11	DiBaP	NA	NA	NA	For partitioning as well repartitioning	NA	.graph	.graph	Multilevel algorithm based on diffusion	Algebraic multigrid and graph based diffusion	Kernighan–Lin	[40-43]

NA-Not Available , GPL- General Public License

Table I Benchmarking of Graph partitioning tools based on influencing performance parameters

Sr No.	Name	License	Software Requirement	Hardware Requirement	Best for	Limitations	Input file format	Output file format	Technique used	Approach used	Refinement scheme	References
12	PaToH	BSD	Linux, Mac OS X 10.6,Sun Solaris	32-bit x86-based, 64-bit x86-based	Multilevel Hypergraph Partitioning	NA	.hygr	.hygr.part.K	Multilevel hypergraph partitioning	Multilevel	Multilevel k -way refinement algorithm.	[46] [30] [44]
13	Chaco	GPL	ANSI C, unix	NA	Sequence graph, ordering for sparse matrix factorization.	NA	.graph,adjacency list	.graph	Spectral method	Multilevel	Multilevel k -way refinement algorithm.	[49][48] [51] [50] [59]
14	Zoltan	BSD	C++ and Fortran90 compilers	MPI	Graph coloring and ordering, load balancing and parallel data management.	More expensive than geometric methods	.graph	NA	Parallel multilevel hypergraph partitioning, dynamic partitioning,	Multilevel	Geometric, graph-based, and hypergraph-base	[53][47]
15	Party	NA	NA	NA	Partitioning library	Limited partitioning methods	Adjacency file	Directed to stdout	Sequential and parallel graph partitioning	Bubble/Sharp-optimization and Helpful Sets	KL-FM, recursive partitioning	[36] [55] [56][57]

NA-Not Available , GPL- General Public License

After reviewing the literature on tools for graph partitioning, Table I recapitulate the benchmarking of reviewed tool on the basis of performance parameters namely; name of the tool, licensing, standalone or network based, its software and hardware requirement, tool best suited or applicable for, limitations, input and output format used by tool, capacity to handle graph size, techniques used for graph partitioning, approach used during graph partitioning and refinement methods used by the tool. On the basis of benchmarking performed so far we may conclude that; merits and demerits of each tool have been reported and researcher would be able to choose the appropriate tool as per the requirement.

IV. GRAPH PARTITIONING TECHNIQUES

This section reviews various popular techniques used during Graph partitioning. Major optimization techniques are considered essentially for best partitioning with optimized cut value.

A. Simulated Annealing (SA)

SA is an optimization technique rather a metaheuristic used to approximate global optimization in a large search space. A process in which a solid is cooled slowly until its structure is eventually frozen at a minimum energy configuration called annealing, a physical process [63]. A simulated annealing can be used in graph partitioning for finding balanced partitioning as well as in multilevel partitioning as a refinement tool. Initial solution is obtained by generating random partition. [64]. Greedy heuristic has been used if final solution is unbalance. The heuristic repeats until the two sets of partition become balanced.

The parameter setting is dependent on the problem instance. The running time shows improvement on small neighborhood size. A vertex in the larger set can be identified to move to the opposite set with the increase in the cut size and moved it. Best feasible solution found or some earlier feasible solution found along the way declared as output. The methods used for VLSI Circuit Partitioning [63].

B. Tabu Search

A tabu search technique is used for solving graph partitioning problem and other problems from combinatorics. This search techniques moves best solution from the neighborhood to an upgraded one iteratively till stopping condition satisfies [65]. Tabu list records the move history to avoid solution cycling so the name of method is tabu search [24].

The stopping instruction followed for tabu search can be a fixed number of iteration, CPU time or number of iteration without any change in process [66]. It can be any iteration where no further best move for local neighborhood exists. Parameter setting is responsible for getting global optimum solution. It is used for balanced partitioning. Refinement suggested in tabu search by combining it with heuristic method [67].

C. Spectral Method

Spectral methods deal with the graph's mathematical representation rather than graph itself. It models the graph by transforms into continuous function [61]. Then minimization of this model is calculated by Laplacian matrix of the graph. Spectral method makes use of eigenvectors and eigenvalue for graph partitioning [68]. It finds a splitting value for partitioning vertices from G by evaluating Fiedler vector i.e second smallest eigenvalue. Various partitioning algorithm based on spectral method are bisection, K-way and Lanczos partitioning algorithm [63] [69].

D. Swarm Intelligence

Simple agents which act together along with their environment and are huge in number are known as swarm. Robust, fast and low cost solutions are obtained for complex problems using swarm based algorithms. Swarm intelligence used to model collective behavior social swarms. It's a branch of AI. Ant colony optimization (ACO) and particle swarm optimization (PSO) are mostly used models of swarm intelligence [70]. ACO gives good partitioning result. To improve the performance of PSO there is need to combine with other heuristic techniques [71].

E. Mean Field Annealing (MFA)

MFA which has combine features of simulated annealing and Hopfield neural network used for solving graph partitioning problems. Results show good performance [72]. In this technique each edge attracts adjacent nodes into the same bin with a force proportional to its weight, this forms clusters. The average spin of a node can be determined from its mean field [73]. MFA is 40 times faster than SA [17].

Table II gives benchmarking of reviewed techniques used for graph partitioning. Their merits and demerits are summarized along with references.

Table II: Benchmarking of Techniques used for Graph Partitioning				
Sr No.	**Technique Name**	**Merits**	**Demerits**	**References**
1	Simulated Annealing	1)Flexible 2)Mostly used for smaller graphs. 3)Avoid getting stuck in local optimum	1) Execution time is longer 2) Adaptation is very slow as compare to other method 3) Needs several values of different aspects to be tried on to get result.	[63-65]
2	Tabu Search	1) Deterministic 2) Gives high quality solutions over previously obtained one with less computational effort 3) Uses Tabu List	1) More aggressive metaheuristic method 2)Too many parameters to be determined 3)Number of iterations can be more	[65-66] [74][67] [24]
3	Spectral Method	1) Computationally fast and easy to implement 2) Robust in nature. 3)Finds optimal solution	1) Computationally expensive	[61][63][68]
4	Swarm Intelligence	1) Flexible 2) Robust in nature 3) Decentralization and self-organization.	1) Execution time is too long	[70-71]
5	Mean Field Annealing	1) Computationally efficient 2) Fast in processing	1) Sometimes gives poor quality solution	[17][72][73][75]

Some of the key contributions made by this research are as below:

• The major focus of graph partitioning is based on lower cut values and few connections between paths which helps to improve the quality of the graph.

• As compared to other optimization algorithms, the time of convergence in spectral bisection, recursive bisection and multilevel graph partitioning algorithm is reduced significantly. A multilevel graph partitioning [2][60] algorithm converges very fast and hence saves the time.

• Couple of algorithm focused on heavy edge matching and greedy graph growing partitions. This mechanism has been developed for partitioning graphs. The current artifacts insight on results that deals with the cut values and computational time is reduced as compared with METIS, CHACO and MITS algorithms.

The current results are really promising and it can be utilized in real life applications. In near future, there is a scope for combining above techniques together in order to achieve good graph partitions.

V. DISCUSSION

In the current study, most popular graph partitioning tools are reviewed for benchmarking purpose. Few tools are freely available and few are license based. Gephi found to be the best tool in terms of visualization and for performing analysis but scalability issue still exists there. This problem could be resolved by using virtual memory allocation techniques.

Parkway, Scotch, Mondriaan, hMETIS, PaToH are developed to work on hypergraphs. In this case to avoid the scalability issues, refinement schemes need to be implemented carefully. Spectral methods implemented by Chaco uses adjacency list as a input format. In case of large graph of more than 1000 nodes, memory requirement is very high to handle such a large adjacency list but there is no such insight discussed. Zoltan, KAHIP and hMETIS found to be more expensive in terms of computation. However, complex computing issues need to be resolved through parallel execution on different processors. Precisely major issues or problems occur due to data size, complexity in data, demoralized data and noisy data. In such cases data preprocessing should be done in optimized way, which may help to resolve the issues up to some extent. Kernighan–Lin, Fiduccia-Mattheyses, greedy and genetic algorithm based refinement schemes helps in getting better output. Most of the tools drawing graph on multilevel approach for performing partitioning which can also be improved using neural network based approaches for getting good quality partitions.

Apart from input data format, data size, algorithms implemented for partitioning and refinement of the same, techniques plays a crucial role in getting good quality partitions. Different techniques are implemented by different graph partitioning tools based on their applicability and compatibility with available hardware and software. Techniques helps to refine the output more optimize manner i.e near to the solution.

In current benchmarking study simulated annealing founds to be more popular and good one than other studied techniques. Spectral methods; due to eigenvalue calculation found to be computationally more expensive.

But this technique gives more optimal solution and proved in many algorithms. Tabu search and swarm intelligence found to be computationally faster than other techniques. In short, there is a need of refine the techniques which help to improve the partition quality by providing most optimal solution. Also helps to manage the partition on its own by considering the existing data size, visualization issues and algorithm implemented.

VI. CONCLUSION

The current paper is mainly focuses on benchmarking of graph partitioning tools and techniques. This paper gives review of various popular tools and technique along with benchmarking in the context of graph partitioning. Surveyed tools are JOSTLE, Gephi, KaHIP, Scotch, Parkway, Mondriaan, METIS, KMETIS, ParMETIS, hMETIS, DiBaP, PaToH, Chaco, Zoltan and Party. It is observed that these partitioning tools are useful for partitioning unstructured meshes, sequential and parallel graph partitioning, graph clustering, hypergraph, irregular and other graph partitioning. Capacity of tool to partition the graph can be increased by the increase in memory. Techniques and approaches used by tools are multilevel, iterative multilevel, multiway and some uses heuristic based. All tools follow the refinement phase using local and global methods. Due to copyright issue some metrics are not available for comparison.

Broadly explained about techniques used for partitioning and few are surveyed namely Simulated Annealing, Tabu Search, Spectral Method, Swarm Intelligence and Mean Field Annealing. Tabu search and simulated annealing are mostly used. Simulated annealing found to be older but good optimization method. Genetic algorithm and neural network based swarm intelligence is robust but it takes more execution time. Spectral methods are computationally expensive due to the calculations of eigenvalue and eigenvectors. Study restricted due to the copyright issue of available tools and techniques. Manuals explained are not up to date for new user. The current benchmarking results are really promising and it can be utilized in real life applications. In near future, there is a scope for combining above techniques together in order to achieve good graph partitions. The paper can help the researchers to choose the appropriate tools and technique for their own partitioning problems based on influencing parameters identified. Future research can be carried out to speed up the partitioning process and generate good quality partitions using neural network by combining above techniques. The present work has considerable theoretical value and can be useful to the policy makers or practitioner and researchers to choose the appropriate tools and technique for their own partitioning problems. Also authors have suggested future research directions and incongruity for improvement in tools and techniques for Graph Partitioning.

REFERENCES

1. G. O. Young, "Synthetic structure of industrial plastics (Book style with paper title and editor)," in *Plastics*, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.
2. W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.
3. H. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer-Verlag, 1985, ch. 4.
4. B. Smith, "An approach to graphs of linear forms (Unpublished work style)," unpublished.
5. E. H. Miller, "A note on reflector arrays (Periodical style—Accepted for publication)," *IEEE Trans. Antennas Propagat.*, to be published.
6. J. Wang, "Fundamentals of erbium-doped fiber amplifiers arrays (Periodical style—Submitted for publication)," *IEEE J. Quantum Electron.*, submitted for publication.
7. C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.
8. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interfaces(Translation Journals style)," *IEEE Transl. J. Magn.Jpn.*, vol. 2, Aug. 1987, pp. 740–741 [*Dig. 9th Annu. Conf. Magnetics* Japan, 1982, p. 301].
9. M. Young, *The Techincal Writers Handbook.* Mill Valley, CA: University Science, 1989.
10. (Basic Book/Monograph Online Sources) J. K. Author. (year, month, day). *Title* (edition) [Type of medium]. Volume(issue). Available: http://www.(URL)
11. J. Jones. (1991, May 10). Networks (2nd ed.) [Online]. Available: http://www.atm.com
12. (Journal Online Sources style) K. Author. (year, month). Title. *Journal* [Type of medium]. Volume(issue), paging if given. Available: http://www.(URL)
13. Muttipati, A.S and Padmaja, P. "Analysis of Large Graph Partitioning and Frequent Subgraph Mining on Graph Data". *International Journal of Advanced Research in Computer Science*, 2015; 6, 29-40
14. Bokhare, Anuja and P. S. Metkewar. "Mathematical Model for quantification of Graph theory Concepts". In: Proceedings of the 2017 Second IEEE International Conference on Electrical, Computer and Communication Technologies, Coimbatore, Tamil Nadu, India, 2017, p.1-7.
15. Abou-Rjeili, A and Karypis, G. "Multilevel algorithms for partitioning power-law graphs". In: Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International. IEEE. In: IPDPS'06 Proceedings of the 20th IEEE International conference on Parallel and distributed processing, Rhodes Island, Greece, April 2006, p.10-PP
16. Gonzalez, H., Han, J., Li, X., Myslinska, M., Sondag, J.P. "Adaptive fastest path computation on a road network: a traffic mining approach". In: Proceedings of the 33rd International Conference on Very large data bases. VLDB Endowment, Vienna, Austria, September 2017, p. 794-805
17. Grady, L. and Schwartz, E.L. "Isoperimetric graph partitioning for image segmentation". *IEEE transactions on pattern analysis and machine intelligence*, 2006; 28(3), 469-475.
18. Kahng, A.B., Lienig, J., Markov, I.L., Hu, J. *VLSI physical design: from graph partitioning to timing closure*. Springer Science & Business Media; 2011 Jan 27
19. Aggarwal, Charu C., 2011. "An introduction to social network data analytics". *Social network data analytics*, Springer, Boston, MA, 2011. 1-15.
20. Yu Z, Wong HS, Wang H. "Graph-based consensus clustering for class discovery from gene expression data". *Bioinformatics*. 2007 Sep 14; **23**(21), 2888-96.
21. Buluç, Aydın, et al. "Recent advances in graph partitioning". *Algorithm Engineering*. Springer, Cham, 2016; 117-158.
22. Fern XZ, and Brodley CE. "Solving cluster ensemble problems by bipartite graph partitioning". *In:*Proceedings of the twenty-first international conference on Machine learning Banff, Canada, 2004 Jul 4, p. 36. ACM.
23. Walshaw, Chris, and Mark Cross. "JOSTLE: parallel multilevel graph-partitioning software–an overview". *Mesh partitioning techniques and domain decomposition techniques.* 2007; 27-58.
24. Walshaw, Chris, and Mark Cross. "Parallel optimisation algorithms for multilevel mesh partitioning". *Parallel Computing. 2000;* **26**.12, 1635-1660.
25. Gephi: Making your relational data pretty, Jesse Fagan, LINKS Center or Social Network Analysis, Available at: http://uknowledge.uky.edu/cgi/viewcontent.cgi?article=1006& context= rdsc_ workshops, accessed 12 June, 2018.
26. The open Graph Viz Platform. Available at:https://gephi.org/, accessed September 2018.
27. Installing Gephi. Available at: https://gephi.org/users/install/ , accessed 7 March, 2018

28. David Combe, Christine Largeron, Elod Egyed-Zsigmond, Mathias Géry. "A comparative study of social network analysis tools". *WEB INTELLIGENCE & VIRTUAL ENTERPRISES*, Oct 2010, SaintEtienne, France. ffhal-00531447f
29. Bofill, P., Guimera, R., Torras, C. "Comparison of simulated annealing and mean field annealing as applied to the generation of block designs". Neural Networks, 2003; 16(10), 1421-1428.
30. KaHIP - Karlsruhe High Quality Partitioning, Overview. Available at: http://algo2.iti.kit.edu/kahip, accessed 19 June, 2018.
31. Sanders, Peter, and Christian Schulz. KaHIP v2. 0--Karlsruhe High Quality Partitioning--User Guide. arXiv preprint arXiv:1311.1714 , 2013
32. Spectral clustering of large network, Available at: http://ondemand.gputechconf.com/gtc/2017- / presentation / s 7241-fender-spectral-clustering-large-networks.pdf, accessed 22 October, 2018.
33. Updates on SCOTCH, Available at: https://www.labri.fr/perso/pelegrin/scotch/, accessed 18 December, 2018
34. Pellegrini, François. Scotch and libScotch 5.1 user's guide. 2008.
35. Chevalier, Cédric, and François Pellegrini. PT-Scotch: "A tool for efficient parallel graph ordering". *Parallel computing*, 2008; **34**.6-8, 318-331.
36. Graph Partition .Software tools, Available at: https:// en.wikipedia.org/wiki/Graph_partition #Software_tools ,accessed 29 June 2019
37. Kabiljo I, Karrer B, Pundir M, Pupyrev S, Shalita A. Social hash partitioner: a scalable distributed hypergraph partitioner. *Proceedings of the VLDB Endowment*. 2017; DOI:10.14778/3137628.3137650 ;**10**(11):1418-29.
38. SWMATH, Parkway, Available at: http://www.swmath.org/software/12863 accessed 17 October, 2018
39. Trifunovic A, Knottenbelt WJ. Parkway 2.0: "A parallel multilevel hypergraph partitioning tool".In: International Symposium on Computer and Information Sciences. Springer, Berlin, Heidelberg, 2004, p. 789-800
40. Bisseling, R.H. and Flesch, I. "Mondriaan sparse matrix partitioning for attacking cryptosystems by a parallel block Lanczos algorithm–a case study". *Parallel Computing*, 2006; 32, p.551-567.
41. Mondriaan for sparse matrix partitioning, available at: http:/ /www. staff.science. uu.nl /~bisse10 1/ Mondriaan/ ,accessed 13 January, 2019.
42. Uçar, Bora, Ümit V. Çatalyürek, and Cevdet Aykanat. "A matrix partitioning interface to PaToH in MATLAB". *Parallel Computing*, 2010, 36, p 254-272.
43. hMETIS - Hypergraph & Circuit Partitioning, Available at: http://glaros.dtc. umn. Edu / gkhome / metis / hmetis/overview, aaccessed 26 August, 2019.
44. Karypis, George and Vipin Kumar. MeTis: "Unstructured Graph Partitioning and Sparse Matrix Ordering System", Version 4.0, 2009.
45. Sui X, Nguyen D, Burtscher M, Pingali K. "Parallel graph partitioning on multicore architectures". *In:* International Workshop on Languages and Compilers for Parallel Computing, Springer, Berlin, Heidelberg 2010 Oct 7, p. 246-260.
46. METIS A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices Version 5.1.0, Available at: file:///C:/Users/Anuja/Downloads/manual.pdf, accessed 12 July 2019.
47. Preis, Robert, and Ralf Diekmann. *The PARTY Partitioning-library: User Guide; Version 1.1*. Univ.-GH, FB Mathematik/Informatik, 1996.
48. High Performance Computing, Available at: http://slideplayer.com/slide/11136125/, accessed 23 July, 2018.
49. ParMETIS-Parallel Graph Partitioning and Fill-reducing Matrix Ordering, Available at: http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview, accessed 9 February, 2019.
50. Karypis, George, and Vipin Kumar. "Analysis of multilevel graph partitioning". *Supercomputing'95: Proceedings of the 1995 ACM/IEEE conference on Supercomputing*. IEEE, 1995.
51. Karypis G, Schloegel K, Kumar V. Parmetis. "Parallel graph partitioning and sparse matrix ordering library". Version. 2003 Aug 15; **2**.
52. DFG Algorithm Engineering, Available at: http://www.algorithm-engineering.de / softwareprojects ?view= project&task=show&id=11, accessed 9 November , 2018.
53. SWMATH, DibaP, Available at: http://www.swmath.org/software/8343, accessed 15 November, 2018.
54. Meyerhenke, H., Monien, B., Sauerwald, T., 2009. "A new diffusion-based multilevel algorithm for computing graph partitions". *Journal of Parallel and Distributed Computing*, 2009; **69**(9), p.750-761.
55. Meyerhenke, H., Monien, B., Sauerwald, T., "A new diffusion-based multilevel algorithm for computing graph partitions of very high quality". *In:* IEEE International Symposium on Parallel and Distributed Processing, 2008. IPDPS 2008.,p. 1-13
56. PaToH, Available at: http://www.cs.bilkent.edu.tr/~aykanat/pargrp/patoh/., accessed 21 December, 2018.
57. Catalyurek, U.V. and Aykanat, C., "Hypergraph-partitioning-based decomposition for parallel sparse- matrix vector multiplication". *IEEE Transactions on parallel and distributed systems*, 1999, **10**(7), p.673-693.
58. Çatalyürek, Ümit, and Cevdet Aykanat. Patoh (partitioning tool for hypergraphs). *Encyclopedia of Parallel Computing*, Springer US. 2011; p.1479-1487.
59. SWMATH, PaToH, Available at: http:/ www.swmath.org/?term= PaToH & which _ searc h= standard&sortby= , accessed 18 January, 2019.
60. Hendrickson, Bruce, and Robert Leland. *The Chaco users guide. Version 1.0*. No. SAND-93-2339. Sandia National Labs., Albuquerque, NM (United States), 1993.
61. Leland, B.H.R. and Hendrickson, B. *The Chaco user's guide: version 2.0*. Tech. Rep. SAND94-2692, Sandia National Labs, Albuquerque, NM, 1995.
62. Sandia National Laboratories, Available at: http://www.cs.sandia.gov/CRF/chac_p2.html, accessed 14 February , 2018.
63. Chinthapanti, Pavan K. "*A comparative analysis of graph partitioning tool"s*. Diss. Texas Tech University, 2004.
64. "Chaco: Algorithms and Software for Partitioning Meshes", Available at:http : / / www . cs . sandia . gov / CRF / chac _p2.html, accessed 7 March, 2019
65. Rajamanickam, Sivasankaran, and Erik G. Boman. "An Evaluation of the Zoltan Parallel Graph and Hypergraph Partitioners". No. SAND2011-8646C. Sandia National Lab. (SNL-NM), Albuquerque, NM (United States), 2011.
66. Devine, Karen D., et al. "Getting started with zoltan: A short tutorial." *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2009.
67. Preis, Robert, and Ralf Diekmann. "PARTY-a software library for graph partitioning". *Advances in Computational Mechanics with Parallel and Distributed Processing*, 1997; p 63-71.
68. Preis, Robert. "Linear time 1/2-approximation algorithm for maximum weighted matching in general graphs". *Annual Symposium on Theoretical Aspects of Computer Science*. Springer, Berlin, Heidelberg, 1999; p 259-269.
69. Monien B, Schamberger S. "Graph partitioning with the party library: Helpful-sets in practice". In: 16th IEEE Symposium on Computer Architecture and High Performance Computing, 2004 Oct 27, p. 198-205.
70. Von Luxburg U. "A tutorial on spectral clustering". *Statistics and computing*. 2007 Dec 1; **17**(4), p 395-416.
71. "Software Requirement Specification for Gephi". Available at: https: // gephi.org /users /gephi_ srs_ documen.pdf, accessed 24 April, 2019.
72. Bokhare, A. and Metkewar, P.S. "Typical model–amalgamation of multilevel graph partitioning". *Advances in Theoretical and Applied Mathematics*, 2016; **11**, pp.115-29.
73. Newman ME. "Spectral methods for community detection and graph partitioning". *Physical Review E*. 2013 Oct 30; **88**(4), p.042822.
74. Deveci M, Kaya K, Uçar B, Çatalyürek ÜV. "Hypergraph partitioning for multiple communication cost metrics: Model and methods". *Journal of Parallel and Distributed Computing.* 2015 Mar 1;**77**, p 69-83.
75. Van Laarhoven, Peter JM, and Emile HL Aarts. "Simulated annealing". *Simulated annealing: Theory and applications*. Springer, Dordrecht, 1987; p.7-15.
76. Johnson, D.S., Aragon, C.R., McGeoch, L.A., Schevon, C. "Optimization by simulated annealing: an experimental evaluation; part I, graph partitioning". *Operations research*, 1989; **37**(6), p. 865-892.
77. Martella, C., Logothetis, D., Loukas, A., Siganos, G. "Spinner: Scalable graph partitioning in the cloud". *In:* IEEE 33rd International Conference on Data Engineering (ICDE), 2017, p. 1083- 1094
78. Hillier, F. S., and G. J. Lieberman. "Introduction to operations research, Elizabeth A". *Jones, New York*, 2005;
79. Wu, J., Wang, P., Lam, S.K., Srikanthan, T. "Efficient heuristic and tabu search for hardware/software partitioning". *The Journal of Supercomputing*, 2013; **66**(1), p.118-134.
80. Fishkind, D.E., Sussman, D.L., Tang, M., Vogelstein, J.T., Priebe, C.E. "Consistent adjacency- spectral partitioning for the stochastic block model when the model parameters are unknown". *SIAM Journal on Matrix Analysis and Applications*, 2013; **34**(1), p.23-39.

81. Barnard, S.T., Simon, H.D. "Fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems". *Concurrency and computation: Practice and Experience*, 1994; **6**(2), p.101-117.
82. Kuntz, P., Snyers, D., Layzell, P.. "A stochastic heuristic for visualising graph clusters in a bi-dimensional space prior to partitioning". *Journal of Heuristics*, 1999; **5**(3), p.327-351.
83. Comellas, F.,Sapena, E. "A multiagent algorithm for graph partitioning". *In:* Workshops on Applications of Evolutionary Computation, Springer, Berlin, Heidelberg, April 2006, p. 279-285.
84. Van Den Bout, D.E., Miller, T.K. "Graph partitioning using annealed neural networks". *IEEE Transactions on neural networks*, 1990; **1**(2), p.192-203.
85. Bilbro, G., Mann, R., Miller, T.K., Snyder, W.E., Van den Bout, D.E.,White, M. "Optimization by mean field annealing". In *Advances in neural information processing systems*, 1989; p. 91-98.
86. Connor, A.M., Shea, K. "A comparison of semi-deterministic and stochastic search techniques". *In*: Evolutionary Design and Manufacture, Springer, London, *2000*, pp. 287-298.
87. Kirkpatrick, S. "Optimization by simulated annealing: Quantitative studies". *Journal of statistical physics*, 1984; *34*(5-6), p.975-986.

AUTHORS PROFILE

Anuja Bokhare is working as Assistant Professor in the department of Computer Science at Symbiosis Institute of Computer Studies and Research, Pune, Maharashtra India. She received M.Phil (Computer Science) at Y.C.M.O.U , Nasik, India. She has 15 years of experience in the field of academics. Her research interest includes applications of fuzzy logic, neural network, Software Engineering. She had published 6 research papers in international journal and one book in her account.

Dr Pravin S Metkewar has awarded with "Parbhani Ratna" award for his excellence in academics in 1996. He stood rank first in M.Sc. (computer science) in 1997. He has received his Ph.D. degree from SRT University, Nanded in computer science under the faculty of science in Aug 2005He has 20 years of experience in the field of teaching, R & D and industry. He has worked in IDRBT R&D centre in the capacity of Research Officer for three years. His specialization is in OOAD, Information Systems, Neural networks, Machine learning, Data analytics, Graph Theory, Image Analysis and Fuzzy Logic. He is a CSI life member.He was a member of Academic council, BOS and BOS Sub-committee in SIU, Pune. He has presented and published 94 research papers in International Journal, National Journal and IEEE proceedings and 05 books in his account. He is a research guide of Symbi osis International (Deemed University), Pune.