Journal of Graph Algorithms and Applications
http://www.cs.brown.edu/publications/jgaa/
vol. 4, no. 2, pp. 1–11 (2000)

Approximation algorithms for some graph partitioning problems

G. He
Battelle, Pacific Northwest National Lab
P. O. Box 999 / MS K9-55, Richland, WA 99352 USA
http://www.pnl.gov/remote/expertise/ggh.htm
George.He@pnl.gov

J. Liu[1]
Department of Mathematics and Computer Sciences
University of Lethbridge, Lethbridge, Alberta, Canada, T1K 3M4
http://www.cs.uleth.ca/
liu@cs.uleth.ca

C. Zhao
Department of Mathematics and Computer Sciences
Indiana State University, Terre Haute, IN 47809 USA
http://math.indstate.edu/zhao.html
cheng@laurel.indstate.edu

Abstract

This paper considers problems of the following type: given an edge-weighted k-colored input graph with maximum color class size c, find a minimum or maximum c-way cut such that each color class is totally partitioned. Equivalently, given a weighted complete k-partite graph, cover its vertices with a minimum number of disjoint cliques in such a way that the total weight of the cliques is maximized or minimized. Our study was motivated by some work called the index domain alignment problem [6], which shows its relevance to optimization of distributed computation. Solutions of these problems also have applications in logistics [3] and manufacturing systems [10]. In this paper, we design some approximation algorithms by extending the matching algorithms to these problems. Both theoretical and experimental results show that the algorithms we designed produce good approximations.

Communicated by D. Eppstein.
Submitted: July 2000. Revised: August 2000.

[1] Research by this author was partially supported by the Natural Sciences and Engineering Research Council of Canada.

1 Introduction

Distributed memory architectures are becoming increasingly popular since the promised scalability at reduced costs, and the availability of high performance microprocessors. This architecture requires that data associated with a given computation be partitioned and distributed to the local storage of each individual processor. How this is done will affect the program performance. However, programming of distributed memory multiprocessors is difficult and error-prone due to the lack of a single uniform global address space. The recent research (Li and Chen [6], Banerjee, Eigenmann, Nicolau and Padua [1], Gupta and Banerjee [4], Knoble, Lukas and Steele [5], and Ramanujam and Sadayappan [9]) has concentrated on automating this process. In [6], Li and Chen modeled this process as the (primary) index domain alignment problem and proved that the problem is NP-complete for the class of graphs with alignment dimension 2.

In [7], we have further generalized the index domain alignment problem into the following four graph partitioning problems, and proved that these problems are polynomially equivalent and NP-complete. We follow the standard definitions and notations in [2].

Let G be a graph. An edge weight on G is a map w from $E(G)$ to $Z^+ \cup \{0\}$, the set of nonnegative integers. If H is a subgraph of G, we denote $w(H) = \sum_{e \in E(H)} w(e)$.

Let $G = (V(G), E(G), w)$ be an edge weighted k-vertex colorable graph with a list of vertex color classes $U_1, \ldots, U_k$, where U_i is the set of vertices with color i. Let $c = max\{|U_1|, |U_2|, \cdots, |U_k|\}$. G is called a *c-way k-colored graph* with color classes $U_1, \ldots, U_k$. If $|U_i| = c$ for $1 \leq i \leq k$, we say that G is a *c-way k-colored balanced graph*.

Let $G = (V(G), E(G), w)$ be a c-way k-colored graph with color classes $U_1, \ldots, U_k$. An *orthogonal partition* of G is a vertex partition of $V(G)$ into $V_1, \ldots, V_c$ such that $|V_i \cap U_j| \leq 1$ for any U_i and V_j. That is, each U_i contains vertices of different colors.

The maximum (minimum) orthogonal partition problem

Let $G = (V(G), E(G), w)$ be an edge weighted, c-way k-colored graph with color classes $U_1, \ldots, U_k$. Find an orthogonal partition $V_1, \ldots, V_c$ of G so that

$$\sum_{i \neq j}^{c} \sum_{e \in E([V_i, V_j])} w(e)$$

is the maximum (minimum) among all the orthogonal partitions of G.

We call such a partition $V_1, \ldots, V_c$ an *optimum partition* of G. Let S be the spanning subgraph of G such that e is an edge of S if and only if e is an edge joining V_i and V_j for some $i \neq j$. S is called a maximum (minimum) *orthogonal subgraph* of G for the maximum (minimum) orthogonal partition problem of G.

Note that a k-vertex colorable graph is a k-partite graph. We have the following variation of the above problems, which are easier to deal with.

The maximum (minimum) disjoint k-clique problem

Let $K_{c,\ldots,c}$ be an edge weighted complete k-partite graph with c vertices in each part. Find a set of c vertex-disjoint k-cliques of $K_{c,\ldots,c}$ that has the maximum (minimum) weight.

We would like to point out that both the maximum and minimum vertex disjoint $k(\geq 2)$-clique problems have applications in logistics (see [3]) and manufacturing systems in industry (see [10]). The logistics application arises in the context of making weekly assignments of sets of drivers to loads. Often we are left with a driver assignment problem that is an instance of c vertex disjoint 3-cliques: we are given an edge weighted complete 3-partite graph with a vertex set V_1 of drivers, a vertex set V_2 of beginning-of-the-week loads and a vertex set V_3 of midweek loads. We are looking for a set of vertex-disjoint triangles, say $\{\Delta_{uvw}\}$ so we can maximize the total revenue or minimize the total cost. The manufacturing system application occurs in the selection of quality tools and assignment of the tools and quality operations to machine centers or inspection stations. This case can also be modeled as an instance of c vertex disjoint 3 (or higher)-cliques.

Since the maximum (minimum) orthogonal partition problem and the maximum (minimum) clique problem are all NP-complete, it is worthwhile to find approximation algorithms for these problems, and this is the goal of our paper.

In Section 2, we extend the optimum matching algorithms to these problems to obtain approximation algorithms, and analyze these algorithms. Section 3 provides experimental results and comparisons with other algorithms, for example, the random search algorithm. Section 4 shows that in some sense these approximations are tight. Section 5 gives concluding remarks.

2 Approximation algorithms

First, we define a useful graph *merge* operation.

Let $G = (V(G), E(G), w)$ be a c-way k-colored graph with color classes $U_1, \ldots, U_k$. Consider the induced subgraph $G_{i,j} = G[U_i \cup U_j]$. Let $F_{i,j} = \{u_1v_1, u_2v_2, \ldots, u_lv_l\}$ be a matching of $G_{i,j}$ where $u_k \in U_i$ and $v_k \in U_j$. Merge U_j with U_i by identifying the vertex v_k with u_k, also identifying an unsaturated vertex v under $F_{i,j}$ in U_i with an unsaturated vertex u under $F_{i,j}$ in U_j in an obvious way, and leaving the vertices which are not $F_{i,j}$-saturated and cannot be paired off unchanged. The new vertex set is denoted by $U_{i(j)}$. Remove the loops and edges within $U_{i(j)}$, and replace multiple edges by a single edge with the sum of weights of the multiple edges as the new weight of the edge. The new weighted graph $(G_{i(j)}, w')$ is called the *merge of U_j to U_i from G along $F_{i,j}$*. We note that $(G_{i(j)}, w')$ is an edge weighted $(k-1)$-colored graph.

Next, we extend G to a complete k-partite balanced graph $C(G)$ with c vertices in each part by adding in some new vertices and edges. Let w' be the extension of w to $C(G)$ such that $w'(e) = 0$ if e is a new edge. Note that

$C(G) = G$ if G is a complete k-colored balanced graph with c vertices in each part.

Now we consider some approximate solutions based on the idea of optimum matching algorithms.

Algorithm A1. (For the maximum orthogonal partition and the minimum k-clique problems)

Begin
1 Input G with color classes $U_1, \ldots, U_k$.
2 If $C(G) \neq G$, construct $C(G)$ with columns $U'_1, \ldots U'_k$;
label the new vertices with symbol $'$;
assign the new edges weight 0 to obtain the weight function w'
3 Set $i = k$, and $G^i = C(G)$.
4 Repeat
Construct $G^i_{1,i} = [U'_1, U'_i]$.
Find a minimum weighted 1-factor F_i of $G^i_{1,i}$.
Relabel the vertices in U'_i so that $F_i = \{v_1 u^i_1, v_2 u^i_2, \ldots, v_c u^i_c\}$.
Construct a merge $G^i_{1(i)}$ from G^i along with F_i.
Set i to $i-1$, $G^i = G^i_{1(i)}$.
Until $i = 2$.
5 Find a minimum weighted 1-factor F_2 of G^2.
6 Set $V'_1 = \{v_1, u^2_1, u^3_1, \ldots, u^k_1\}, \ldots, V'_c = \{v_c, u^2_c, u^3_c, \ldots, u^k_c\}$.
7 Set $V_i = V'_i - \{$ vertices with label $'\}$.
8 Output $K = G[V_1] \cup \cdots \cup G[V_c]$ and $X = G \backslash E(K)$.
End.

Theorem 1 *Let G be an edge weighted complete k-colored graph with c vertices in each part. Algorithm A1 computes in $O(kc^3)$ time an approximation solution K for the minimum disjoint k-clique problem satisfying*

$$w(K) \leq \frac{1}{c} w(G).$$

If G is an edge weighted c-way k-colored graph, then the subgraph X constructed by Algorithm A1 is an orthogonal subgraph for the maximum partition problem with $w(X) \geq \frac{c-1}{c} w(G)$.

Proof: For the clique problem, $G = C(G)$ and $U_i = U'_i$. We prove the theorem by induction on k. If $k = 2$, G is a complete bipartite graph $K_{c,c}$, $E(G)$ has a decomposition into c edge-disjoint 1-factors. Therefore, Algorithm A1 produces an optimum 1-factor with the minimum weight. Being an optimum one, we must have $w(X) \leq \frac{1}{n} w(G)$. The theorem is true for $k = 2$.

Suppose that the theorem is true for any weighted complete $(k-1)$-partite graphs with c vertices in each part. Now let $k > 2$. We are going to prove that the theorem is true for k.

Let G be a weighted complete k-partite graph with n vertices in each part. In Algorithm A1, first at step 4, we consider the subgraph $G_{1,k} = [U_1, U_k]$.

Then Algorithm A1 delivers a minimum weighted 1-factor F_k of $G_{1,k}$. We have that $w(F_k) \le \frac{1}{c} w(G_{1,k})$. Now we let G' be a merge graph $(G_{1(k)}, w')$ from G along the 1-factor F_m. Let K' be the result obtained by applying Algorithm A1 to G'. Then $w'(K') \le \frac{1}{c} w'(G')$ by the induction hypothesis. We note that K' consists of c disjoint $(k-1)$-cliques. At step 6, we construct K^* from K' as follows: if a clique C in K' contains the vertex u_i, then we include v_i to C to obtain a k-clique. It is clear that K^* is a feasible solution to G for the minimum clique problem. Then we have

$$\begin{aligned} w(K) = w(K^*) &= w'(K') + w(F_k) \\ &\le \frac{1}{c} w'(G') + w(F_k) \\ &= \frac{1}{c}\Big(\sum_{e \notin [U_1, U_k]} w(e)\Big) + w(F_k) \\ &\le \frac{1}{c}\Big(\sum_{e \notin [U_1, U_k]} w(e)\Big) + \frac{1}{c}\Big(\sum_{e \in [U_1, U_k]} w(e)\Big) \\ &= \frac{1}{c} w(G). \end{aligned}$$

For the second part of the theorem, we have that $w(X) = w'(C(G)) - w(K) = w(G) - w(K) \ge w(G) - \frac{1}{c} w(G) = \frac{c-1}{c} w(G)$.

As for the running time, we see that the main operation in Algorithm A1 is finding a minimum weighted 1-factor which takes $O(c^3)$ steps (see [2] and [8]). There are $k-1$ such operations, and other operations can be performed in $O(c^3)$ steps, hence the above algorithm has time complexity $O(kc^3)$.

This completes the proof. ∎

Algorithm A2. (For the minimum orthogonal partition and the maximum clique problems)

Begin
1 Input G with color classes $U_1, \ldots, U_k$, where $|U_1| = c$.
2 Set $i = k$, and $G^i = G$.
3 Repeat until $i = 2$.
  Construct $G^i_{1,i} = [U_1, U_i]$.
  Find a maximum weighted 1-factor F_i of ${G^i}_{1,i}$.
  Relabel the vertices in U_i so that $F_i = \{v_1 u^i_1, \ldots, v_l u^i_l\}$.
  Construct a merge $G^i_{1(i)}$ from G^i along with F_i.
  Set i to $i-1$, $G^i = G^i_{1(i)}$.
4 Find a maximum weighted 1-factor F_2 of G^2.
5 Let $V_1 = \{v_1, u^2_1, u^3_1, \ldots, u^{k_1}_1\}, \ldots, V_c = \{v_c, u^2_c, u^3_c, \ldots, u^{k_c}_c\}$.
  Output $K = G[V_1] \cup \cdots \cup G[V_c]$ and $X = G - E(K)$.
End.

Similarly, we can prove the following:

Theorem 2 *Let G be an edge weighted complete k-partite graph with c vertices in each part. Algorithm A2 computes in $O(kc^3)$ time an approximation solution K for the maximum disjoint k-clique problem satisfying*

$$w(K) \geq \frac{1}{c} w(G).$$

If G is a c-way k-colored edge weighted graph, then the subgraph X constructed by Algorithm A2 is an orthogonal solution for the minimum orthogonal partition problem with $w(X) \leq \frac{c-1}{c} w(G)$.

We say that an algorithm A is a δ-*approximation algorithm* for a problem P if there is a number δ such that for every instance I of P, the approximate solution $S_A(I)$ given by A is related to the exact solution on $S(I)$ by

$$|\frac{S_A(I) - S(I)}{S(I)}| \leq \delta.$$

It is desirable to design approximation algorithm with a small δ.

Theorem 3 *Algorithm A1 is a $\frac{1}{c}$-approximation algorithm for the maximum orthogonal partition problem.*

Proof: Given a c-way k-colored edge weighted graph G, let X^* be an optimum solution and X be a solution obtained by Algorithm A1 for the maximum orthogonal partition problem. Then $w(X^*) \geq w(X) \geq \frac{c-1}{c} w(G)$. Hence $\frac{w(X)}{w(X^*)} \geq \frac{w(X)}{w(G)} \geq \frac{c-1}{c}$, and $0 \leq 1 - \frac{w(X)}{w(X^*)} \leq 1 - \frac{c-1}{c} = \frac{1}{c}$. ■

Remark 4 *We note that Algorithm A1 is significant when c is large.*

Similarly, we can prove the following.

Theorem 5 *Algorithm A2 is a $\frac{c-1}{c}$-approximation algorithm for the maximum k-clique problem.*

Remark 6 *We point out that if G is a complete k-partite graph with c vertices in each part and with constant weight for each edge, then both of our algorithms produce optimum solutions. One natural question is, for which input graphs, will Algorithms A1 and A2 generate optimum solutions?*

To investigate the above question, let G be an edge weighted c-way k-colored graph with color classes $\{U_i : i = 1, \ldots, k\}$. We construct a new graph $T(G)$ from G with vertex set $\{U_i : i = 1, \ldots, k\}$ and $U_i U_j$ is an edge of $T(G)$ if and only if there is an edge in G which joins U_i and U_j. In other words, $T(G)$ is obtained from G by shrinking each of U_i into one vertex and deleting the multiple edges.

We easily have the following observation.

Theorem 7 *Let $G = (V, E, w)$ be a c-way k-colored edge weighted graph with color classes $U_1, \ldots, U_k$. If $T(G)$ is acyclic, then we can obtain optimum solutions for the maximum and minimum orthogonal partition problems using Algorithms A1 and A2.*

Proof: We only prove the case of minimum orthogonal partition. We note that if $T(G)$ is acyclic, then in each merge operation of Algorithm A2, there are no multiple edges. The matching produced in step 4 in Algorithm A2 is actually a maximum matching between U_i and U_{i+1} in G for some i. Let X be a solution produced by Algorithm A2 with the orthogonal partition $V_1, V_2, \ldots, V_c$ and let X^* be an optimum solution to the problem. Then $K = C(G) - E(X)$ is the solution for the minimum clique problem. Let $K^* = C(G) - E(X^*)$. Then $K^* \cap [U_i, U_{i+1}]$ is a matching, hence $w(K^* \cap [U_i, U_{i+1}]) \leq w(K \cap [U_i, U_{i+1}])$. Therefore, $w(K^*) = \sum_i w(K^* \cap [U_i, U_{i+1}]) \leq \sum_i w(K \cap [U_i, U_{i+1}]) = w(K)$. It follows that $w(X) = w(G) - w(K) \leq w(G) - w(K^*) = w(X^*)$. Therefore, $w(X)$ is an optimum solution. ■

3 Experimental Comparison

In this section, we compare Algorithms A1 and A2 with some simple algorithms when k is small. We first describe two simple algorithms for the four partition problems when $k = 3$. The input consists of an edge weighted graph G and a list of color classes U_1, U_2, U_3.

Random Selection Algorithm

This procedure randomly picks three vertices v_1, v_2 and v_3 from U_1, U_2 and U_3 respectively to form a triangle (3-clique). Remove this triangle and repeat this procedure until a set of c vertex-disjoint 3-cliques is found.

The Cubic (3-clique) Greedy Algorithm

This algorithm searches for an optimum (maximum or minimum) weighted triangle containing a starting vertex $v_1 \in U_1$. If v_1, v_2, v_3 are the vertices of this triangle, we remove them from the sets: U_1 , U_2 and U_3, and apply the algorithm again.

Our comparison results are based on 1000 tests of graphs with 30 vertices in each set and with integer weights uniformly in the range from 0 to 9. The tables below give the mean values and the standard deviations of the 1000 approximation solutions obtained by the algorithms.

Method	Random	Greedy	Algorithm A1
Mean	405	218	60
Standard Deviation	26	18	6.6

Comparison results for minimum 3-cliques

Method	Random	Greedy	Algorithm A2
Mean	405	736	749
Standard Deviation	26	8.8	6.6

Comparison results for maximum 3-cliques

We see that Cubic Greedy Algorithms are better than the Random select Algorithm and Algorithms A1 and A2 are better than Cubic Greedy Algorithms.

In the following we will compare only Algorithm A2 with the (4-clique) Greedy Algorithm when $k = 4$. The results are also based on 1000 trials of graphs with 100 vertices in each set and with integer weights uniformly in the range from 1 to 100.

Method	Greedy	Algorithm A2
Mean	41806	52801
Standard Deviation	391	173

Comparison results for maximum 4-cliques

To summarize, Algorithms A1 and A2 appear to be the better performers both in terms of speed and quality of the solution. We are interested in studying how often these methods find an optimum solution. By looking at some smaller instances G of the problem (where we could find an optimum solution by exhaustive searching) with $k = 3$, $c = 5$ (k is the number of sets and c is the number of vertices in each set) and uniform weight, the matching method found an optimum solution about 40% of the time and the cubic greedy method found it about 30% of the time.

4 How good are these algorithms?

We consider the following decision problem.

Problem *Let W be any positive integer and G an edge weighted c-way k-colored graph with color classes $U_1, \ldots, U_k$. Is there an orthogonal subgraph X of G such that $w(X) \geq W$ ($w(X) \leq W$)?*

We note that these two problems are actually other versions of the optimum orthogonal partition problems. Therefore, they are NP-complete. We recall that our algorithms A1 (A2) can produce an approximate solution X with $w(X) \geq \frac{c-1}{c} w(G)$ ($w(X) \leq \frac{c-1}{c} w(G)$). The following theorem shows that there is no efficient algorithm for finding an approximation solution X with a better ratio $\frac{w(X)}{w(G)}$.

Theorem 8 *For every real $0 < \epsilon < \frac{1}{c}$, it is NP-complete to decide whether a given edge weighted, c-way, k-colored graph $G = (V, E, w)$ has an orthogonal subgraph X such that $w(X) \geq (\frac{c-1}{c} + \epsilon) w(G)$, (or $w(C) \leq (\frac{c-1}{c} - \epsilon) w(G)$).*

Proof: We will prove the case for maximum orthogonal partition problem only. For each pair of input instance G and W in the decision problem, we are going to construct, in polynomial time, a c-way k'-colored graph G' with weight function w' such that $w(X) \geq W$ is equivalent to $w'(X') \geq (\frac{c-1}{c} + \epsilon)w'(G')$, where X' is an orthogonal subgraph of G'.

(I) If $W < \lceil(\frac{c-1}{c} + \epsilon)w(G)\rceil$, then add in one pendant edge to G with weight W' to obtain G' (the new vertex forms one color class in G'), where W' satisfies $\lceil(\frac{c-1}{c} + \epsilon)(w(G) + W')\rceil - W' = W$ (the smallest integer W' satisfying $\frac{W+W'}{w(G)+W'} \geq \frac{c-1}{c} + \epsilon$ will do). Let the weight function on $E(G')$ be w'. Then $w'(G') = w(G) + W'$ and $w'(X') = w(X) + W'$. Therefore,

$$w'(X') \geq \lceil(\frac{c-1}{c} + \epsilon)w'(G')\rceil$$

is equivalent to

$$w(X) + W' \geq \lceil(\frac{n-1}{n} + \epsilon)(w(G) + W')\rceil,$$

which is equivalent to

$$w(X) \geq \lceil(\frac{c-1}{c} + \epsilon)(w(G) + W')\rceil - W' = W.$$

(II) Let $W > \lceil(\frac{c-1}{c} + \epsilon)w(G)\rceil$. We note that there exists a c-way, W-colored, edge weighted graph (H, w_1) such that $w_1(X_H) = \frac{c-1}{c}w_1(H)$, where X_H is an orthogonal subgraph with the maximum weight (the complete W-partite graph with c-vertices in each part and with a constant weight for each edge will do). Now we choose such a graph H satisfying (let the weight of $w_1(H)$ be big enough)

$$W + \frac{c-1}{c}w_1(H) \leq \lceil(\frac{c-1}{c} + \epsilon)(w(G) + w_1(H))\rceil.$$

Two cases arise:

Case (a). If $W = \lceil(\frac{c-1}{c} + \epsilon)(w(G) + w_1(H))\rceil - \frac{c-1}{c}w_1(H)$, then let G' be the disjoint union of G and H. The weight function w' of G' is defined as follows: $w'(e) = w(e)$ if $e \in E(G)$ and $w'(e) = w_1(e)$ if $e \in E(H)$. It is easy to see that $w'(G') = w(G) + w_1(H)$, $w'(X') = w(X) + \frac{c-1}{c}w_1(H)$. Therefore,

$$w'(X') \geq \lceil(\frac{c-1}{c} + \epsilon)w'(G')\rceil,$$

which is equivalent to

$$w(C) + \frac{c-1}{c}w_1(H) \geq \lceil(\frac{c-1}{c} + \epsilon)(w(G) + w_1(H))\rceil,$$

which is equivalent to

$$w(C) \geq \lceil(\frac{c-1}{c} + \epsilon)(w(G) + w_1(H))\rceil - \frac{c-1}{c}w_1(H) = W.$$

Case (b). If Case (a) does not hold, then $W_H = W + \frac{c-1}{c}w(H) < \lceil(\frac{c-1}{c} + \epsilon)(w(G) + w_1(H))\rceil$. Let G' and w' be defined as in Case (a). Since $w'(G') = w(G) + w_1(H)$, $w'(X') = w(X) + \frac{c-1}{c}w_1(H)$, it follows that

$$W_H < \lceil(\frac{c-1}{c} + \epsilon)(w'(G'))\rceil.$$

We modify G' to obtain G'' by the method used in (I) [that is, we add in one pendant edge with weight W'_H to G' (the weight W'_H is determined by the method in (I)), the resulting graph is denoted by $G^{''}$ and its weight function is w'']. Let X'' be an orthogonal subgraph of G''. Then

$$w''(X'') \geq \lceil(\frac{c-1}{c} + \epsilon)w''(G'')\rceil$$

is equivalent to

$$w'(X') + W'_H \geq \lceil(\frac{c-1}{c} + \epsilon)(w'(G') + W'_H)\rceil,$$

which is equivalent to

$$w'(X') \geq \lceil(\frac{c-1}{c} + \epsilon)(w'(G') + W'_H)\rceil - W'_H = W_H,$$

which is equivalent to

$$w(X) + \frac{c-1}{c}w_1(H) \geq W + \frac{c-1}{c}w_1(H),$$

which is equivalent to

$$w(X) \geq W.$$

In any case, we have constructed, in polynomial time, a c-way k'-colored graph G' with the weight function w' such that $w(X) \geq W$ is equivalent to $w'(X') \geq (\frac{c-1}{c} + \epsilon)w'(G')$, where X' is an orthogonal subgraph of G'. Therefore, it is NP-complete to decide whether or not $w(X) \geq (\frac{c-1}{c} + \epsilon)w(G)$. This completes the proof. ∎

5 Conclusion

We have provided efficient and deterministic algorithms for four graph partition problems. Our algorithms produce good approximate solutions, both in theory, as shown by our mathematical analysis, and in practice, as confirmed by our experimental results. We have also shown that there is no efficient algorithm is likely to yield a better ratio of the weight of the approximation solution over the weight of the input graph.

Acknowledgments

We thank the anonymous referees for their valuable comments.

References

[1] U. Banerjee, R. Eigenmann, A. Nicolau and D. A. Padua, Automatic Program Parallelization, Proceedings of the IEEE, Vol. 81, No. 2, 1993, pp. 211–243.

[2] J. A. Bondy and U. S. R. Murty, Graph Theory with Applications, Elsevier, New York, 1976.

[3] J. Exoo, J. Jiang and C. Zhao, An Application of Multi-Dimensional Matching to Logistics, Utilitas Mathematica 57(2000), pp. 227–235.

[4] M. Gupta and P. Banerjee, Demonstration of Automatic Data Partitioning Techniques for Parallelizing Compilers on Multicomputers, IEEE Transactions on Parallel and Distributed Systems, No. 2, 3 (1992), pp. 179–193.

[5] K. Knoble, J. Lukas and G. L. Steele, Data Optimization, Allocation of Arrays to Reduce Communication in SIMD Machines, Journal of Parallel and Distributed Computing, 8 (1990), pp., 102–118.

[6] J. Li and M. Chen, Index Domain Alignment: Minimizing Cost of Cross-reference between Distributed Arrays, in Proceedings of the third Symposium on Frontiers of Massively Parallel Computation, College Park, October, 1990, pp. 424–433.

[7] G. He, Jiping Liu and Cheng Zhao, Extremal n-colorable Subgraphs with Index Domain Alignment Restrictions, Submitted for publication.

[8] L. Lovász and M. Plummer, Matching Theory, Annals of Discrete Mathematics 29, North-Holland, 1986.

[9] J. Ramanujam and P. Sadayappan, Compile-Time Techniques for Data Distribution Memory Machines, IEEE Transactions on Parallel and Distributed Systems, No.4, 2 (1991), pp. 472–483.

[10] M. Zhou and C. Zhao, Quality Planning in Manufacturing Systems, In the Proceedings, 8th Annual Industrial Engineering Research Conference (IERC), May 23–26, 1999, Pheonix, AZ.