

Chapter 11

Applications of Preemptive Open Shop Scheduling



11.1 Introduction

Before we present more applications of open shop scheduling to real-life problems and scheduling theory let us briefly summarize the applications discussed in previous chapters of this book. Chapter 4 presents applications to University timetabling, Chap. 6 presents application to scheduling wireless networks with primary interference, Chap. 5 presents applications to product design and scheduling customer orders among other applications of open shop scheduling where operations of a job can be processed concurrently, Chaps. 7 and 8 present application to scheduling large oncology centers, Chap. 9 presents application to timetabling and just-in-time scheduling, and Chap. 10 presents applications to scheduling without intermediate storage and optical networks. Other applications are also listed in Ahmadian et al. [1].

This chapter introduces further applications of open shop scheduling. Those include satellite-switched time-division multiple access method used to allocate the communication bandwidth provided by a satellite link to carry traffic between earth stations; scheduling reconfigurable data centers; file transfer; scheduling crossbar switches to guarantee 100% throughput for traffic with given rates; multimessage unicasting and multicasting; scheduling and bandwidth allocation problem; and scheduling theory. Majority of those applications permit preemptions which allow for polynomial-time algorithms when it comes to makespan minimization. Details will be described in the following sections.

11.2 Satellite Communication

Satellite communication systems have been built to connect a large number of earth stations. The *satellite-switched time-division multiple access* (SS/TDMA) method has been used to allocate the communication bandwidth provided by a satellite link to carry traffic between earth stations, see Inukai [30], and Lewandowski and Liu [35] for further references to the SS/TDMA method. In a nutshell, a satellite equipped with spot-beam antennas, and a switch periodically switches the connections between uplink beams and downlink beams to connect beam zones. A TDMA frame is a sequence of *time slots* of different durations. In each of the time slots, a different *switching mode* determines a specific set of interconnections so that the traffic between zones is routed without conflicts. The SS/TDMA time slot assignment problem instance is defined by an $n \times n$ non-negative *traffic matrix* \mathcal{T} where t_{ij} is the amount of traffic which is to be routed from the uplink beam i to the downlink beam j , and n is the number of spot-beams. The quantities t_{ij} may be expressed in terms of some basic traffic units such as a number of $T1$ channels, Inukai [30], or they can be the amount of time, Inukai [30], and Dell'Amico and Martello [18]. We let t_{ij} to be the amount of time in our discussion. The problem is to find a *switching mode matrix* S_i , and a time duration t_i for each time slot so as to meet the traffic demand defined by \mathcal{T} within the shortest possible TDMA frame. A switching mode matrix is a square $n \times n$ matrix where *at most* one entry in each row and column is equal to 1, and all other entries are equal to 0. To find a solution to the time slot assignment problem let us have a closer look at the traffic matrix \mathcal{T} . The total demand for its row i is equal to $R_i = \sum_j t_{ij}$ which is the amount of time required by the uplink beam i to transmit information to all downlink beams, and the total demand for its column j is equal to $C_j = \sum_i t_{ij}$ which is the amount of time required by the downlink beam j to receive information from all uplink beams. Clearly, the TDMA frame cannot be shorter than

$$\alpha = \max\{\max_i\{R_i\}, \max_j\{C_j\}\}.$$

Due to Birkhoff–von Neumann theorem we can get the frame of length α as follows. We take the following $(2n) \times (2n)$ matrix:

$$\mathcal{D} = \frac{1}{\alpha} \begin{bmatrix} \mathcal{T} & A \\ B & \mathcal{T}^T \end{bmatrix},$$

where the diagonal matrix

$$A = \begin{bmatrix} \alpha - R_1 & 0 & \dots & 0 \\ 0 & \alpha - R_2 & \dots & 0 \\ 0 & \dots & & 0 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \dots & \alpha - R_n \end{bmatrix}$$

complements each row sum of D to α , and the diagonal matrix

$$B = \begin{bmatrix} \alpha - C_1 & 0 & \dots & 0 \\ 0 & \alpha - C_2 & \dots & 0 \\ 0 & \dots & & 0 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \dots & \alpha - C_n \end{bmatrix}$$

complements each column sum of D to α . By the theorem there are $(2n) \times (2n)$ permutation matrices P_1, \dots, P_q and positive numbers t_1, \dots, t_q such that $t_1 + \dots + t_q = 1$ and

$$D = t_1 P_1 + \dots + t_q P_q. \tag{11.1}$$

Let S_i be the submatrix of the permutation matrix P_i lying in the rows $1, \dots, n$ and the columns $1, \dots, n$ of $P_i, i = 1, \dots, q$. Clearly, the matrices S_i are switching modes and

$$\mathcal{T} = \alpha(t_1 S_1 + \dots + t_q S_q).$$

Thus there are q time slots with durations $\alpha t_1, \dots, \alpha t_q$. Since $t_1 + \dots + t_q = 1$ the total duration is equal to α . The number of permutation matrices $q \leq n^2 - n + 1$, see Horn and Johnson [29]. The decomposition in (11.1) is not unique. Gonzalez and Sahni [26] would view the matrix \mathcal{T} as an instance of an open shop problem with preemptions and makespan minimization and solve the problem to optimality by their algorithm running in time $O(r(r + n \log n))$ where r is the number of positive entries in \mathcal{T} . In their solution a matching of jobs with machines would correspond to a switching mode, and a TDMA frame would correspond to an open shop preemptive schedule. Interestingly enough the algorithm of Gonzalez and Sahni preceded this of Inukai [30]. The latter seems to have been designed exclusively with the SS/TDMA time slot assignment in mind. The reader is referred to Dell’Amico and Martello [18] for detailed comparison of the algorithms by Gonzales and Sahni [26], and Inukai [30] in a broader historical context.

Inukai [30] emphasizes reduction of the number q as an important secondary criterion for the SS/TDMA time slot assignment problem. Clearly, the secondary criterion naturally carries over to the open shop with preemptions problem, $O|pmtn|C_{\max}$. A well known upper bound on q is $n^2 - n + 1$, see Horn and Johnson [29]. The algorithm of Inukai reduces that bound to $n^2 - 2n + 2$. Farahat and Mirsky [23] prove that this bound is the best possible. Brualdi [4], and Brualdi and Gibson [5] provide further analysis of the Birkhoff algorithm for doubly stochastic matrices. The minimization of q is NP-hard in the strong sense which was recently proved by Dufossé and Uçar [22]. Kulkarni et al. [33] further prove that the minimization of q is not fixed parameter tractable, and show polynomial-time algorithm for $q = 2, 3$. Dufossé et al. [21] show that a family of heuristics based on the original proof of Birkhoff can miss optimal decompositions which was earlier conjectured by Brualdi [4].

Lewandowski and Liu [35] extend the TDMA time slot assignment problem to take into account the ability to demultiplex each of the uplink beams into at most k signals, transmit each of these signals to different downlink beams, and then multiplex up to k of these signals into each of the downlink beams. For the open shop this means that at most k operations of each job can be processed at a time, and at most k jobs can be processed by each machine at a time. Observe that the single-operation machine model studied in Sect. 8.1 permits each stage to be processed several jobs at a time by allowing parallel identical machines in each stage, yet it still requires at most one operation of each job to be processed at a time. The concurrent open shops in Chap. 4 on the other hand allow all operations of each job to be processed in parallel. We have the following lower bound for the makespan of the frame:

$$\alpha = \max \left\{ \max_{i,j} t_{ij}, \max_i \frac{1}{k} R_i, \max_j \frac{1}{k} C_j \right\},$$

see Lewandowski and Liu [35]. de Werra [13] gives the following decomposition of the traffic matrix \mathcal{T} :

$$\mathcal{T} = t_1 Z_1 + \dots + t_q Z_q,$$

where each matrix Z_1, \dots, Z_q is a 0, 1 matrix with at most k 1's in each row, and at most k 1's in each column, and importantly

$$t_1 + \dots + t_q = \alpha.$$

The decomposition can be computed in polynomial time using network flow algorithms, see de Werra [10, 11, 13].

11.3 Reconfigurable Data Centers

Reconfigurable data centers and software defined networks connect servers within a data center by optical connections, Chen et al. [8]. An advantage of such network technology is that as traffic between servers changes over time, the network topology can be reconfigured to better match the change and to prevent localized bottlenecks. A traffic matrix in a reconfigurable data center represents traffic that needs to be routed among a set of n servers. A route is a matching between senders and receivers. The change between routes however requires moving laser pointers and receivers that comes at a cost, Kulkarni et al. [33]. Therefore, finding a shortest schedule for a given traffic matrix with as few as possible preemptions improves performance of the routing algorithms.

11.4 Crossbar Switches

An input-buffered crossbar switch with n input ports and n output ports has a buffer for each input port. In such a switch, time is slotted and synchronized so that packets from different input buffers can be read out simultaneously within a time slot. In a time slot, a crossbar switch sets up a connection pattern corresponding to a permutation matrix. As a permutation matrix is a one-to-one mapping from input ports to output ports, packets destined to the same output ports cannot be transmitted at the same time, Chang et al. [7]. A crossbar switch scheduling algorithm needs to guarantee 100% throughput for traffic with given rates $r_{i,j}$ from input i to output j . The rates are specified by an $n \times n$ traffic rate matrix

$$\mathbb{R} = \begin{bmatrix} r_{1,1} & \dots & r_{1,n} \\ r_{2,1} & \dots & r_{2,n} \\ \cdot & \dots & \cdot \\ \cdot & r_{i,j} & \cdot \\ \cdot & \dots & \cdot \\ r_{n,1} & \dots & r_{n,n} \end{bmatrix}.$$

The traffic rate matrix needs to be substochastic, total of each row and column must not exceed 1, Horn and Johnson [29], in order for 100% throughput to be achieved at all. Chang et al. [7] propose a scheduling algorithm that guarantees 100% throughput for substochastic matrices. The algorithm first finds a convex combination of permutation matrices for a substochastic \mathbb{R} . This stage is based on Birkhoff–von Neumann theorem; however, algorithms for makespan minimization for open shop with preemptions like the one by Gonzales and Sahní [26] can also be used for this stage. The convex combination of permutation matrices is then used by fair queueing algorithms, see Demers et al. [19] and Chapter 10 in Kubiak [32],

to switch between permutation matrices to guarantee 100% throughput for traffic defined by the traffic rate matrix \mathbb{R} .

11.5 Multimessage Unicasting and Multicasting

Gonzalez [27] considers multimessage unicasting and multicasting problem where processors communicate by sending and receiving messages over a fully connected network. Each processor can send or receive a message; however, no processor may send more than one message at a time, and no processor may receive more than one message at a time. There are $t_{i,j}$ messages to be sent from processor i to processor j . The problem is to find a shortest possible communication schedule for all messages to be transmitted. This problem reduces to a preemptive open shop where each processor i in the communication network represents a job J_i and a machine M_i , and the processing time of operation $O_{i,h}$ equals $t_{i,h}$, Gonzalez [27] and [25]. The multimessage unicasting is a special case of multimessage multicasting where the same message needs to be sent to many processors. The algorithms for the latter transform the multicasting problem into a unicasting problem and apply the open shop algorithms to solve the unicasting problem, see Gonzalez [28].

11.6 Scheduling and Wavelength Assignment problem

Bampis and Rouskas [2] study the following scheduling and wavelength assignment problem. Consider sources $s = 1, \dots, n$ and destinations $d = 1, \dots, n$ and an $n \times n$ matrix P where $p_{s,d}$ is the number of packets to be transmitted from s to d . The transmission happens at a certain wavelength assigned to the destination node d . Typically there are fewer wavelengths, m , than destination nodes, thus several destination nodes may be assigned to the same wavelength. Hence a wavelength is a machine. For a given assignment of destination nodes to machines, we obtain an open shop where a source node s is a job made up of m operations corresponding to wavelengths. The operation of job s on machine h is made up of all destinations d assigned to the same wavelength h . Thus s transmits at wavelength h to all destinations d with that wavelength. If it does so, it cannot use any other wavelength machine at the same time (at most one operation of a job can be processed at a time). To illustrate the problem consider the following source to destination matrix:

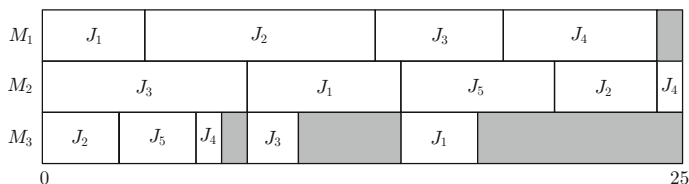


Fig. 11.1 A schedule and wavelength assignment

$$P = \begin{bmatrix} 2 & 1 & 3 & 2 & 5 \\ 3 & 3 & 3 & 6 & 1 \\ 1 & 5 & 2 & 4 & 3 \\ 4 & 0 & 1 & 2 & 1 \\ 0 & 2 & 3 & 0 & 4 \end{bmatrix} .$$

Assuming three wavelengths M_1 , M_2 , and M_3 and the assignment of destinations 1 and 4 to M_1 , 2 and 5 to M_2 , and 3 to M_3 we obtain a three-machine open shop

$$\mathbb{P} = \begin{bmatrix} 4 & 6 & 3 \\ 9 & 4 & 3 \\ 5 & 8 & 2 \\ 6 & 1 & 1 \\ 0 & 6 & 3 \end{bmatrix} .$$

The source 2 uses the wavelength M_1 to transmit 3 packets to $d = 1$ and 6 packets to $d = 4$, it uses the wavelength M_2 to transmit 3 packets to $d = 2$ and 1 packet to $d = 5$, and it uses the wavelength M_3 to transmit 3 packets to $d = 3$. This corresponds to job J_2 with operations having processing time 9 on M_1 , processing time 4 on M_2 , and processing time 3 on M_3 . A schedule for job J_2 and the remaining four jobs is given in Fig. 11.1.

11.7 Scheduling Theory

The application to scheduling theory have been observed in the study of the two-phase method for preemptive scheduling in de Werra [12, 14]. We illustrate this method using the scheduling on unrelated processors with preemptions to minimize makespan problem, $R|pmtn|C_{max}$. An instance of the problem is specified, like in the open shop problem, by an $n \times m$ non-negative real matrix

$$\mathbb{P} = \begin{bmatrix} p_{1,1} & \dots & p_{1,m} \\ p_{2,1} & \dots & p_{2,m} \\ \cdot & \dots & \cdot \\ \cdot & p_{i,h} & \cdot \\ \cdot & \dots & \cdot \\ p_{n,1} & \dots & p_{n,m} \end{bmatrix}.$$

However its entries have a different meaning than in the open shop scheduling problem. To explain the difference suppose for the time being that preemptions are not allowed. Then each job J_i is completely processed by one of the m machines. If that machine is machine M_h , then the job J_i processing time equals $p_{i,h}$. The remaining processing times of job J_i , or the remaining entries in row i , are then irrelevant for the solution. Now returning to the preemptive case let us take a convex combination $p_{i,1}x_{i,1} + \dots + p_{i,m}x_{i,m}$ of the processing times of job J_i , $i = 1, \dots, n$. For given n convex combinations, one for each job, we obtain the following instance of the open shop:

$$\mathbb{P}_x = \begin{bmatrix} p_{1,1}x_{1,1} & \dots & p_{1,m}x_{1,m} \\ p_{2,1}x_{2,1} & \dots & p_{2,m}x_{2,m} \\ \cdot & \dots & \cdot \\ \cdot & p_{i,h}x_{i,h} & \cdot \\ \cdot & \dots & \cdot \\ p_{n,1}x_{n,1} & \dots & p_{n,m}x_{n,m} \end{bmatrix},$$

where

$$\Delta(\mathbb{P}_x) = \left\{ \max_i \left\{ \sum_{h=1}^m p_{i,h}x_{i,h} \right\}, \max_h \left\{ \sum_{i=1}^n p_{i,h}x_{i,h} \right\} \right\}. \quad (11.2)$$

The solution to the instance \mathbb{P}_x of the open shop problem $O|pmtn|C_{\max}$ with makespan $C_{\max} = \Delta(\mathbb{P}_x)$ can be found in polynomial time, see Sect. 3.7. The solution solves $R|pmtn|C_{\max}$ by scheduling fraction $x_{i,h}$ of processing time $p_{i,h}$ of J_i on machine M_h . In other words job J_i has now m operations $O_{i,1}, \dots, O_{i,m}$ with processing times $p_{i,1}x_{i,1}, \dots, p_{i,m}x_{i,m}$, respectively. Thus to find an optimal solution to $R|pmtn|C_{\max}$ one needs to find the coefficients $x_{i,h}$ that minimize $\Delta(\mathbb{P}_x)$. This can be done by solving a linear program, see Lawler and Labetoulle [34], Brucker [6] and Błażewicz et al. [3]. A similar two-phase approach based on LP and reduction to open shop works for

- The problem $R|pmtn, r_i|L_{\max}$, see Lawler and Labetoulle [34], Brucker [6] and Błażewicz et al. [3].
- The problems in Sects. 4.3 and 4.10.
- The problem $O|pmtn, r_j|L_{\max}$ in Sect. 3.7.2.
- The multiprocessor open shop scheduling in Chap. 8.

11.8 Data Migration and File Transfer Scheduling

Data migration problem arises whenever large amount of data need to be rapidly transferred to new locations within storage area networks in order to better respond to changing demand for data. During the migration the network still needs to provide efficient though unlikely best service. Therefore it is important to compute as short as possible data migration schedule that produces the target data layout. Coffman et al. [9], see also Gandhi et al. [24] and Khuller et al. [31], introduce file transfer multigraph $G = (V, E)$ to model the problem. The vertices in V represent storage devices (computers or computer centers). The labeled edges correspond to the files to be transferred between the vertices. The label $p_{i,j}$ of an edge (i, j) represents transfer time required to transfer the file between i and j . Each vertex i completes transfers only after all edges incident with v in G complete transfers. The key constraint is that no two edges incident with the same vertex can be transferred at the same time. This constrain can be relaxed by increasing the maximum number of simultaneous file transfers to more than one by increasing the number of communication ports at vertices, see Coffman et al. [9]. The problem can be recast as open shop scheduling with simultaneity constraints, see de Werra [15] de Werra and Erschler [16] de Werra et al. [17], as it was shown in Chap. 6 for wireless networking with primary interference.

From a different angle, each vertex in G corresponds to machine in open shop. Each edge (i, j) labeled with $p_{i,j}$ is a bi-processor job with processing time $p_{i,j}$. The job needs to be processed on machines i and j simultaneously for exactly $p_{i,j}$ units of time. The problem becomes a subproblem of the problem $P|\text{fix } j|C_{\max}$ with $|\text{fix } j| = 2$, see Drozdowski [20], where each job is assigned a fixed subset of parallel machines to be processed on simultaneously.

Problems

11.1 Find more applications of preemptive open shop scheduling in scheduling theory, see Sect. 11.7.

References

1. M.M. Ahmadian, M. Khatami, A. Salehipour, T.C.E. Cheng, Four decates of research on the open-shop scheduling problem to minimize makespan. *Eur. J. Oper. Res.* (2021). <https://doi.org/10.1016/j.ejor.2021.03.026>
2. E. Bampis, G.N. Rouskas, The scheduling and wavelength assignment problem in optical wdm networks. *J. Lightwave Tech.* **20**, 782–789 (2002)
3. J. Błażewicz, K. Ecker, E. Pesch, G. Schmidt, J. Węglarz, *Handbook on Scheduling: From Theory to Applications*. International Handbooks on Information Systems (Springer, 2007)

4. R.A. Brualdi, Notes on the Birkhoff algorithm for doubly stochastic matrices. *Can. Math. Bull.* **25**, 127–147 (1982)
5. R.A. Brualdi, P.M. Gibson, Convex polyhedra of doubly stochastic matrices: I. applications of permanent function. *J. Comb. Theory A* **22**, 194–230 (1977)
6. P. Brucker, *Scheduling Algorithms* (Springer, 1995)
7. C.-S. Chang, W.-J. Chen, H.-Y. Huang, On Service Guarantees for Input Buffered Crossbar Switches: A Capacity Decomposition Approach by Birkhoff and von Neumann, in *Seventh International Workshop on Quality of Service* (IEEE, 1999), pp. 79–86
8. K. Chen, A. Singla, A. Singh, K. Ramachandran, L. Xu, Y. Zhang, X. Wen, Y. Chen, Osa: An optical switching architecture for data center networks with unprecedented flexibility. *IEEE/ACM Trans. Netw.* **22**, 498–511 (2014)
9. E.G. Coffman Jr., M.R. Garey, D.S. Johnson, A.S. LaPaugh, Scheduling file transfers. *SIAM J. Comput.* **14**, 744–780 (1985)
10. D. de Werra, A decomposition property of polyhedra. *Mathematical Programming* **30**, 261–266 (1984)
11. D. de Werra, Preemptive scheduling, linear programming and network flows. *SIAM J. Alg. Disc. Meth.* **5**, 11–20 (1984)
12. D. de Werra, On the two-phase method for preemptive scheduling. *Eur. J. Oper. Res.* **37**, 227–235 (1988)
13. D. de Werra, A note on SS/TDMA satellite communication. *Linear Algebra Appl.* **135**, 69–77 (1990)
14. D. de Werra, Almost nonpreemptive schedules. *Ann. Oper. Res.* **26**, 243–256 (1990)
15. D. de Werra, Extensions of coloring models for scheduling purposes. *Eur. J. Oper. Res.* **92**, 474–492 (1996)
16. D. de Werra, J. Erschler, Open shop with some additional constraints. *Graphs Comb.* **12**, 81–93 (1996)
17. D. de Werra, N.V.R. Mahadev, U. Peled, Edge chromatic scheduling with simultaneity constraints. *SIAM J. Disc. Math.*, 631–641 (1993)
18. M. Dell'Amico, S. Martello, Open shop, satellite communication and a theorem by Egerváry (1931). *Oper. Res. Lett.* **18**, 207–211 (1996)
19. A. Demers, S. Keshav, S. Shenkar, Analysis and symulation of a fair queueing algorithm. *Internetworking Res. Exp.* **1**, 3–26 (1990)
20. M. Drozdowski, Scheduling multiprocessor tasks - an overview. *Eur. J. Oper. Res.* **94**, 215–230 (1996)
21. F. Dufossé, K. Kaya, I. Panagiotas, B. Uçar, Further notes on Birkhoff-von Neumann decomposition od doubly stochastic matrices. *Linear Algebra Appl.* **554**, 68–78 (2018)
22. F. Dufossé, B. Uçar, Notes on Birkhoff-von Neumann decomposition of doubly stochastic matrices. *Linear Algebra Appl.* **479**, 108–115 (2016)
23. H.K. Farahat, L. Mirsky, Permutation endomorphisis and refinement of a theorem of Birkhoff. *Proc. Camb. Philos. Soc.* **56**, 322–328 (1960)
24. R. Gandhi, M.M. Halldórsson, G. Kortsarz, H. Shachnai, Improved results for data migration and open shop scheduling. *ACM Trans. Algorithms* **2**, 116–129 (2006)
25. T. Gonzalez, Open shop scheduling, in eds. by Joseph Y-T. Leung, *Handbook on Scheduling: Algorithms, Models, and Performance Analysis* (Chapman and Hall/CRC, 2004), pp. 6–1 to 6–14
26. T. Gonzalez, S. Sahni, Open shop scheduling to minimize finish time. *J. ACM* **23**, 665–679 (1976)
27. T.F. Gonzalez, Complexity and approximations for multimessage multicasting. *J. Par. Dist. Comput.* **55**, 215–235 (1998)
28. T.F. Gonzalez, Simple algorithms for multimessage multicasting with forwarding. *Algorithmica* **29**, 511–533 (2001)
29. R.A. Horn, C.R. Johnson, *Matrix Analysis*, 2nd edn. (Cambridge University Press, 2013)
30. T. Inukai, An efficient SS/TDMA time slot assignment algorithm. *IEEE Trans. Commun.* **27**, 1449–1455 (1979)

31. S. Khuller, Y. Kim, Y.C. Wan, Algorithms for data migration with cloning, in *Proc. of the 22nd ACM Symposium on Principles of Database Systems*, pp. 27–36 (2003)
32. W. Kubiak, *Proportional Optimization and Fairness* (Springer, 2009)
33. J. Kulkarni, E. Lee, M. Singh, Minimum Birkhoff-von Neumann decomposition, in *Proc. 19th International Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pp. 343–354, Waterloo, ON, Canada (2017)
34. E.L. Lawler, J. Labetoulle, On preemptive scheduling on unrelated parallel processors by linear programming. *J. ACM* **25**, 612–619 (1978)
35. J.L. Lewandowski, C.L. Liu, SS/TDMA satellite communications with k-permutation switching modes. *SIAM J. Alg. Disc. Meth.* **8**, 519–534 (1987)