Journal of Combinatorial Optimization, 9, 49–57, 2005

Scheduling in Switching Networks with Set-Up Delays*

FOTO AFRATI afrati@cs.ece.ntua.gr
TIMOS ASLANIDIS
NTUA, Department of Electrical and Computer Engineering, Athens, Greece

EVRIPIDIS BAMPIS
LaMI, CNRS UMR 8042, Université d'Evry Val d'Essonne, France

IOANNIS MILIS
Department of Informatics, Athens University of Business and Economics, Greece

Received July 17, 2002; Revised December 30, 2002; Accepted December 31, 2002

Abstract. We consider the (*preemptive bipartite scheduling problem PBS*) (Crescenzi et al., "On approximating a scheduling problem," *Journal of Combinatorial Optimization*, vol. 5, pp. 287–297, 2001) arising in switching communication systems, where each input and output port can be involved in at most one communication at the same time. Given a set of communication tasks to be communicated from the transmitters to the receivers of such a system, we aim to find a schedule minimizing the overall transmission time. To achieve this, we allow the preemption of communication tasks. However, in practice preemption comes with a cost, d, and this renders the problem NP-hard (Gopal et al., "An optimal switching algorithm for multibeam satellite systems with variable bandwidth beams," *IEEE Trans. Commun.*, vol. 30, pp. 2475–2481, 1982). In this paper, we present a $2 - \frac{1}{d+1}$ approximation algorithm, which is the first one for the PBS problem with approximation ratio strictly less than two. Furthermore, we propose a simple optimal polynomial time algorithm for a subclass of instances of the PBS problem.

Keywords: switching networks, scheduling, set-up delays, approximation algorithms

1. Introduction

Switching networks are widely used in many communication (Goudreau et al., 1996) and multiprocessor systems (Gopal et al., 1982). In such a context, a set of communication devices communicate through the switching network by exchanging messages. A switching network is composed by a set of transmitters (input ports), a set of receivers (output ports), and a set of properly connected 2×2 switching elements. A transmitter (resp. receiver) cannot transmit (resp. receive) more than one messages at a time. Of course, the transmission of messages between different transmitters and receivers can take place simultaneously. The

*This work has been partially supported by the the European Union (FET-Working Group APPOL II), and the Greek General Secretariat of Research and Technology.

objective is to minimize the *total transmission time*. The problem can be formalized by a *weighted bipartite graph* whose bipartition classes correspond to the transmitters, and the receivers. Messages (communication tasks) are represented by the (weighted) edges of this bipartite graph. A *parallel transmission step* is a communication phase during which we have simultaneous transmissions between several transmitters and receivers, and corresponds to a *matching* in the bipartite graph. A usual approach in order to minimize the overall transmission time is to allow *preemption*, i.e. the possibility to interrupt the transmission of a message and complete it later. In graph-theoretic terms, each edge can be subdivided into many edges adding up to the edge's original weight. However in practice, there is a non negligible *cost* for the set-up of a new parallel transmission step referred to *set-up delay*. Therefore, the total transmission time for a given traffic pattern will be equal to the schedule length of the (eventually) preempted messages plus the number of parallel transmission steps (matchings) multiplied by the set-up delay. This problem has been studied in the literature, and it is known as the *Preemptive Bipartite Scheduling* (PBS) problem (Crescenzi et al., 2001).

1.1. Formulation of the problem

The problem can be formulated in terms of a weighted bipartite graph $G = (V \cup U, E, w)$, where the sets V and U represent the senders and the receivers respectively. The set of edges E represent the communication tasks and a positive integer weight $w(e)$ associated to any edge $e \in E$ represents the transmission time required by the corresponding communication task. The set-up delay for a transmission will be denoted by d.

In this context the Preemptive Bipartite Scheduling (PBS) problem can be stated as follows:

Find a collection $\{M_1, M_2, \ldots, M_s\}$ of matchings of G and corresponding weights $w(M_1), w(M_2), \ldots, w(M_s)$ (recall that an edge e can be split among several matchings and let $w_k(e)$ be its portion belonging to matching M_k) such that:

(i) $\sum_{k=1}^{s} w_k(e) = w(e)$, for each $e \in E$, and
(ii) $\sum_{k=1}^{s} w(M_k) + s \cdot d$ is minimized, where $w(M_k) = \max\{w_k(e) \mid e \in M_k\}$.

An alternative way to represent the input as well as the solution is by using a traffic matrix $A = (a_{ij})$, where every element a_{ij} is equal to the transmission time from sender i to receiver j. Our goal is to find a collection of switching matrices (i.e. with at most one non zero entry in each row and in each column) $A_1, A_2, \ldots, A_s$, such that (i) $A_1 + A_2 + \cdots + A_s = A$, and (ii) $\sum_{k=1}^{s} w(A_k) + s \cdot d$ is minimized, where $w(A_k) = \max\{a_{ij} \mid a_{ij} \in A_k\}$.

1.2. Previous results

PBS was proven to be NP-hard for the very restricted class of bipartite graphs where the weights appearing are among the numbers 1, 2 or 3 and even for graphs where the maximum degree of any node is less than 4 (Even et al., 1976; Gopal et al., 1982). Furthermore approximating the problem by a factor less than 7/6 is impossible, unless $P = NP$ (Crescenzi et al., 2001). Up to our knowledge, the best performance ratio found so far for PBS is equal to two (Crescenzi et al., 2001). Indeed, the authors of Crescenzi et al. (2001) have proposed two different polynomial time approximation algorithms proving that both have a performance ratio equal to two (however one of them behaves better in practice (Crescenzi et al., 2001)). In the case where all the weights of the edges are equal the problem is already proven to be solvable in polynomial time since it is equivalent to the bipartite edge coloring problem. In this paper, we propose a polynomial-time algorithm for the PBS problem with performance guarantee $2 - \frac{1}{d+1}$, which is strictly less than 2. The performance ratio of the algorithm tends asymptotically to 2 when d goes to infinity, but it gives good results for small values of d (e.g. $\frac{3}{2}$ for $d = 1$, $\frac{4}{3}$ for $d = 2$).

2. Approximation algorithms

2.1. Preliminary definitions

We define the weight $w(u)$ of a node u of G to be the sum of the weights of all edges incident to node u, i.e. $w(u) = \sum_{(u,v)\in E} w(u, v)$. By $W(G)$ we denote the maximum $w(u)$ over all vertices, i.e. $W(G) = \max\{w(u) \mid u \in V \cup U\}$. By $\Delta(G)$ we denote the maximum degree of G, i.e. $\Delta(G) = \max\{d(v) \mid v \in V \cup U\}$. A straightforward lower bound for the solution of PBS is $W(G) + d \cdot \Delta(G)$. This is because the sum of weights $w(M_k)$ of all the matchings must cover the largest node weight of the graph, i.e. $\sum_{k=1}^{s} w(M_k) \geq W(G)$. Furthermore, there must be at least as many matchings as the largest number of edges incident to a node, i.e. $s \geq \Delta(G)$.

2.2. The algorithm A-PBS(α)

The algorithm is in three steps:

1. Round up the weight of every edge of the initial graph $G = (V \cup U, E, w)$ to a multiple of a given value α. Call the obtained graph $G' = (V \cup U, E, w')$.
2. Split every edge e_{ij} of G' into $w'(e_{ij})/\alpha$ edges having each a weight equal to α. Call the induced graph G_α.
3. Find exactly $\frac{W(G_\alpha)}{\alpha}$ matchings in G_α (covering all edges of G_α).

Notice that the weights in G_α are all equal to α.

Example 1. Consider the graph having the following traffic matrix A, with $d = 1$.

$$A = \begin{array}{|c|c|c|}\hline 3 & 7 & 1 \\ \hline 5 & 0 & 3 \\ \hline 1 & 2 & 5 \\ \hline \end{array}$$

For $\alpha = 2$, A-PBS(2) gives rise, in its first step, to the following matrix A' corresponding to G':

$$A' = \begin{array}{|c|c|c|}\hline 4 & 8 & 2 \\ \hline 6 & 0 & 4 \\ \hline 2 & 2 & 6 \\ \hline\end{array}$$

A solution obtained by A-PBS(2) is the following one:

$$A' = \begin{array}{|c|c|c|}\hline 0 & 2 & 0 \\ \hline 2 & 0 & 0 \\ \hline 0 & 0 & 2 \\ \hline\end{array} + \begin{array}{|c|c|c|}\hline 0 & 2 & 0 \\ \hline 2 & 0 & 0 \\ \hline 0 & 0 & 2 \\ \hline\end{array} + \begin{array}{|c|c|c|}\hline 0 & 2 & 0 \\ \hline 0 & 0 & 2 \\ \hline 0 & 0 & 0 \\ \hline\end{array} + \begin{array}{|c|c|c|}\hline 2 & 0 & 0 \\ \hline 0 & 0 & 2 \\ \hline 0 & 2 & 0 \\ \hline\end{array} + \begin{array}{|c|c|c|}\hline 2 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 2 \\ \hline\end{array} + \begin{array}{|c|c|c|}\hline 0 & 2 & 0 \\ \hline 2 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline\end{array} + \begin{array}{|c|c|c|}\hline 0 & 0 & 2 \\ \hline 0 & 0 & 0 \\ \hline 2 & 0 & 0 \\ \hline\end{array}$$

The cost of this solution $2 \cdot 7 + 1 \cdot 7 = 21$, while the lower bound for the original graph G was $11 + 3 = 14$. An optimal solution for this instance is given below:

$$A = \begin{array}{|c|c|c|}\hline 3 & 0 & 0 \\ \hline 0 & 0 & 3 \\ \hline 0 & 2 & 0 \\ \hline\end{array} + \begin{array}{|c|c|c|}\hline 0 & 7 & 0 \\ \hline 5 & 0 & 0 \\ \hline 0 & 0 & 5 \\ \hline\end{array} + \begin{array}{|c|c|c|}\hline 0 & 0 & 1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 0 & 0 \\ \hline\end{array}$$

The cost of this solution is $(3 + 7 + 1) + 1 \cdot 3 = 14$ and it is optimal since it achieves the lower bound.

Theorem 1. *Algorithm A-PBS($d + 1$) provides a solution for PBS within $2 - \frac{1}{d+1}$ of the optimal.*

Proof: For G_α the weight of each matching is α, therefore $\frac{W(G_\alpha)}{\alpha}$ matchings have a total cost

$$\frac{W(G_\alpha)}{\alpha} \cdot \alpha + \frac{W(G_\alpha)}{\alpha} \cdot d$$

(for both the transmission and the total set-up delay).

But $W(G_\alpha) \leq W(G) + (\alpha - 1)\Delta(G)$, and thus the total cost of A-PBS(α) is bounded above by:

$$\begin{aligned} & W(G) + (\alpha - 1)\Delta(G) + \frac{W(G) + (\alpha - 1)\Delta(G)}{\alpha} \cdot d \\ &= \frac{d + \alpha}{\alpha} \cdot (W(G) + (\alpha - 1))\Delta(G)) \end{aligned} \tag{1}$$

Assuming that $\alpha = d + 1$, we get by (1) that the total cost of the solution obtained by A-PBS($d + 1$) is at most:

$$\frac{2d + 1}{d + 1} \cdot (W(G) + d\Delta(G)).$$

Given now that a lower bound of the total cost of any algorithm is given by $W(G) + d\Delta(G)$, we get that the performance ratio of A-PBS($d+1$) is bounded above by $\frac{2d+1}{d+1} = 2 - \frac{1}{d+1}$, as stated above. □

Unfortunately, Algorithm A-PBS(α) is a *pseudo-polynomial* one as its running time is linear in $W(G_\alpha)$ (the algorithm finds $\frac{W(G_\alpha)}{\alpha}$ matchings in G_α). Next proposition turns A-PBS(α) into a polynomial one by applying a technique which have been also used in Bongiovanni et al. (1981) and Crescenzi et al. (2001).

Proposition 1. *There exists a* $(2 - \frac{1}{d+1})$ *polynomial-time approximation algorithm for PBS.*

Proof: We add new edges and nodes to G' in order to produce a weight regular graph $\bar{G}'$ where the total weight of edges incident to any node is equal to $W(G')$. Then, the algorithm is as follows: Find a perfect matching M in the induced graph $\bar{G}'$ and set $w(M) = \min\{w'(e) \mid e \in M\}$. For every edge $e \in M$, we reduce its weight by $w(M)$, and we remove the edges with weight 0. The obtained graph is still weight-regular and we repeat the same procedure until all the edges to be removed. The complexity of this algorithm is clearly $O(|E|)$, since at each step at least one of the edges of G' is completely removed. Thus, the graph G' will be partitioned in at most $|E|$ matchings each one containing at least one whole edge. Since the weight of each edge of the graph G' is a multiply of $d+1$, any output of this algorithm can be transformed to an output of Algorithm A-PBS($d+1$) of greater or equal cost. Therefore, the approximation factor of $(2 - \frac{1}{d+1})$ still holds. □

Notice that Proposition 1, besides of reducing the number of iterations, reduces the number of preemptions as well, giving a solution that may be closer to the optimal than the solution obtained by A-PBS($d+1$).

Remark 1. The upper bound established by the previous theorem is tight. Indeed, consider an instance of the problem where the graph G is just a matching where every edge has a weight equal to one. A-PBS($d+1$) will have a cost of $2d+1$ while the optimum is $d+1$.

2.3. *The choice of* α

A natural question arises: *is it possible to get an improvement on the performance ratio of A-PBS(α) by choosing a different value of α?*

The following proposition shows that this is not possible.

Proposition 2. *The best performance ratio that we can expect from A-PBS(α) is* $2 - \frac{1}{d+1}$, *i.e. there is no α turning the performance ratio of the algorithm smaller than* $2 - \frac{1}{d+1}$.

Proof: By (1), in the proof of Theorem 1, we know that the total cost of A-PBS(α) is bounded above by $\frac{d+\alpha}{\alpha} \cdot (W(G) + \alpha - 1)\Delta(G)$. Given that the lower bound is independent from α, we get that the minimization of the performance ratio will be obtained for a value of α that minimizes the total cost. Elementary computations give that this minimization is obtained for $\alpha = \sqrt{\frac{d(W(G)-\Delta(G))}{\Delta(G)}}$. For this value of α, the total cost becomes $(\sqrt{W(G)-\Delta(G)} + \sqrt{d\Delta(G)})^2$, and thus the performance ratio is given by the following function:

$$\frac{(\sqrt{W(G)-\Delta(G)} + \sqrt{d\Delta(G)})^2}{W(G)+d\Delta(G)} = 1 + \frac{2\sqrt{(W(G)-\Delta(G))d\Delta(G)} - \Delta(G)}{W(G)+d\Delta(G)}$$

where $W(G) \geq \Delta(G) \geq 1$ and $d \geq 1$. What we need now, in order to prove the claim of the proposition, is to find the maximum value of the function $f = \frac{2\sqrt{(W(G)-\Delta(G))d\Delta(G)}-\Delta(G)}{W(G)+d\Delta(G)}$. This task is simplified by observing that f is an homogeneous function of degree 0 with respect to $\Delta(G)$ and $W(G)$. By substituting $W(G)$ by $\Delta(G)(1+dx^2)$, with $x > 0$, we have to maximize the expression $\frac{2dx-1}{1+d+dx^2}$. If d is fixed then the maximum[1] is obtained for $x = \frac{d+1}{d}$, and it is $1 - \frac{1}{d+1}$. Thus the best performance ratio of A-PBS(α), for any α, is $2 - \frac{1}{d+1}$. □

2.4. *A special case: Weights multiples of a same value*

It is interesting to point out that the performance ratio of A-PBS becomes more favorable for instances with weights satisfying the following property: after dividing the weights by d, every edge, say e, has a weight of the form $w(e) = b(e)\cdot m,\ b(e),\ m \in \mathbf{N}$. In other words, all edges have weights which are multiples of the same value m, and $d = 1$. The problem remains $\mathcal{NP}$-hard even in this special case since otherwise the general case of integer weights would also be polynomial (divide all edge weights by m). A-PBS(m) produces an $(m+1)/m$-approximation algorithm.

Theorem 2. *If the normalized by d weights are of the form* $w(e) = b(e)\cdot m,\ b(e),\ m \in \mathbf{N}$, *then A-PBS*($m$) *yields a solution within* $(m+1)/m$ *times the optimal.*

Proof: It is clear that $W(G_\alpha) = W(G_m) = W(G)$ and $\Delta(G_\alpha) = \Delta(G_m) = W(G)/m$. The total cost for the solution provided by the algorithm is

$$\begin{aligned} W(G_\alpha) + \Delta(G_\alpha) &= W(G_m) + W(G_m)/m = \frac{m+1}{m} W(G_m) \\ &= \frac{m+1}{m} W(G) \leq \frac{m+1}{m}(W(G) + \Delta(G)). \end{aligned}$$

Given now that after normalization we have $d = 1$, the lower bound for any algorithm is $W(G) + \Delta(G)$. Hence, A-PBS(m) is an $(m+1)/m$-approximation algorithm. □

Notice, that for $m = 1$ the performance ratio is equal to 2, but as m gets bigger, the performance ratio is improved.

3. A class of graphs with polynomial solution to the problem

In this section we present a polynomial time algorithm for the PBS problem when restricted to a subclass of weighted bipartite graphs, that we call *unvarying* weighted bipartite graphs.

Definition. A weighted bipartite graph will be called *unvarying* if the number of edges of any specific weight w incident to any node u is less than or equal to the number of edges of the same weight, which are incident to the node with the maximum degree.

Example 2. Consider the graphs with input matrices A and B:

$$A = \begin{array}{|c|c|c|}\hline 3 & 3 & 1 \\ \hline 0 & 3 & 0 \\ \hline 1 & 0 & 0 \\ \hline \end{array} \qquad B = \begin{array}{|c|c|c|}\hline 3 & 3 & 1 \\ \hline 3 & 2 & 2 \\ \hline 1 & 0 & 0 \\ \hline \end{array}$$

A corresponds to an unvarying graph as the maximum number of edges having a weight equal to 3 is 2 (the node corresponding to the first line is of maximum degree) and no row or column has more than two such edges, while the number of edges with weight equal to 1 does not in any case exceed one (the number of edges with weight 1 for the first row).

B corresponds to a graph that is not unvarying. The number of 2's in row 2 exceeds the number of 2's in row 1 and the number of ones in row 1 exceeds the number of ones in row 2 (notice that both rows 1 and 2 correspond to the nodes with the maximum degree).

Algorithm A-PBS-UN:

1. Split the unvarying graph in $|W|$ graphs, each having edges of the same weight, where $|W|$ is the number of different edge weights appearing in G. Call these graphs G_1, $G_2, \ldots, G_{|W|}$.
2. Find exactly $\Delta(G_i)$ matchings in each graph G_i so that the union of these matchings covers all edges of G_i.
3. The union of the matchings found for the $|W|$ graphs is an optimal solution for PBS.

Theorem 3. *Algorithm A-PBS-UN produces the optimal solution for unvarying weighted bipartite graphs.*

Proof: Suppose the edge weights are numbers in the range $1, 2, \ldots, a$. Let G be the unvarying graph and $G_1, \ldots, G_a$ be the graphs described in Step 1 (some of the G_i's may be $\emptyset$).

Suppose that u is a node of maximum degree. From the fact that G is unvarying, we get that there are exactly $\Delta(G_1)$ edges with weight one, $\Delta(G_2)$ edges with weight two, $\ldots$, $\Delta(G_a)$ edges with weight a, incident to node u. ($\Delta(G_i)$ may be zero for some i.)

Therefore $\Delta(G_1) + \Delta(G_2) + \cdots + \Delta(G_a) = \Delta(G)$ and $w(u) = \Delta(G_1) + 2\Delta(G_2) + \cdots + a\Delta(G_a)$.

Consequently, W(G)$\geq \Delta(G_1) + 2\Delta(G_2) + \cdots + a\Delta(G_a)$

The cost for each of the graphs is $d\Delta(G_k) + k\Delta(G_k)$, $k = 1, 2, \ldots, a$; $\Delta(G_k)$ for the number of matchings and $k\Delta(G_k)$ for the transmission time. Consequently the cost for all subgraphs is

$$\begin{aligned}
&((d\Delta(G_1) + \Delta(G_1)) + (d\Delta(G_2) + 2\Delta(G_2)) + \cdots + (d\Delta(G_a) + a\Delta(G_a)) \\
&\quad = d(\Delta(G_1) + \Delta(G_2) + \cdots + \Delta(G_a)) + \Delta(G_1) + 2\Delta(G_2) + \cdots + a\Delta(G_a) \\
&\quad = d\Delta(G) + (\Delta(G_1) + 2\Delta(G_2) + \cdots + a\Delta(G_a)) \\
&\quad \leq d\Delta(G) + W(G)
\end{aligned}$$

It is then clear that the algorithm is optimal since the lower bound is exactly $d\Delta(G) + W(G)$. □

Acknowledgments

We would like to thank Charles Delorme for helpful discussions.

Note

1. Details of the computation for the maximum of $\frac{2dx-1}{1+d+dx^2}$:

$$\begin{aligned}
&f' = \frac{2d(1 + d + x - dx^2)}{(1 + d + dx^2)^2} \\
&f' = 0 \text{ gives } x = -1 \text{ and } x = \frac{1+d}{d} \\
&f'' = \frac{2d(1 + d(1 - 6x - 3x^2) + 2d^2x(x^2 - 3))}{(1 + d + dx^2)^3} \\
&f''(x = -1) = \frac{2d(1 + 4d + 4d^2)}{(1 + 2d)^3} > 0, \text{ minimum} \\
&f''\left(x = \frac{1+d}{d}\right) = -\frac{2d^3}{(1 + d)^2(1 + 2d)} < 0, \text{ maximum} \\
&f\left(x = \frac{1+d}{d}\right) = \frac{d}{d+1}
\end{aligned}$$

References

G. Bongiovanni, D. Coppersmith, and C.K. Wong, "An optimal time slot assignment for an SS/TDMA system with variable number of transponders," *IEEE Trans. Commun.*, vol. 29, pp. 721–726, 1981.

P. Crescenzi, X. Deng, and C.H. Papadimitriou, "On approximating a scheduling problem," *Journal of Combinatorial Optimization*, vol. 5, pp. 287–297, 2001.

S. Even, A. Itai, and A. Shamir, "On the complexity of timetable and multicommodity flow problems," *SIAM J. Comput.*, vol. 5, pp. 691–703, 1976.

I.S. Gopal, G. Bongiovanni, M.A. Bonucelli, D.T. Tang, and C.K. Wong, "An optimal switching algorithm for multibeam satellite systems with variable bandwidth beams," *IEEE Trans. Commun*., vol. 30, pp. 2475–2481, 1982.

I.S. Gopal and C.K. Wong, "Minimizing the number of switchings in an SS/TDMA system," *IEEE Trans. Commun.*, vol. 33, pp. 497–501, 1985.

M. Goudreau, K. Lang, S.B. Rao, T. Suel, and T. Tsantilas, "Towards efficiency and portability: Programming with the BSP model," in *Proc. SPAA*, 1996.

T. Inukai, "An efficient SS/TDMA time slot assignment algorithm," *IEEE Trans. Commun.*, vol. 27, pp. 1449–1455, 1979.

K.S. Natarajan and S.B. Calo, "Time slot assignment in an SS/TDMA system with minimum switchings," *IBM Res. Rep.*, 1981.

L.G. Valiant, "A bridging model for parallel computation," *Comm. ACM*, vol. 33, pp. 103–111, 1990.