JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS: Vol. 110, No. 3, pp. 545–556, September 2001 (© 2001)

Quadratic Integer Programming with Application to the Chaotic Mappings of Complete Multipartite Graphs[1,2]

H. L. Fu,[3] C. L. Shiue,[4] X. Cheng,[5] D. Z. Du,[6] and J. M. Kim[7]

Communicated by P. M. Pardalos

Abstract. Let α be a permutation of the vertex set $V(G)$ of a connected graph G. Define the total relative displacement of α in G by

$$\delta_\alpha(G) = \sum_{x,y \in V(G)} |d_G(x,y) - d_G(\alpha(x), \alpha(y))|,$$

where $d_G(x,y)$ is the length of the shortest path between x and y in G. Let $\pi^*(G)$ be the maximum value of $\delta_\alpha(G)$ among all permutations of $V(G)$. The permutation which realizes $\pi^*(G)$ is called a chaotic mapping of G. In this paper, we study the chaotic mappings of complete multipartite graphs. The problem is reduced to a quadratic integer programming problem. We characterize its optimal solution and present an algorithm running in $O(n^5 \log n)$ time, where n is the total number of vertices in a complete multipartite graph.

Key Words. Chaotic mapping, complete multipartite graph, quadratic integer programming, optimal solution.

[1]This paper was supported by ROC Grant NSC-89-2115-M-009-019, ROC Grant NSC-89-2119-M-009-002, and PRC Program 973 on Information Technology and High-Performance Software.

[2]This work was motivated by the work of Ref. 1 and a personal communication with Dr. K. B. Reid. We wish to thank him for his references and encouragement.

[3]Professor, Department of Applied Mathematics, National Chiao Thung University, Hsin Chu, Taiwan.

[4]Professor, Mathematics Division, National Center for Theoretical Science, National Tsing Hua University, Hsin Chu, Taiwan.

[5]Graduate Student, Department of Computer Science and Engineering, University of Minnesota, Minneapolis, Minnesota.

[6]Professor, Department of Computer Science and Engineering, University of Minnesota, Minneapolis, Minnesota.

[7]Professor, Institute of Applied Mathematics, Chinese Academy of Sciences, Beijing, PRC.

1. Introduction

Let α be a permutation of the vertex set $V(G)$ of a connected graph G. Define the total relative displacement of α in G by

$$\delta_\alpha(G) = \sum_{x,y \in V(G)} |d_G(x-y) - d_G(\alpha(x), \alpha(y))|,$$

where $d_G(x, y)$ is the length of the shortest path between x and y in G. It is easy to see that a permutation α of $V(G)$ is an automorphism of G if and only if the total relative displacement of α in G is zero. Let $\pi(G)$ and $\pi^*(G)$ denote respectively the smallest nonzero total relative displacement and the largest total relative displacement in G. The permutation which realizes $\pi^*(G)$ is called a chaotic mapping of G. The chaotic mapping is related to the sorting problem in computer science (Refs. 2–4). Computing $\pi(G)$ and $\pi^*(G)$ is an important research topic in graph theory (Refs. 1, 5, 6). The exact value of $\pi(G)$ has been obtained for G, when G is a path or a complete multipartite graph $K_{n_1,n_2,\ldots,n_t}$ in which all vertices are partitioned into t subsets with cardinalities $n_1, n_2, \ldots, n_t$, respectively; an edge (u, v) exists if and only if two vertices u and v belong to different subsets.

(i) See Ref. 5. Let G be a path with n vertices. Then, the minimum total relative displacement is $\pi(G) = 2n - 4$.

(ii) See Ref. 1. Let $1 \leq n_1 \leq n_2 \leq \cdots \leq n_t$, where $t \geq 2$ and $n_t \geq 2$. Then,

$$\pi(K_{n_1,n_2,\ldots,n_t}) = \begin{cases} 2n_{h+1} - 2, & \text{if } 1 = n_1 = \cdots = n_h < n_{h+1} \leq \cdots \leq n_t, \\ & \text{and } t \geq (h+1), \text{ for some } h \geq 2, \\ 2n_{k_0}, & \text{if } 1 = n_1 < n_2 \text{ or } n_1 \geq 2 \text{ and} \\ & n_{k+1} = n_k + 1 \text{ for some } k,\ 1 \leq k \leq t-1, \\ & \text{and } 2 + n_{k_0} \leq n_1 + n_2, \text{ where } k_0 \\ & \text{is the smallest index for which} \\ & n_{k_0+1} = n_{k_0} + 1, \\ 2(n_1 + n_2 - 2), & \text{otherwise.} \end{cases}$$

In this paper, we study how to compute $\pi^*(K_{n_1,n_2,\ldots,n_k})$. This problem can be reduced to a quadratic integer programming due to the following result.

Lemma 1.1. See Ref. 1. Let $K_{n_1,n_2,\ldots,n_t} = (X_1, X_2, \ldots, X_t)$ be a complete t-partite graph with partite sets $X_1, X_2, \ldots, X_t$. Let α be a permutation of

$V(K_{n_1,n_2,\ldots,n_t})$. For each $1\leq i, j\leq t$, define

$$a_{ij} = |A_{ij}(\alpha)| = |\{x \mid x\in X_i \text{ and } \alpha(x)\in X_j\}|.$$

Then,

$$\delta_\alpha(K_{n_1,n_2,\ldots,n_t}) = \sum_{i=1}^{t} n_i^2 - \sum_{1\leq i,j\leq t} a_{ij}^2. \tag{1}$$

Since $\sum_{i=1}^{t} n_i^2$ is fixed for a given complete multipartite graph, the problem of determining $\pi^*(K_{n_1,n_2,\ldots,n_k})$ is equivalent to the following quadratic integer programming:

$$\text{(QIP)} \quad \min \quad \sum_{1\leq i,j\leq t} a_{ij}^2,$$

$$\text{s.t.} \quad \sum_{i=1}^{t} a_{ij} = n_j, \qquad \text{for } 1\leq j\leq t,$$

$$\sum_{j=1}^{t} a_{ij} = n_i, \qquad \text{for } 1\leq i\leq t,$$

$$a_{ij}\geq 0 \text{ are integers.}$$

In this paper, we characterize the optimal solution of this minimization problem and present an algorithm running in $O(n^5 \log n)$ time, where n is the number of vertices in a complete multipartite graph. We also give explicit values of $\pi^*(K_{n_1,n_2,\ldots,n_k})$, based on the characterization, in some special cases.

2. Characterization of the Optimal Solution

Let $A = (a_{ij})$ be a $t\times t$ nonnegative matrix. We call

$$C = (a_{i_1j_1}, a_{i_1j_2}, a_{i_2j_2}, a_{i_2j_3}, a_{i_3j_3}, \ldots, a_{i_sj_s}, a_{i_sj_1})$$

a cycle of length $2s, s\geq 2$, in A. A cycle C of length $2s$ is said to be overweight if either $a_{i_kj_k}\geq 1$ for $1\leq k\leq s$ and

$$a_{i_1j_1} - a_{i_1j_2} + a_{i_2j_2} - a_{i_2j_3} + a_{i_3j_3} - \cdots + a_{i_sj_s} - a_{i_sj_1} > s,$$

or $a_{i_kj_{k+1}}\geq 1$ for $1\leq k\leq s$, where $j_{s+1} = j_1$, and

$$-a_{i_1j_1} + a_{i_1j_2} - a_{i_2j_2} + a_{i_2j_3} - a_{i_3j_3} + \cdots - a_{i_sj_s} + a_{i_sj_1} > s.$$

Below, we show a matrix with overweight cycle of length 2:

$$A=\begin{bmatrix} 3 \rightarrow 1 & & 1 \\ \uparrow \quad \downarrow & & \\ 1 \leftarrow 2 & & 1 \\ & & \\ 1 \quad\; 1 & & 0 \end{bmatrix}.$$

It is not difficult to see that, since $A=(a_{ij})$ has an overweight cycle, $\sum a_{ij}^2 = 19$ is not of minimum value under the constraints that the row sums and column sums are fixed. The next matrix $A'=(a'_{ij})$ reaches a smaller value $\sum a'^2_{ij} = 17$,

$$A'=\begin{bmatrix} 2 & 2 & 1 \\ 2 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

Theorem 2.1. $A=(a_{ij})$ is an optimal solution of Problem (QIP) if and only if no overweight cycle exists in A.

Proof. Necessity. Suppose that A has an overweight cycle $C=(a_{i_1 j_1}, a_{i_1 j_2}, \ldots, a_{i_s j_1})$. Without loss of generality, assume that $a_{i_k j_k} \geq 1$ for $1 \leq k \leq s$ and

$$a_{i_1 j_1} - a_{i_1 j_2} + a_{i_2 j_2} - a_{i_2 j_3} + a_{i_3 j_3} - \cdots + a_{i_s j_s} - a_{i_s j_1} > s.$$

Define $A'=(a'_{ij})$, where

$$a'_{ij} = \begin{cases} a_{ij} - 1, & \text{if } (i,j) = (i_k, j_k) \text{ for some } 1 \leq k \leq s, \\ a_{ij} + 1, & \text{if } (i,j) = (i_k, j_{k+1}) \text{ for some } 1 \leq k \leq s, \\ a_{ij}, & \text{otherwise.} \end{cases}$$

Now,

$$\begin{aligned} \sum a_{ij}^2 - \sum a'^2_{ij} &= a_{i_1 j_1}^2 + a_{i_1 j_2}^2 + \cdots + a_{i_s j_1}^2 - a'^2_{i_1 j_1} - a'^2_{i_1 j_2} - \cdots - a'^2_{i_s j_1} \\ &= 2(a_{i_1 j_1} + a_{i_2 j_2} + \cdots + a_{i_s j_s}) \\ &\quad -2(a_{i_1 j_2} + a_{i_2 j_3} + \cdots + a_{i_s j_1}) - 2s \\ &> 0. \end{aligned}$$

Therefore, $\sum a_{ij}^2$ is not the minimum. Note that the row sums and column sums of A and A' are equal respectively. Hence, we have the proof for necessity.

Sufficiency. For contradiction, assume that all cycles of A are not overweight and $\sum a_{ij}^2$ is not the minimum. Let $A^* = (a_{ij}^*)$ denote an optimal solution.

Let

$$\Delta_{ij} = a_{ij} - a_{ij}^*, \qquad 1 \leq i, j \leq t.$$

Define a directed bipartite multigraph G with bipartition (X, Y), where

$$X = \{x_1, x_2, \ldots, x_t\}, \qquad Y = \{y_1, y_2, \ldots, y_t\},$$

x_i joins to y_j with Δ_{ij} edges if $\Delta_{ij} > 0$, and x_i joins from y_j with Δ_{ij} edges if $\Delta_{ij} < 0$. Since

$$\sum_{j=1}^{t} \Delta_{ij} = 0, \qquad \text{for } 1 \leq i \leq t,$$

and

$$\sum_{i=1}^{t} \Delta_{ij} = 0, \qquad \text{for } 1 \leq j \leq t,$$

the outdegree and indegree of each vertex in G are equal. Thus, each component of G has a directed Eulerian circuit, and hence G can be decomposed into directed cycles $C_1, C_2, \ldots, C_m$. For each cycle C_l, define

$$w(C_l) = \sum_{(x_i, y_j) \in C_l} a_{ij} - \sum_{(y_j, x_i) \in C_l} a_{ij}.$$

Note that,

$$\Delta_{ij} > 0, \qquad \text{for } (x_i, y_j) \in C_l,$$

and that $\Delta_{ij} > 0$ implies $a_{ij} \geq 1$, since $a_{ij}^* \geq 0$. Thus,

$$a_{ij} \geq 1, \qquad \text{for } (x_i, y_j) \in C_l.$$

This means that, if $w(C_l) > |E(C_l)|/2$, where $|E(C_l)|$ is the number of edges in cycle C_l, then C_l introduces an overweight cycle in A. Since A has no overweight cycle, we have

$$w(C_l) \leq |E(C_l)|/2, \qquad \text{for } 1 \leq l \leq m.$$

Therefore,

$$\begin{aligned}\sum_{1\le i,j\le t} a_{ij}^2 - \sum_{1\le i,j\le t} a_{ij}^{*2} &= \sum_{1\le i,j\le t} a_{ij}^2 - \sum_{1\le i,j\le t} (a_{ij} - \Delta_{ij})^2 \\ &= 2 \sum_{1\le i,j\le t} a_{ij}\Delta_{ij} - \sum_{1\le i,j\le t} \Delta_{ij}^2 \\ &= 2 \sum_{l=1}^{m} w(C_l) - \sum_{1\le i,j\le t} \Delta_{ij}^2 \\ &\le 2 \sum_{l=1}^{m} (1/2)|E(C_l)| - \sum_{1\le i,j\le t} \Delta_{ij}^2 \\ &= |E(G)| - \sum_{1\le i,j\le t} \Delta_{ij}^2 \\ &= \sum_{i,j} |\Delta_{ij}| - \sum_{1\le i,j\le t} \Delta_{ij}^2 \\ &\le 0,\end{aligned}$$

where $|E(G)|$ denotes the number of edges in G. This contradicts the fact that $A^* = (a_{ij}^*)$ is an optimal solution, while $A = (a_{ij})$ is not. □

With the above characterization, we are able to find a chaotic mapping for certain complete multipartite graphs.

Corollary 2.1. Let $K_{m,n}$ be a complete bipartite graph, and let $l = \min\{m, n\}$, where $m + n \ge 4$. Then,

$$\pi^*(K_{m,n}) = 2(m + n - 2)\, l.$$

Proof. Let

$$A = \begin{bmatrix} m-l & l \\ l & n-l \end{bmatrix}.$$

Then, A has no overweight cycle. By Theorem 2.1 and (1),

$$\begin{aligned}\pi^*(K_{m,n}) &= (m^2 + n^2) - [(m - l)^2 + 2l^2 + (n - l)^2 \\ &= (-4)l^2 + 2(m + n)\, l] \\ &= 2(m + n - 2)\, l.\end{aligned}$$

□

Corollary 2.2. In $K_{n_1,n_2,\ldots,n_t}$, if $a_{ij}=(1/t)(n_i+n_j)-(1/t^2)\sum_{i=1}^{t} n_i$ is a nonnegative integer for each $1\le i, j\le t$, then $A=(a_{ij})$ gives a chaotic mapping of $K_{n_1,n_2,\ldots,n_t}$ and

$$\pi^*(K_{n_1,n_2,\ldots n_t})=(1-2/t)\sum_{i=1}^{t} n_i^2+(1/t^2)\left(\sum_{i=1}^{t} n_i\right)^2.$$

Proof. Since for each i and i',

$$a_{ij}-a_{i'j}=(n_i-n_{i'})/t$$

and for each j and j',

$$a_{ij}-a_{ij'}=(n_j-n_{j'})/t,$$

all cycles in A have weight zero. By Theorem 2.1, $\sum a_{ij}^2$ is minimum. Thus, A determines a chaotic mapping of $K_{n_1,n_2,\ldots n_t}$ by mapping a_{ij} elements of the partite set X_i to the partite set X_j. Furthermore, it is easy to check that

$$\pi^*(K_{n_1,n_2,\ldots,n_t})=\sum_{i=1}^{t} n_i^2-\sum_{1\le i,j\le t} a_{ij}^2. \qquad \square$$

Example 2.1. For a complete 3-partite graph $K_{3,6,9}$ with partite sets

$$X_1=\{1,2,3\},$$

$$X_2=\{4,5,6,7,8,9\},$$

$$X_3=\{10,11,12,13,14,15,16,17,18\},$$

from Corollary 2.2, we have

$$A=\begin{bmatrix}0&1&2\\1&2&3\\2&3&4\end{bmatrix},$$

and

$$\pi^*(K_{3,6,9})=(1-2/3)(3^2+6^2+9^2)+(1/9)(3+6+9)^2=78.$$

One of the chaotic mappings is as follows:

$$\alpha=\left[\begin{array}{ccc|cccccc|ccccccccc}1&2&3&4&5&6&7&8&9&10&11&12&13&14&15&16&17&18\\4&10&11&1&5&6&12&13&14&2&3&7&8&9&15&16&17&18\end{array}\right].$$

For a special t-tuple $(n_1,n_2,\ldots,n_t)$, we can also find $\pi^*(K_{n_1,n_2,\ldots,n_t})$.

Corollary 2.3. Let $s=(s_1, s_2, \ldots, s_t)$ be a t-tuple such that $s_i=0$ or 1, $1 \leq i \leq t$, and let $n_1-s_1, n_2-s_2, \ldots, n_t-s_t$ be a graphical sequence of a simple graph. Then,

$$\pi^*(K_{n_1,n_2,\ldots,n_t}) = \sum_{i=1}^{t} n_i(n_i-1).$$

Proof. Since $(n_1-s_1, n_2-s_2, \ldots, n_t-s_t)$ is a graphical sequence, let G be a graph with such degree sequence. Moreover, let $B(G)=[b_{ij}]$ be the adjacent matrix of G. Now, define $A=[a_{ij}]$ such that

$$a_{ij}=b_{ij}, \quad \text{for } i \neq j,$$

$$a_{ii}=s_i, \quad \text{for } 1 \leq i, j \leq t.$$

By direct counting, we see that A has no overweight cycles; thus, by Theorem 2.1, we have an optimal solution. By (1), we can figure out easily $\pi^*(K_{n_1,n_2,\ldots,n_t})$. □

3. Algorithm

Theorem 2.1 suggests the following algorithm to compute $\pi^*(K_{n_1,n_2,\ldots,n_t})$.

Algorithm 3.1. Start from an initial matrix (a_{ij}),

$$a_{ij} = \begin{cases} n_i, & \text{if } i=j, \\ 0, & \text{if } i \neq j. \end{cases}$$

Carry out the following steps in each iteration.

Step 1. Check whether the matrix (a_{ij}) has an overweight cycle. If not, then stop; we obtain

$$\pi^*(K_{n_1,n_2,\ldots,n_t}) = \sum_{i=1}^{t} n_i^2 - \sum_{1 \leq i,j \leq t} a_{ij}^2.$$

Otherwise, go to Step 2.

Step 2. Suppose that $a_{i_1j_1}, a_{i_1j_2}, a_{i_2j_2}, \ldots, a_{i_sj_s}, a_{i_sj_1})$ is an overweight cycle, with $a_{i_kj_k} \geq 1$ for $1 \leq k \leq s$ and $a_{i_1j_1} - a_{i_1j_2} + a_{i_2j_2} - \cdots + a_{i_sj_s} - a_{i_sj_1} > s$. Then, set

$$a_{i_1j_1} \leftarrow a_{i_1j_1} - 1,$$

$$a_{i_1j_2} \leftarrow a_{i_1j_2} + 1,$$

$a_{i_2 j_2} \leftarrow a_{i_2 j_2} - 1,$

$\ldots,$

$a_{i_s j_s} \leftarrow a_{i_s j_s} - 1,$

$a_{i_s j_1} \leftarrow a_{i_s j_1} + 1.$

Go to the next iteration.

Note that, initially,

$$\sum_{1 \leq i,j \leq t} a_{ij}^2 = \sum_{i=1}^{t} n_i^2.$$

In each iteration, if Step 2 is performed, then this sum decreases at least by one. Therefore, the algorithm stops within $\sum_{i=1}^{t} n_i^2 = O(n^2)$ iterations, where $n = \sum_{i=1}^{t} n_i$. In the following, we explain how to implement each iteration in $O(n^3 \log n)$ time. Therefore, we have the following theorem.

Theorem 3.1. $\pi^*(K_{n_1, n_2, \ldots, n_t})$ can be computed in $O(n^5 \log n)$ time.

Proof. First, we construct a directed graph H with vertex set

$$V = \{v_{ij} | a_{ij} \geq 1\},$$

edge set

$$E = \{(v_{ij}, v_{i'j'}) | i \neq i', j \neq j'\},$$

and edge weight

$$w(v_{ij}, v_{i'j'}) = (1/2)(a_{ij} + a_{i'j'}) - a_{ij'} - 1.$$

Note that

$$|V| \leq \sum_{i=1}^{t} n_i = n.$$

Therefore, H can be constructed in $O(n^2)$ time. □

Lemma 3.1. The matrix (a_{ij}) has an overweight cycle if and only if H has a simple directed cycle with positive total weight.

Proof. H has a directed cycle $(v_{i_1 j_1}, v_{i_2 j_2}, \ldots, v_{i_s j_s})$ with a positive total weight if and only if $a_{i_k j_k} \geq 1$ for $1 \leq k \leq s$ and

$$a_{i_1 j_1} - a_{i_1 j_2} + a_{i_2 j_2} - \cdots + a_{i_s j_s} - a_{i_s j_1} > s.$$ □

Now, let P be the subgraph of H induced by all edges with positive weight, and let Q be the subgraph of H induced by all edges with nonpositive weight. Replace each negative edge-weight in Q by its absolute value. Let Q' be the resulting edge-weighted directed graph.

(i) Check Whether P Contains a Directed Cycle. This can be done in at most $O(n^2)$ time. If yes, then we have found already a directed cycle of H with positive total weight. Otherwise, P is acyclic.

(ii) Compute the Shortest Path in Q' for Every Pair of Vertices. This can be done in $O(n^3)$ time. Let $q(v_{ij}, v_{i'j'})$ denote the length of the shortest path from v_{ij} to $v_{i'j'}$ in Q'.

Make $n+1$ disjoint copies $P_0, P_1, \ldots, P_n$ of P. Denote by v_{ij}^k the copy of the vertex v_{ij} in P_k. For each pair of vertices v_{ij} and $v_{i'j'}$, with $q(v_{ij}, v_{i'j'}) < \infty$, add edges $v_{ij}^{k-1}, v_{i'j'}^k)$ for $1 \leq k \leq n$, each with weight $-q(v_{ij}, v_{i'j'})$. Meanwhile, delete all edges in P_0. This results in an acyclic directed graph R with $O(n^2)$ vertices.

Lemma 3.2. *H has a simple directed cycle with positive total weight if and only if either P contains a cycle or there exist i, j, k such that R has a directed path from v_{ij}^0 to v_{ij}^k with positive total weight.*

Proof. Suppose that H has a simple directed cycle C with positive total weight. If every edge has a positive weight, then P contains a cycle. Otherwise, C can be decomposed into $2k$ paths alternatively in P and Q. Suppose that v_{ij} is the starting vertex of such a path in Q. It is easy to see that we can find a path from v_{ij}^0 to v_{ij}^k in graph R, with positive total weight, corresponding to cycle C.

Conversely, if P contains a cycle or if there exist i, j, k such that R has a directed path from v_{ij}^0 to v_{ij}^k with positive total weight, then it is easy to find a directed cycle C with positive total weight in H. This cycle may not be simple. However, it can be decomposed into several simple directed cycles, at least one with positive total weight, since the sum of weights of those simple directed cycles equals the positive total weight of C. □

Now, for each pair of v_{ij}^0 and v_{ij}^k, compute the longest path from v_{ij}^0 to v_{ij}^k in R. Since R is acyclic, this can be done trivially by dynamic programming in $O(n^5)$ time.

In fact, there are at most n v_{ij}^0s and finding all longest paths from v_{ij}^0 to other vertices needs at most $O(n^2)$ time. However, we describe next a clever way running in $O(n^3 \log n)$ time.

Construct R' from R by adding edges (v_{ij}^{k-1}, v_{ij}^k) for all $1 \leq k \leq n$ and $1 \leq i, j \leq t$. Clearly, R' is a disjoint union of n copies of the subgraph induced by the vertices in $P_0 \cup P_1$. Let $g(u, v)$ denote the length of the longest path from vertex u to vertex v in R'. It is easy to see that there exist i, j, k such that R has a directed path from v_{ij}^0 to v_{ij}^k with positive total weight if and only if $g(v_{ij}^0, v_{ij}^n) > 0$.

(iii) For every pair of vertices v_{ij}^1 and $v_{i'j'}^2$, compute the longest path from v_{ij}^1 to $v_{i'j'}^1$. This can be done in $O(n^3)$ time since P_1 is acyclic.

(iv) For every pair of vertices v_{ij}^0 and $v_{i'j'}^1$, compute the longest path from v_{ij}^0 to $v_{i'j'}^1$

$$g(v_{ij}^0, v_{i'j'}^1) = \max_{u \in V(P_1)} (-q(v_{ij}^0, u) + g(u, v_{i'j'}^2)),$$

where $V(P_1)$ denotes the vertex set of P_1. This can be done in $O(n^3)$ time.

Similarly, we can compute all

$$g(v_{ij}^0, v_{i'j'}^2) = \max_{u \in V(P_1)} (g(v_{ij}^0, u) + g(u, v_{i'j'}^2)),$$

$$g(v_{ij}^0, v_{i'j'}^4) = \max_{u \in V(P_2)} (g(v_{ij}^0, u) + g(u, v_{i'j'}^4)),$$

$$g(v_{ij}^0, v_{i'j'}^8) = \max_{u \in V(P_4)} (g(v_{ij}^0, u) + g(u, v_{i'j'}^8)),$$

$$\ldots$$

$$g(v_{ij}^0, v_{ij}^n),$$

totally in $O(n^3 \log n)$ time.

This completes the proof of Theorem 3.1. □

4. Conclusions

We have studied a quadratic integer programming problem with application in a graph problem and found an algorithm to solve the graph problem in $O(n^5 \log n)$ time, where n is the number of vertices in the given graph. This is the first polynomial-time algorithm for this graph problem.

References

1. Reid, K. B., *Total Relative Displacement of Vertex Permutations of $K_{n_1,n_2,\ldots,n_t}$*, Preprint, California State University, San Marcos, California, 2000.
2. Chao, C. C., and Liang, W. Q., *Arranging n Distinct Numbers on a Line or a Circle to Reach Extreme Total Variations*, European Journal of Combinatorics, Vol. 13, pp. 325–334, 1992.

3. Hwang, F. K., *Extremal Permutations with Respect to Weak Majorizations*, European Journal of Combinatorics, Vol. 17, pp. 637–645, 1996.
4. Leveopoulos, C., and Petersson, O., *Heapsort-Adapted for Presorted Files*, Proceedings of a 1989 Workshop on Algorithms and Data Structures, Lecture Notes in Computer Science, Springer Verlag, Berlin, Germany, Vol. 382, pp. 499–509, 1989.
5. Aitken, W., *Total Relative Displacement of Permutations*, Journal of Combinatorial Theory, Vol. 87A, pp. 1–21, 1999.
6. Chartrand, G., Gavlas, H., and Vanderjagt, D. W., *Near-Automorphisms of Graphs*, Proceedings of the 1996 8th Quadrennial International Conference on Graph Theory, Combinatorics, Algorithms, and Applications, Edited by Y. Alavi, D. Lick, and A. J. Schwenk, New Issues Press, Kalamazoo, Michigan, Vol. 1, pp. 181–192, 1999.