Information Sciences 369 (2016) 765–787

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

On minimizing vertex bisection using a memetic algorithm

Pallavi Jain, Gur Saran, Kamal Srivastava*

Department of Mathematics, Dayalbagh Educational Institute, Agra, India

ARTICLE INFO

Article history:
Received 1 December 2014
Revised 19 July 2016
Accepted 21 July 2016
Available online 25 July 2016

Keywords:
Vertex bisection minimization
Vertex width
Memetic algorithm

ABSTRACT

Vertex Bisection Minimization problem (VBMP) consists of partitioning a vertex set into two sets B and B' where $|B| = |V|/2$ such that vertex width, *VW*, is minimized which is defined as the number of vertices in B which are adjacent to at least one vertex in B'. It is an NP-complete problem in general. VBMP has applications in fault tolerance and is related to the complexity of sending messages to processors in interconnection networks via vertex disjoint paths. In this paper, a memetic algorithm has been designed for this problem (MAVBMP) in which four construction heuristics have been proposed to generate the initial population. These heuristics are analyzed statistically and accordingly used in proportion to generate the initial population for MAVBMP. A new crossover type search operator has been proposed for recombination and a local improvement operator has also been developed. Extensive experiments have been conducted on several classes of graphs such as complete bipartite graphs, 2-dimensional ordinary meshes, 2-dimensional toroidal meshes, 3-dimensional toroidal meshes, complete split graph, join of hypercubes and Harwell-boeing graphs which are a subset of public domain Matrix Market library. Trends observed in the experimental results for some of the classes of graphs have led to conjectures for vertex width.

1. Introduction

Vertex Bisection Minimization problem (VBMP) consists of partitioning a vertex set of a graph $G = (V, E)$, $|V| = n$, into two sets B and B' where $|B| = \lfloor n/2 \rfloor$ such that vertex width (*VW*) is minimized where vertex width is defined as the number of vertices in B which are adjacent to at least one vertex in B'. Formally, for a partition $P = (B, B')$, its vertex width is $VW(G, P) = |\{u \in B : \exists v \in B' \wedge (u, v) \in E(G)\}|$. VBMP is to find a partition P^* such that $VW(G, P^*) = \min_{\forall partition\ P} VW(G, P)$. In this paper, a partition $P = (B, B')$ is taken to be a solution and $VW(G, P)$ as its *cost*. This problem has been treated as a graph layout problem by Diaz et al. [6]. Vertex Bisection Minimization Problem is relevant to fault tolerance and is related to the complexity of sending messages to processors in interconnection networks via vertex disjoint paths [6]. VBMP is NP-complete for general graphs but is polynomially solvable for trees and hypercubes [3]. Fraire et al. [9] have presented a branch and bound algorithm, Integer Linear Programming formulation and Quadratic Programming formulation for VBMP. Recently Jain et al. [12] have presented new Integer Linear Programming formulation and Quadratic Programming formulations for VBMP which require lesser number of variables and constraints than given by Fraire et al. [9]. Further, a new branch and bound algorithm has also been proposed in [13] which improves on the branch and bound algorithm proposed in [9].

* Corresponding author.
E-mail addresses: pallavijain.t.cms@gmail.com (P. Jain), gursaran@dei.ac.in (G. Saran), kamal.sri@dei.ac.in (K. Srivastava).

http://dx.doi.org/10.1016/j.ins.2016.07.055
0020-0255/

VBMP is related to graph bisection problem which is a special case of graph partitioning problem (GPP). The graph k-partition problem is defined as follows: Given an unweighted graph $G=(V, E)$, partition V into k subsets, $V_1, V_2, \ldots, V_k$ such that $V_i \cap V_j = \emptyset$ for $i \neq j$, $\cup V_i = V$ and $|V_i| \leq (1 + x/100) \times \lceil |V|/k \rceil$ where $x \geq 0$. The objective is to minimize the number of edges of E whose incident vertices belong to different subsets [10]. If $k = 2$ and the partition is balanced ($x = 1$) then the problem is a graph bisection or edge bisection problem. GPP is NP-hard in general and has practical applications in load balancing, VLSI, mesh distributing and several others. Due to its practical applications, a number of heuristics have been proposed in the literature. Kernighan and Lin [16] have proposed an iterative heuristic (*KL*) for this problem. In each iteration, swapping of two vertices of different partitions is performed greedily. If the swapping of vertices reduces the number of cut edges then it is performed otherwise not. Fiduccia and Mattheyses [8] presented a variant of the *KL* procedure that can be implemented more efficiently. For the bisection problem, Johnson et al. [14] have designed a simulated annealing heuristic. Genetic algorithms [5] and Ant Colony optimization techniques [15] have also been proposed for this problem. In 2006, Bichot [2] proposed a metaheuristic based on fusion and fission. Many of the efficient heuristics combine local search with the so-called multilevel approach whose principle is to build a series of graphs of decreasing size obtained by coarsening the input graph recursively [4,19]. A recent approach for GPP is a memetic algorithm (MAGP) by Galinier et al. [10]. They have designed a construction heuristic to generate an initial population. This heuristic starts from an empty k-partitioning. In each iteration, an unassigned vertex is introduced into one of the subsets. Each subset grows using breadth first search approach, and the k subsets are built in parallel. Initially, $3 \times ps$ number of solutions are created using this heuristic where ps denote the population size, but for the initial population ps number of best solutions are considered. In the generational phase, two parent solutions are selected randomly to generate an offspring using crossover operator which then undergoes tabu search operator. Worst solution among two parent solutions is replaced by offspring to update the population. The experimental results obtained by them show that MAGP outperforms other state-of-the-art algorithms.

In this paper, we have proposed a memetic algorithm (MAVBMP) for VBMP which is tested on a large number of test graphs including those for which optimal results are known in the literature. MAVBMP achieves the known optimal values in all such cases. Despite practical applications of VBMP, no heuristics have been proposed for this problem per se, for general graphs, in the literature surveyed so far. However, as discussed above, there is a wide body of literature for GPP which could possibly be used for VBMP with adequate interpretation. In order to examine this, we have compared our approach for VBMP with a modified MAGP for VBMP. The modification incorporates a necessary adaptation of MAGP for the vertex bisection problem. We refer to this adaptation of MAGP as AMAGP further in this paper.

1.1. Contributions of this work

In this section we outline the salient features of MAVBMP. In order to generate a good initial population, we have designed four construction heuristics which attempt to place adjacent vertices in the same partition in a way that they do not contribute to the vertex width. Experiments are carried out to compare the performance of these construction heuristics as well as another heuristic used in MAGP [10]. The results of the experiments have been analyzed statistically. Based on the results, we have selected three heuristics, other than the one used in MAGP, for generating initial population in MAVBMP in different proportions. It is interesting to note that these heuristics have been able to achieve optimal results for some of the input instances. A new crossover type search operator has been designed for exploration of the search space. A problem dependent local improvement operator has also been designed and deployed in the generational phase of the memetic algorithm.

Experiments were carried out to study the impact of memetic operators (local search and crossover) and also for tuning of parameters such as population size, crossover rate etc. Extensive experiments have also been carried out to assess the performance of the proposed algorithm on large number of graphs belonging to different classes. Optimal results for hypercubes are known in the literature [3] and MAVBMP simulations on these graphs are able to achieve these results. MAVBMP is also executed on some classes of graphs for which optimal results are not known such as complete bipartite graph, 2-dimensional meshes, 2-dimensional toroidal meshes, 3-dimenssional toroidal meshes, complete split graph and join of hypercubes. The experiments have led to conjectures on the vertex width of 2-dimensional toroidal meshes and join of hypercubes. The test suite also includes a set of *Small* graphs and Harwell-Boeing graphs which are a subset of the public domain Matrix Market library (http://www.optsicom.es/cutwidth). Experiments are also performed to compare MAVBMP with one of its variants in which the initial population is generated randomly (MAVBMPR). We have also experimentally compared the performance of the AMAGP and MAVBMP. It is noticed that MAVBMP runs significantly faster (Section 5.7).

1.2. Organization

The rest of the paper is organized as follows. Section 2 describes MAVBMP. In Section 3 proposed heuristics for generating initial population are described. Implementation details of MAVBMP are presented in Section 4. Section 5 describes the experiments and their results. This is followed by conclusions in Section 6.