https://doi.org/10.26552/com.C.2003.4.47-48

KOMUNIKÁCIE COMMUNICATIONS

Peter Czimmermann *

A NOTE ON USING GRAPHS IN REGULAR SCHEDULING PROBLEMS

This paper deals with regular permutation scheduling on graphs. Peško and Czimmermann introduced this problem (in [3]) and it is generalisation of a matrix permutation problem. The goal is to minimise differences between row sums of a real matrix that represents a schedule, but external conditions don't allow moving matrix elements arbitrarily. The conditions can be represented by permutation obtained from a certain graph.

1. Introduction

The matrix permutation problem (MPP) is to minimise differences between row sums by permuting elements in columns of a matrix. This problem was defined first by Š. Peško in [5]. The exact definition of MPP (from [1]) is:

There is given matrix $(a_{i,j}) \in R^{m \times n}$ (we will call it scheduling matrix). We need to find permutations of the numbers 1, ..., m, $\pi_j = (\pi_j(1), ..., \pi_j(i))$ for $j = 1, ..., n$ such that a certain Schur-convex function $f(s_1, ..., s_m)$ is minimal (where s_i is the sum of elements in i-th row $s_i = \sum_{j=1}^{n} a_{\pi_j(i),j}$). The most used Schur-convex functions are:

- $f_{sqr}(s_1, ..., s_m) = s_1^2 + s_2^2 + ... + s_m^2$
- $f_{dif}(s_1, ..., s_m) = \max(s_1, ..., s_m) - \min(s_1, ..., s_m)$
- $f_{max}(s_1, ..., s_m) = \max(s_1, ..., s_m)$
- $f_{sqr}^{\delta}(s_1, ..., s_m) = (s_1 - \delta)^2 + ... + (s_m - \delta)^2$ where $\delta = \frac{(s_1, ..., s_m)}{m}$

In [7] it was shown that MPP is *NP*-hard except two column case for which the polynomial algorithm was found ([2]).

We can gain a generalisation of MPP, if we will not limit ourselves to column permutations and permutations will be allowed by a certain graph (its vertices represent elements of a scheduling matrix, element $a_{i,j}$ is represented by the vertex $v_{i,j}$). We will call it *regular permutation scheduling on graphs* (RPSG). In [3] there were given two ways how to generate needed permutations. We can use:

1. graph of permitted moves,
2. graph automorphisms.

(The graph of permitted moves will be denoted GPM and the related problem GPM-P.)

2. Graph of permitted moves

In this section we will continue in our work started in [3]. It is known from this work that MPP can be solved as a special case of GPM-P and if GPM is a complete digraph with loops on every vertex, two-column case can be solved for irregularity measure fsqr as a minimal perfect matching in a certain complete graph. Unfortunately there exist graphs of permitted moves for which two column case can't be solved as a minimal perfect matching in a graph (it was one of the open problems introduced in [3] related to two column GPM-P).

Example 1. *In Figure 1 we can see the graph of permitted moves G_M and related graph G whose vertex set is the same as the vertex set of G_M and vertices $v_{i,j}$, $v_{k,l}$ are connected by an edge of weight $w = (a_{i,j} + a_{k,l})^2$ if and only if there exist directed edges $(v_{i,j}, v_{x,y})$ and $(v_{k,l}, v_{x,z})$ in G_M ($y, z \in \{1, 2\}$ $y \neq z$) it means that $a_{i,j}$ and $a_{k,l}$ could form the row in a new matrix). We can see that for perfect matching with edges $v_{1,1}$, $v_{1,2}$, $v_{2,1}$, $v_{2,2}$ and $v_{3,1}$, $v_{3,2}$ there doesn't exist any permitted permutation of elements of the matrix A for which elements $a_{1,1}$, $a_{1,2}$ form some row of new scheduling matrix and elements $a_{2,1}$ $a_{2,2}$ another one.*

If we use for two column case another Schur-convex function f and we can transform GPM to a graph in which we find perfect matching then the solution of GPM-P will be perfect matching that minimises f.

Example 2. *There is given a matrix $A_{3x2} = (a_{i,j})$, GPM is a complete directed graph with loop on every vertex and fmax $(s_1, s_2, s_3) = \max(s_1, s_2, s_3)$ is the irregularity measure. We can solve this problem as finding the perfect matching with minimal $f_{max}(s_1, s_2, s_3)$ in a complete graph with the same vertex set as GPM has (where s_1, s_2, s_3 are weights of edges used in this matching and edge $[v_{i,j}, v_{k,l}]$ has weight $a_{i,j} + a_{k,l}$).*

* **Peter Czimmermann**
University of Žilina, Faculty of Managements and Informatics, Slovakia, E-mail: petocimo@frcatel.fri.utc.sk

We present an algorithm that finds perfect matching in the graph minimizing the function f_{max} in graph $G = (V, E)$. (A similar problem is presented in [6] p. 260.)

1. let $n = |V|$ is even and $m = |E|$
2. sort the edges into nondecreasing sequence S

$$S = (\underbrace{e_1, ..., e_{i_1}}_{c_1}, \underbrace{e_{i_1+1}, ..., e_{i_2}}_{c_2}, ..., \underbrace{e_{i_{k-1}+1}, ..., e_{i_k}}_{c_3})$$

where $i_k = m$ and $c_1, ..., c_k$ are weights of the edges

3. find $j \in \{1, ..., k\}$ for which $i_{j-1} + 1 \leq n/2 \leq i_j$
4. if such j does not exist *then* STOP (there is no perfect matching in G) else let $r := j$
5. delete every edge e in G for which its weight $c(e) > c_r$ (we obtain a graph

$$G' = G - \{e; e \in E, c(e) > c_r\})$$

6. for every edge $e = \{u, v\}$ $(u, v \in V)$ of weight $c(e) = c_r$ do:
   - let $G'' = G' - u - v$
   - find perfect matching in G''
   - if M is perfect matching in G'' then STOP (edge e and edges of M form perfect matching in G for which is function f_{max} minimal)
7. if $r < k$ then $r := r + 1$ and go to 5 else STOP there isn't any perfect matching in G.

It isn't hard to show that the presented algorithm is polynomial. In step 2 it makes $O(n^2 \log n)$ steps. An $O(n^{5/2})$ algorithm is known for finding perfect matching in graphs [4, 6] and this algorithm must be made $m < n^2$ times at most so that our algorithm has complexity $O(n^{9/2})$.

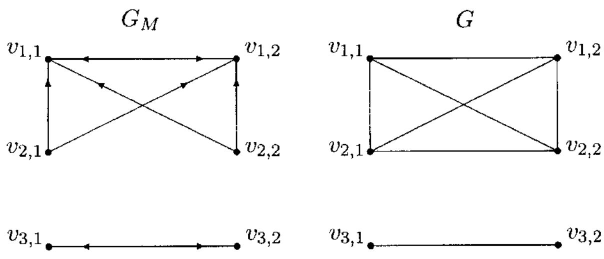

Fig. 1. Graphs GM and G

3. Conclusions

In this paper we dealt with using the graphs in regular scheduling problems. It seems that perfect matching that is minimal with consideration to some Schur-convex function is an important concept. For most Schur-convex functions f it remains an open problem to find an polynomial algorithm for perfect matching that is minimal with consideration to f. Similarly, two column case is unsolved if perfect matching can't be used for representing a convenient solution.

The research of the author is supported by Slovak Scientific Grant Agency under grant NO.1/0490/03.

References

[1] ČERNÝ, J., KLUVÁNEK, P.: *Principles of Mathematical Theory of Transport* (in Slovak), VEDA, Bratislava, (1991)

[2] ČERNÝ, J., VAŠEK, K., PEŠKO, Š., PALÚCH, S., ENGELTHALLER, D.: *Transport schedulings and their optimization* (in Slovak), Research report III-8-9/03, Research Institute of Transport, Žilina, (1986)

[3] CZIMMERMANN P., PEŠKO, Š.: *The Regular Permutation Scheduling on Graphs,* Journal of Information, Control and Management Systems, vol. 1, (2003)

[4] EVEN, S., KARIV, O.: *An $O(n^{5/2})$ Algorithm for Maximum Matching in General Graphs*, Proc. 16th Annual Symp. on Foundations of Computer Science, IEEE, New York, (1975)

[5] PEŠKO, Š., VAŠEK, K.: *Optimization of Transport Scheduling* (in Slovak), Research report III-8-6/09.3, Research Institute of Transport, Žilina, (1983)

[6] PLESNÍK, J.: *Graph Algorithms* (in Slovak), VEDA, Bratislava, (1983)

[7] TEGZE, M., VLACH, M.: *The Matrix Permutation Problem,* Tech. Univ. Graz Bericht 84-54, (1984).