Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/080116

Vertex Ordering, Clustering, and Their Application to Graph Partitioning

Yourim Yoon[1] *and Yong-Hyuk Kim*[2,*]

[1] Future IT R & D Lab., LG Electronics, Umyeon R & D Campus, 38, Baumoe-ro, Secho-gu, Seoul 137-724, Korea
[2] Department of Computer Science & Engineering, Kwangwoon University, 20 Kwangwoon-ro, Nowon-gu, Seoul 139-701, Korea

Received: 2 Jun. 2013, Revised: 16 Sep. 2013, Accepted: 17 Sep. 2013
Published online: 1 Jan. 2014

Abstract: We propose a new heuristic for vertex ordering and a method that splits the vertex ordering into clusters. We apply them to the graph partitioning problem. The application of these ideas incorporates reordering in genetic algorithms and the identification of clustered structures in graphs. Experimental tests on benchmark graphs showed that the new vertex-ordering scheme performed better than existing methods in terms of genetic algorithms, and that the clusters were successfully captured.

Keywords: graph algorithms, design of algorithms, graph partitioning, vertex ordering, graph clustering, genetic algorithm.

1 Introduction

Assume that $G = (V,E)$ is an undirected unweighted graph, where V and E are the sets of n vertices and e edges, respectively. A bipartition $\{C_1,C_2\}$ of G satisfying $C_1,C_2 \subset V$, $C_1 \cup C_2 = V$, $C_1 \cap C_2 = \phi$, and $||C_1| - |C_2|| \leq 1$ is called a bisection. The cut size of a bipartition $\{C_1,C_2\}$ is the cardinality of the edge set $|\{(v,w) \in E : v \in C_1, w \in C_2\}|$. The graph bisection problem is to find a bisection with the minimum cut size.

Genetic algorithms (GAs) are a kind of bio-inspired meta-heuristics, and they have been applied to various optimization problems [13,16]. They have also been successfully used to solve the graph partitioning problem [3,5,8,9,10,11,12]. Kim *et al.* [6] presents a deep survey of genetic approaches for graph partitioning.

In GAs for solving graph bisection, each individual is usually represented by an n-bit binary linear string. Each individual is called a chromosome; i.e., a chromosome corresponds to a bisection of the given graph. Each gene corresponds to a vertex in the given graph. It has the value zero if the corresponding vertex belongs to C_1 and has the value one otherwise.

A GA for this problem evolves a group of individuals (n-bit linear strings) under a genetic process. A schema is a pattern of bit strings that can be described by a template consisting of ones, zeros, and asterisks; here, ones and zeros represent the pattern and the asterisks represent "don't care." GAs start with a set of randomly-generated initial individuals. Of course, the qualities of the individuals are quite low in the early stages of the GA. However, most low-quality individuals include some schemata common to high-quality individuals. The crossover operators of GAs generate larger schemata by juxta-position of smaller schemata. It is crucial to preserve valuable schemata. However, they are easy to be destroyed by crossovers if the positions related to the schema are scattered.

Vertex ordering in GAs is used as a schema preprocessing technique in order to preserve perceived valuable schemata. Some studies on vertex ordering have been conducted [1]. There have also been several studies for gene rearrangement in GAs. They reported superior results to GAs without gene reordering. Bui and Moon [3] first used BFS ordering as a technique of schema preprocessing in GAs. Kim *et al.* [7] presented problem-independent gene reordering using a constructed gene interaction graph. Hwang *et al.* [5] significantly improved the results of Bui and Moon [3]. In this paper, we propose a novel vertex ordering that detects the clustered structure of graphs. We use this vertex ordering

* Corresponding author e-mail: yhdfly@kw.ac.kr

as a schema preprocessing technique and apply it to the identification of clusters in graphs.

2 Vertex Ordering

If the set of vertices $V = \{v_1, v_2, \ldots, v_n\}$ is given, a vertex ordering $\{v_{\sigma(1)}, v_{\sigma(2)}, \ldots, v_{\sigma(n)}\}$ is represented by the permutation map $\sigma : \{1,2,\ldots,n\} \rightarrow \{1,2,\ldots,n\}$. If $\sigma(j) = i$, vertex v_j is the i-th vertex in the vertex ordering. The objective of a vertex ordering is to preserve the clustered structure of the graph as well as possible.

BFS ordering conducts a breadth-first search (BFS) on the given graph starting at a randomly-chosen vertex. The visiting order of vertices in which the vertices are visited by the BFS is used to order the vertices of the given graph. Max-Adjacency ordering starts at a randomly-chosen vertex, and it iteratively adds the vertex v_r with the most edges incident to previous added vertices to the ordering.

Figure 1 shows our ordering method. As in previous approaches [1], our method starts at randomly-chosen vertices, and it iteratively adds a new vertex to the ordering based on already added vertices. The main difference is that the already added vertices are classified. Two classes (A and B) are used. One (A) and the other (B) are the former-half added vertices and the latter-half ones, respectively. Balancing two classes, un-added vertices are added to the ordering iteratively. Assume $E(v,S)$ is the number of vertices adjacent to the vertex v in a vertex set S. The vertex v_r with the maximal value of $E(v_r,A) - E(v_r,B)$ is added to the former-half order and the set A. Analogously, the vertex v_r with the minimal value of $E(v_r,A) - E(v_r,B)$ is added to the latter-half order and the set B. In the following, we denote our vertex-ordering heuristic by Bi-Constructive ordering. When we use a bucket data structure as in FM [4], our vertex-ordering heuristic takes $\Theta(n+e)$ time.

Choose two different random numbers $a, b \in \{1,2,\ldots,n\}$;
$\sigma(1) \leftarrow a, \sigma(n/2+1) \leftarrow b$;
$A \leftarrow \{v_a\}, B \leftarrow \{v_b\}$;
for $i \leftarrow 2$ to $n/2$ {
  Find an unordered vertex $v_r \in V - (A \cup B)$
    such that $E(v_r,A) - E(v_r,B)$ is maximal;
  $\sigma(i) \leftarrow r$;
  $A \leftarrow A \cup \{v_r\}$;
  Find an unordered vertex $v_r \in V - (A \cup B)$
    such that $E(v_r,A) - E(v_r,B)$ is minimal;
  $\sigma(n/2+i) \leftarrow r$;
  $B \leftarrow B \cup \{v_r\}$;
}
Order the vertices using the permutation map σ;

Fig. 1: Bi-constructive vertex ordering

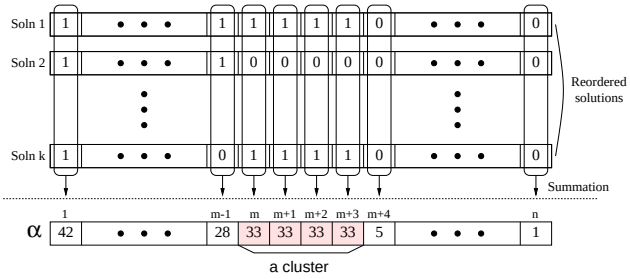

Fig. 2: Clustering derived from vertex ordering

3 Clustering

Clustering [14,16] of graphs has been used to reduce the search space of graph partitioning instances. For example, clustering improves Fiduccia-Mattheyses (FM) bisection [4] through two-phase methodology [2]. In this section, we describe our clustering method.

In graph bisection, we represent each solution by an n-bit linear code. Each bit corresponds to a vertex in the graph, and it has the value 0 or 1. Our clustering method is as follows. After ordering, we get a number of local optimum solutions and get an n-integer linear code through bit-wise summation of them. If two positions have the same integer value in the linear code, we regard it as a strong symptom of cluster. We pull out a contiguous subset of vertices with the same integer value in the linear code. In our experiments, we used fifty local optimum solutions. Figure 2 illustrates an example case of a cluster in the linear code. In the figure, a cluster is composed of a reordered vertex subset $\{v_{\sigma(m)}, v_{\sigma(m+1)}, v_{\sigma(m+2)}, v_{\sigma(m+3)}\}$.

4 Experimental Results

We made experiments on the six graphs that have been used in many studies [3,5,8,9,10,11,12,15]. The different classes of graphs are briefly described below.

- G$n.d$: Random graph with n vertices, in which an edge is made between any two vertices with probability p independently. The probability p is selected so that the expected vertex degree is $d = p(n-1)$.
- U$n.d$: Random geometric graph with n vertices. Vertices lie in the unit square $[0,1] \times [0,1]$ and their coordinates are uniformly selected from the unit interval $[0,1]$. An edge is made between two vertices if their Euclidean distance is less than or equal to t, where the expected vertex degree is $d = \pi n t^2$.
- cat.n: Caterpillar graph with n vertices. Each vertex has six legs. Starting with a straight line, each vertex has degree 2 except the outermost ones. Then, each vertex on the straight line is connected to six new

ones called the legs of the caterpillar. rcat.n means caterpillar graph with n vertices, in which each vertex on the straight line has $\sqrt{n}$ legs. The optimal cut size of two classes of caterpillar graphs becomes 1.

Table 1 shows the performance of genetic bisection algorithms (GBA) preprocessed by BFS [3], Max-Adjacency, and Bi-Constructive vertex-ordering. The column "None" shows the performance of GBA without schema preprocessing. GBA has the same framework as the GA of [8,10]. All the algorithms expended a comparable amount of time; however, the Bi-Constructive vertex-ordering algorithm considerably outperformed the others.

Table 1: Comparison of Schema Preprocessing Methods in a GA

Graph	None†	BFS [3]†	Max-Adj‡	Bi-Cons‡
G1000.05	452.75	453.00	453.63	**452.16**
G1000.20	3386.21	3386.05	3386.27	**3384.63**
U2000.05	43.85	8.89	9.01	**4.38**
U5000.10	319.37	140.85	129.40	**80.33**
cat.5252	119.32	2.90	3.00	**2.08**
rcat.5114	21.03	2.75	2.92	**2.20**

† Data from [8].

‡ Average over 100 runs.

The next experiment examines the effect of clustering in a hierarchical partitioning. Here we chose the two-phase methodology in the FM algorithm as a sample case. The FM method [4] is considered as a representative traditional iterative improvement partitioning method, and the two-phase approach [2] would be a traditional paradigm for hierarchical partitioning. In general, the two-phase FM method is processed as follows. FM is first run on the clustered graph, and the resultant bipartition is a starting point of a second FM run on the given graph which is the unclustered graph. Table 2 shows the performance. Two-phase FM significantly outperformed FM. The experimental results showed that the proposed clustering algorithm effectively detects the clustered structure.

Table 2: Effect of Clustering in a Hierarchical Partitioning

Graph	FM	Two-Phase FM
G1000.05	501.64	**495.04**
G1000.20	3482.53	**3471.01**
U2000.05	160.95	**12.31**
U5000.10	961.25	**248.19**
cat.5252	252.44	**6.69**
rcat.5114	187.59	**6.65**

Average over 1,000 runs.

5 Concluding Remarks

We described a bi-constructive vertex-ordering method for graph partitioning and explored its application to clustering. We improved the Max-Adjacency vertex-ordering by considering the attraction power of the partitions and captured the clusters through the common properties of local optimum solutions. It would be interesting to extend it to the vertex-ordering method with general k classes and find an optimal k. More efficient utilization of clusters and application to the multi-way partitioning problem are left for future study.

Acknowledgment

The present research has been conducted by the Research Grant of Kwangwoon University in 2014. This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2012-0001855). This work was also partly supported by the Advanced Research on Meteorological Sciences through the National Institute of Meteorological Research of Korea in 2013 (NIMR-2012-B-1).

References

[1] C. Alpert and A. B. Kahng. A general framework for vertex orderings, with applications to netlist clustering. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 63–67, (1994).

[2] T. N. Bui, C. Heigham, C. Jones, and T. Leighton. Improving the performance of the kernighan-lin and simulated annealing graph bisection algorithms. In *Proceedings of the 26th ACM/IEEE Design Automation Conference*, pages 775–778, (1989).

[3] T. N. Bui and B. R. Moon. Genetic algorithm and graph partitioning. *IEEE Transactions on Computers*, **45**(7):841–855, (1996).

[4] C. Fiduccia and R. Mattheyses. A linear time heuristics for improving network partitions. In *Proceedings of the 19th ACM/IEEE Design Automation Conference*, pages 175–181, (1982).

[5] I. Hwang, Y.-H. Kim, and B.-R. Moon. Multi-attractor gene reordering for graph bisection. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1209–1215, (2006).

[6] J. Kim, I. Hwang, Y.-H. Kim, and B.-R. Moon. Genetic approaches for graph partitioning: A survey. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 473–480, (2011).

[7] Y.-H. Kim, Y.-K. Kwon, and B.-R. Moon. Problem-independent schema synthesis for genetic algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference - Lecture Notes in Computer Science*, volume **2723**, pages 1112–1122, (2003).

[8] Y.-H. Kim and B.-R. Moon. A hybrid genetic search for graph partitioning based on lock gain. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 167–174, (2000).

[9] Y.-H. Kim and B.-R. Moon. Investigation of the fitness landscapes and multi-parent crossover for graph bipartitioning. In *Proceedings of the Genetic and Evolutionary Computation Conference - Lecture Notes in Computer Science*, volume **2723**, pages 1123–1135, (2003).
[10] Y.-H. Kim and B.-R. Moon. Lock-gain based graph partitioning. *Journal of Heuristics*, **10**(1):37–57, (2004).
[11] Y.-H. Kim, Y. Yoon, A. Moraglio, and B.-R. Moon. Geometric crossover for multiway graph partitioning. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1217–1224, (2006).
[12] A. Moraglio, Y.-H. Kim, Y. Yoon, and B.-R. Moon. Geometric crossover for multiway graph partitioning. *Evolutionary Computation*, **15**(4):445–474, (2007).
[13] M. Thangamani and P. Thangaraj. Fuzzy ontology for distributed document clustering based on genetic algorithm. *Applied Mathematics & Information Sciences*, **7**(4):1563–1574, (2013).
[14] M. Wang, X. Shang, X. Li, Z. Li, and W. Liu. Efficient mining of differential co-expression constant row bicluster in real-valued gene expression datasets. *Applied Mathematics & Information Sciences*, **7**(2):587–598, (2013).
[15] Y. Yoon and Y.-H. Kim. New bucket managements in iterative improvement partitioning algorithms. *Applied Mathematics & Information Sciences*, **7**(2):529–532, (2013).
[16] L. Zhang, X. Zou, and Z. Su. Ga optimization model for time/cost trade-off problem in repetitive projects considering resource continuity. *Applied Mathematics & Information Sciences*, **7**(2):611–617, (2013).

Yourim Yoon received the B.E. degree in computer engineering and the Ph.D. degree in computer science and engineering from Seoul National University, Seoul, Korea, in 2003 and 2012, respectively. Since March 2012, she has been a senior research engineer at Future IT R&D Lab. in LG Electronics, Seoul, Korea. Her research interests include optimization theory, machine learning, combinatorial optimization, evolutionary computation, discrete mathematics, operations research, smart grid, and sensor networks. She served as a reviewer for BIC-TA 2007, BMIC 2011, IEEE TEVC, and TIIS.

Yong-Hyuk Kim received the B.S. degree in computer science and the M.S. and Ph.D. degrees in computer science and engineering from Seoul National University (SNU), Seoul, Korea, in 1999, 2001, and 2005, respectively. From March 2005 to February 2007, he was a Postdoctoral Scholar in SNU and also a research staff member at the Inter-University Semiconductor Research Center in SNU. Since March 2007, he has been a professor at Department of Computer Science and Engineering in Kwangwoon University, Seoul, Korea. His research interests include algorithm design/analysis, discrete mathematics, optimization theory, combinatorial optimization, evolutionary computation, operations research, and data/web mining. Dr. Kim has served as an Editor of TIIS journal in 2010-2013, a Committee Member of GECCO 2005-2006,2013 & IEEE CEC 2009-2011, and a reviewer for journals (CIM, IS, TC, TEVC, TKDE, TPDS, TSE, TVT) of IEEE since 2003. He is a member of the IEEE SMC Society and the IEEE Communications Society.