Chapter 4
Multiprocessor Operations

4.1 Introduction

In the open shop scheduling with multiprocessor operations a set of jobs $\mathcal{J} = \{J_1, \ldots, J_n\}$ is scheduled on machines $\mathcal{M} = \{M_1, \ldots, M_m\}$. The set of machines is partitioned into p *disjoint* groups $\mathcal{G}_\ell$, $\ell = 1, \ldots, p$. Each job consists of *single-processor* and *multiprocessor* operations. A single-processor operation $O_{j,h}$ of job J_j requires a single machine $M_h \in \mathcal{M}$, and a multiprocessor operation $\hat{O}_{j,\ell}$ requires *all* machines from the group, $\mathcal{G}_\ell$, $\ell = 1, 2, \ldots, p$ simultaneously. The processing time of $O_{j,h}$ equals $b_{jh} \geq 0$, and the processing time of $\hat{O}_{j,\ell}$ equals $a_{j\ell} \geq 0$. In this chapter we depart from the notation $p_{j,h}$ (we use b_{jh} instead) introduced in Chap. 1 for processing time of operation $O_{j,h}$. This is to further emphasize the presence of individual and group operations in the open shop with multiprocessor operations. All processing times are integers for the time being. The processing time b_{jh} equals 0 means that J_j is missing on M_h, similarly the processing time $a_{j\ell}$ equals 0 means that J_j is missing on $\mathcal{G}_\ell$. In a feasible schedule each machine can process at most one operation at a time, and no two operations of the same job can be processed simultaneously. Any operation can be preempted at any moment and resumed at any moment later at no cost. The makespan is to be minimized.

An instance of the open shop scheduling with multiprocessor operations naturally decomposes into two instances of open shops. One referred to as the *group* open shop consists of p group machines $\mathcal{G}_1, \ldots, \mathcal{G}_p$, and n jobs in $\hat{\mathcal{J}}$, where the processing time of job $\hat{J}_j \in \hat{\mathcal{J}}$ on a group machine $\mathcal{G}_\ell$, $\ell = 1, \ldots, p$ equals $a_{j\ell}$. The other referred to as *individual* open shop consists of m machines $\mathcal{M} = \{M_1, \ldots, M_m\}$, and n jobs in $\mathcal{J}$, where the processing time of job $J_j \in \mathcal{J}$ on an individual machine $M_h \in \mathcal{M}$ equals $b_{j,h}$. For the group open shop machine $\mathcal{G}_\ell$ workload equals $\Delta(\mathcal{G}_\ell) = \sum_j a_{j\ell}$ for $\ell = 1, \ldots, p$, and job length equals $\Delta(\hat{J}_j) = \sum_\ell a_{j\ell}$ for $\hat{J}_j \in \hat{\mathcal{J}}$. Thus by König's edge-coloring theorem there is an

W. Kubiak, *A Book of Open Shop Scheduling*, International Series in Operations Research & Management Science 325,
https://doi.org/10.1007/978-3-030-91025-9_4

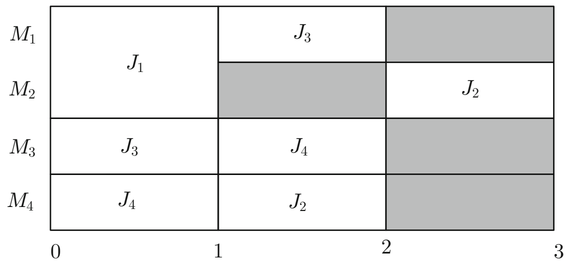


Fig. 4.1 An instance with $\Delta = 2$ and optimal schedule with $C_{\max} = 3$

optimal schedule $S_{\mathcal{G}}$ with makespan $\Delta(\mathcal{G}) = \max\{\max_\ell \Delta(\mathcal{G}_\ell), \max_j \Delta(\hat{J}_j)\}$ for the group open shop. For the individual open shop machine $M_h \in \mathcal{M}$ workload equals $\Delta(M_h) = \sum_j b_{jh}$, and job length equals $\Delta(J_j) = \sum_h b_{jh}$ for $J_j \in \mathcal{J}$. Thus again by König's edge-coloring theorem there is an optimal schedule $S_{\mathcal{M}}$ with makespan $\Delta(\mathcal{M}) = \max\{\max_h \Delta(M_h), \max_j \Delta(J_j)\}$ for the individual open shop. Both schedules, $S_{\mathcal{G}}$ and $S_{\mathcal{M}}$, respectively, can be obtained in polynomial time, see Gabow and Kariv [11] or Cole et al. [6]. Either schedule permits preemptions at integer points only and so does their concatenation $S_{\mathcal{G}}S_{\mathcal{M}}$. The makespan of the concatenation equals $\Delta(\mathcal{G}) + \Delta(\mathcal{M})$.

Now instead of looking at the two instances of the decomposition one at a time let us consider the original instance. The machine M_h workload equals $L_h = \sum_j a_{j\ell} + \sum_j b_{jh} = \Delta(\mathcal{G}_\ell) + \Delta(M_h)$, where $M_h \in \mathcal{G}_\ell$, and job J_j length $P_j = \sum_\ell a_{j\ell} + \sum_h b_{jh} = \Delta(\hat{J}_j) + \Delta(J_j)$. Therefore, $\Delta = \max\{\max_j P_j, \max_h L_h\}$ is a lower bound on the makespan of an optimal schedule. Since $\Delta \geq \Delta(\mathcal{G})$ and $\Delta \geq \Delta(\mathcal{M})$, the algorithm that gives the concatenation $S_{\mathcal{G}}S_{\mathcal{M}}$ is a 2-approximation algorithm for the makespan minimization of the open shop scheduling problem with multiprocessor operations. To illustrate consider the instance in Fig. 4.1, we have $p = 2$, $\Delta(\mathcal{G}_1) = 1$, $\Delta(\mathcal{G}_2) = 0$, $\Delta(M_1) = \Delta(M_2) = 1$, $\Delta(M_3) = \Delta(M_4) = 2$, and $\Delta(J_2) = \Delta(J_3) = \Delta(J_4) = 2$, $\Delta(J_1) = 0$, $\Delta(\hat{J}_1) = 1$, $\Delta(\hat{J}_2) = \Delta(\hat{J}_3) = \Delta(\hat{J}_4) = 0$. Thus $\Delta(\mathcal{G}) = 1$ and $\Delta(\mathcal{M}) = 2$ and the schedule $S_{\mathcal{G}}S_{\mathcal{M}}$ has makespan 3. On the other hand $\Delta = 2$. Observe that a schedule with $C_{\max} = 2$ does not exist for this instance. Such a schedule would need individual operations of four different jobs to be scheduled in parallel on individual machines. This however contradicts the fact that only three jobs have individual operations in the instance.

Observe also that allowing preemptions at any point, not necessarily at integer points, may reduce schedule makespan. The schedule in Fig. 4.2 by allowing preemptions at any point reduces the makespan from $C_{\max} = 3$ to $C_{\max} = \frac{7}{3}$ for the instance in Fig. 4.2. Sections 4.2–4.9 focus on schedules with preemptions allowed at integer points only. Those schedules are solutions to the University timetabling problem. Section 4.10 considers preemptive schedules which solve preemptive open shop scheduling problem with multiprocessor operations. Those schedules allow preemptions at any points thus they do not necessarily solve the University timetabling problem; however, they become a good point of departure for approximate solutions, see Sect. 4.11.

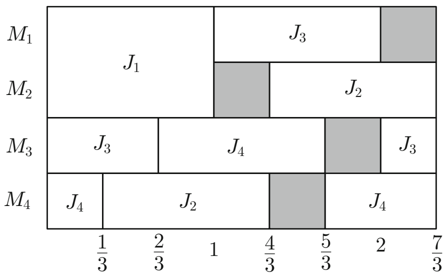


Fig. 4.2 An optimal schedule with preemptions allowed at any point for the instance in Fig. 4.1

4.2 Complexity of Short Schedules with Preemptions at Integer Points

Asratian and de Werra [1] prove the following.

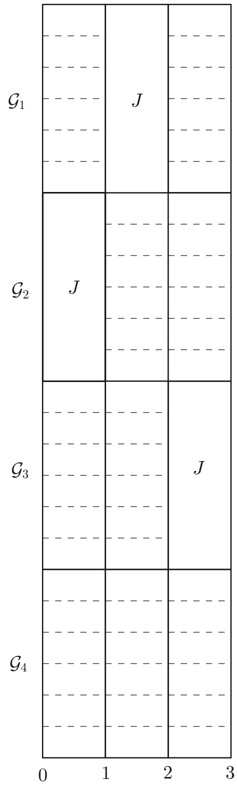
Theorem 4.1 *The problem to determine if there is a schedule with $C_{\max} \le 3$ for an open shop with multiprocessor operations and preemptions allowed at integer points only is NP-complete in the strong sense even for $p \le 4$.*

Proof The proof is by reduction from the following edge-coloring problem with pre-assigned colors. Let $G = (X, Y, E)$ be a bipartite graph with $\Delta(G) = 3$, where each vertex $v \in X$ is of degree 2 or 3. Moreover each vertex $v \in X$ has a set $C(v) \subseteq \{1, 2, 3\}$ of colors pre-assigned, and $|C(v)| = \deg_G(v)$. Can the edges of G be colored with colors $1, 2,$ and 3 so that the edges incident with $v \in X$ are colored with colors in $C(v)$? The problem is shown NP-complete in the strong sense in Even at al. [10], see also Asratian and Kamalian [2]. In the corresponding open shop instance we have $m = |X|$ machines, $\mathcal{M} = X$, partitioned into four disjoint groups $\mathcal{G}_1 = \{v \in X : C(x) = \{1, 3\}\}$, $\mathcal{G}_2 = \{v \in X : C(x) = \{2, 3\}\}$, $\mathcal{G}_3 = \{v \in X : C(x) = \{1, 2\}\}$, and $\mathcal{G}_4 = \{v \in X : C(x) = \{1, 2, 3\}\}$. The jobs in Y are processed on machines in X so that the operations of job $u \in Y$ are of unit processing time each, and processed on machines $v \in X$ adjacent with u in G. Moreover, there is one more job, the job J, with three group operations on $\mathcal{G}_1$, $\mathcal{G}_2$, and $\mathcal{G}_3$, no individual operations, and no group operation on $\mathcal{G}_4$ in the open shop instance. The three group operations of the job J have unit processing time each. Thus $\mathcal{J} = Y \cup \{J\}$, and $C_{\max} = 3$.

Suppose there is an edge-coloring of G with three colors 1, 2, and 3 so that the edges incident with $v \in X$ are colored with the colors in $C(x)$. Then a schedule can be readily obtained where each individual machine in $\mathcal{G}_1$ is occupied in [0, 1] and [2, 3], each individual machine in $\mathcal{G}_2$ is occupied in [1, 3], each individual machine in $\mathcal{G}_3$ is occupied in [0, 2], and each individual machine in $\mathcal{G}_4$ is occupied in [0, 3]. This allows to schedule J on $\mathcal{G}_1$ in [1, 2], on $\mathcal{G}_2$ in [0, 1], and on $\mathcal{G}_3$ in [2, 3] to get a schedule with $C_{\max} = 3$, see the schedule in Fig. 4.3.

Now suppose $\mathcal{S}$ is a schedule with $C_{\max} = 3$. Thus job J is processed at any time in the interval [0, 3]. Without loss of generality we can assume that group operation of J on $\mathcal{G}_2$ is in [0, 1], on $\mathcal{G}_1$ is in [1, 2], and on $\mathcal{G}_3$ is in [2, 3] in $\mathcal{S}$. To see this

Fig. 4.3 Scheduling job J and individual operations of jobs in $\mathcal{J}$



suppose that J is processed in the interval $[i-1, i]$ on $\mathcal{G}_1$, in $[j-1, j]$ on $\mathcal{G}_2$, and in $[k-1, k]$ on $\mathcal{G}_3$ in $\mathcal{S}$. We have $\{i, j, k\} = \{1, 2, 3\}$. Let $\mathcal{O}_i$, $\mathcal{O}_j$, and $\mathcal{O}_k$ be the sets of all unit-time operations, group or individual, processed in the unit-time intervals $[i-1, i]$, $[j-1, j]$, and in $[k-1, k]$, respectively, in $\mathcal{S}$. Schedule each operation from $\mathcal{O}_i$ in $[1, 2]$, each operation from $\mathcal{O}_j$ in $[0, 1]$, and each operation for $\mathcal{O}_k$ in $[2, 3]$. This permutation of the three unit-time intervals gives a feasible schedule with $C_{\max} = 3$, and the required order of processing for the operations of job J.

Thus an individual machine $v \in \mathcal{G}_1$ processes individual operations in $[0, 1]$ and $[2, 3]$. Those operations belong to jobs $u_1, u_2 \in Y$. Thus the edges $(v, u_1), (v, u_2) \in E$ incident with v will be colored with colors 1 and 3 which makes precisely the set $C(v) = \{1, 3\}$ required for vertex v. Similar argument works for any individual machine $v \in \mathcal{G}_2$, and any individual machine $v \in \mathcal{G}_3$. Thus the edges incident with $v \in \mathcal{G}_2$ and $v \in \mathcal{G}_3$ will be colored with colors 2 and 3, and 1 and 2 respectively. Therefore we obtain the required edge-coloring of G. □

de Werra et al [8] further strengthen Theorem 4.1 by proving it for three groups, $p = 3$. We will omit the proof and leave it as an exercise, see Problem 4.1. However the schedules, if any exist, with $C_{max} \leq 2$ and for an arbitrary number of groups p can be obtained in polynomial time. We have the following theorem.

Theorem 4.2 *The problem to determine if there is a schedule with $C_{\max} \leq 2$ for an open shop with multiprocessor operations and preemptions allowed at integer points only is polynomial.*

Proof Without loss of generality we can assume that each operation o, individual or group, has processing time 0, 1, or 2. We assume an arbitrary number of groups p. Split each operation o with processing time 2 into two unit-time operations o' and o''. The two belong to the same job and require the same machines, individual, or group, for processing as does o. For each unit-time operation o, define $\alpha_o = \mathcal{G}_\ell$ and $\beta_o = \{j\}$ if $o = \hat{O}_{j,\ell}$ is a group operation, and $\alpha_o = \{M_h\}$ and $\beta_o = \{j\}$ if $o = O_{j,h}$ is an individual operation. Let $G = (O, E)$ be a simple graph where O is the set of all unit-time operations, and E is a set of edges (o, o') such that the operations o and o' either share a machine, i.e., $\alpha_o \cap \alpha_{o'} \neq \emptyset$, or a job, i.e., $\beta_o \cap \beta_{o'} \neq \emptyset$. We claim that there is a schedule with $C_{\max} \leq 2$ if and only if the *vertices* of G can be colored with at most two colors so that any two vertices connected by and edge in E are colored with different colors. That is G is 2-colorable, Bondy and Murty [3]. Suppose G is 2-colorable with colors 1 and 2. Schedule each operation $o \in O$ on machines in α_o in the interval [0, 1] if the vertex o is colored with color 1, and in the interval $[1, 2]$ if the vertex o is colored with color 2. The schedule is feasible since $\alpha_o \cap \alpha_{o'} = \emptyset$ for any two operations o and o' both scheduled in the same time interval [0, 1] or $[1, 2]$, i.e., no two such operations share a machine (each machine processes at most one operation at a time). Moreover, $\beta_o \cap \beta_{o'} = \emptyset$ for any two operations o and o' both scheduled in the same time [0, 1] or $[1, 2]$, i.e., no two such operations belong to the same job (each job is processed by at most one machine, individual, or group, at a time). Therefore, there is a feasible schedule with $C_{\max} \leq 2$. Now suppose that there is a feasible schedule $\mathcal{S}$ with $C_{\max} \leq 2$. Without loss of generality we may assume that each operation, group, or individual completes at 1 or 2 in $\mathcal{S}$. Color each o that completes at 1 with color 1, and each o that completes at 2 with 2. Suppose for contradiction that there are operations o and o' connected by an edge $(o, o') \in E$ and colored with the same color $i = 1$ or 2 by the coloring. Thus both are scheduled in the same time interval $[i - 1, i]$ in $\mathcal{S}$, and since the schedule is feasible they must belong to different jobs and must not share a machine. Therefore $(o, o') \notin E$ which gives a contradiction.

Any simple graph is 2-colorable if and only if it is bipartite, Bondy and Murty [3]. Therefore there is a schedule with $C_{max} \leq 2$ if and only if the graph G is bipartite. The test whether G is bipartite or not can be done in $O(|O| + |E|)$ time. Therefore we just obtained a linear-time algorithm to test if there is a schedule with $C_{\max} \leq 2$. □

4.3 University Timetabling. A Polynomial-Time Algorithm and Conjecture for Two Groups

The University timetabling studied in this chapter was first introduced by Asratian and de Werra in [1]. The University timetabling is a generalization of the well-known, see Gotlieb [13], de Werra [7], and Bondy and Murty [3], class–teacher timetabling model. In the generalization, in addition to the lectures given by a single teacher to a single class, there are some lectures given by a single teacher to a group of classes simultaneously. We look for a minimum number of periods (period is a unit of time allocated to a lecture and it cannot be fractional in a solution to the timetabling problem) in which to complete all lectures without conflicts. The University timetabling model is motivated by the situation where various study programs share some courses which are common to all programs (classes). Asratian and de Werra [1] point out that such situation arises at Luleå University of Technology in Sweden and Ecole Polytechnique Fédérale de Lausanne (EPFL) in Switzerland. At EPFL for instance groups of three or four classes are created for courses of mathematics or physics which correspond to group-lectures. Besides those group-lectures there are individual lectures for courses given to one class (program) only, [1]. de Werra et al. [8] describe a similar situation at some French autonomous universities. Later on de Werra et al. [9], and Kis et al. [15] recast the problem as an equivalent open shop scheduling with multiprocessor operations and preemptions allowed at integer points only.

For two groups, $p = 2$, de Werra et al. [9] and Kis et al. [15] observe that a feasible schedule can be partitioned in the following four parts: part (a) consists of multiprocessor operations on $\mathcal{G}_1$, and single-processor operations or idle time on the machines in $\mathcal{G}_2$; part (b) consists of multiprocessor operations on both groups $\mathcal{G}_1$ and $\mathcal{G}_2$; part (c) consists of multiprocessor operations on $\mathcal{G}_2$, and single-processor operations or idle time on the machines in $\mathcal{G}_1$; and part (d) consists of single-processor operations or idle time on all machines, see Fig. 4.4. The parts (a), (b), (c), and (d) have sizes $\Delta(\mathcal{G}_1) - r$, r, $\Delta(\mathcal{G}_2) - r$, and w respectively for some r and w, where $\Delta(\mathcal{G}_\ell) = \sum_{j \in \mathcal{J}} a_{j,\ell}$ for $\ell = 1, 2$. Therefore the total of $\Delta(\mathcal{G}_1) + \Delta(\mathcal{G}_2) - r + w$ equals the schedule makespan, and the minimization of makespan reduces to the minimization of $w - r$. To simplify the notation we will often use h instead of M_h when referring to machine $M_h \in \mathcal{M}$, and j instead of J_j when referring to job $J_j \in \mathcal{J}$ in the remainder of this chapter.

The following integer linear program $\mathcal{ILP}$ with variables r, w, and y_{jh}, $x_{j\ell}$, for $j \in \mathcal{J}$, $h \in \mathcal{M}$, and $\ell = 1, 2$ was given in de Werra et al. [9] and Kis et al. [15] to minimize the makespan for $p = 2$:

$$ILP = \min(w - r). \tag{4.1}$$

Subject to

$$\sum_j b_{jh} - (\Delta(\mathcal{G}_2) - r) \le \sum_j y_{jh} \le w \qquad h \in \mathcal{G}_1 \tag{4.2}$$

$$\sum_j b_{jh} - (\Delta(\mathcal{G}_1) - r) \le \sum_j y_{jh} \le w \qquad h \in \mathcal{G}_2 \tag{4.3}$$

$$\sum_h y_{jh} \le w \qquad j \in \mathcal{J} \tag{4.4}$$

$$0 \le y_{jh} \le b_{jh} \qquad h \in \mathcal{M} \qquad j \in \mathcal{J} \tag{4.5}$$

$$\sum_j x_{j1} = r \tag{4.6}$$

$$\sum_j x_{j2} = r \tag{4.7}$$

$$x_{j1} + x_{j2} \le r \qquad j \in \mathcal{J} \tag{4.8}$$

$$0 \le x_{j\ell} \le a_{j\ell} \qquad j \in \mathcal{J} \qquad \ell = 1, 2 \tag{4.9}$$

$$\sum_{h \in \mathcal{G}_1} (b_{jh} - y_{jh}) + a_{j2} - x_{j2} \le \Delta(\mathcal{G}_2) - r \qquad j \in \mathcal{J} \tag{4.10}$$

$$\sum_{h \in \mathcal{G}_2} (b_{jh} - y_{jh}) + a_{j1} - x_{j1} \le \Delta(\mathcal{G}_1) - r \qquad j \in \mathcal{J} \tag{4.11}$$

$$\text{all variables } r,\ w,\ \text{and } y_{jh},\ x_{j\ell},\ \text{for } j \in \mathcal{J},\ h \in \mathcal{M},\ \text{and } \ell = 1, 2 \text{ are integers.} \tag{4.12}$$

The variable y_{jh} represents the amount of $j \in \mathcal{J}$ on $h \in \mathcal{M}$ in part (d). The variable $x_{j\ell}$ represents the amount of $j \in \mathcal{J}$ on $\mathcal{G}_\ell$, $\ell = 1, 2$, in part (b). The variable w is the size of (d), and the variable r is the size of (b). The constraints (4.2)–(4.5) guarantee that the size of part (d) does not exceed w. The constraints (4.6)–(4.9) guarantee that the size of part (b) equals r. The constraints (4.10)–(4.11) along with the left hand side inequalities in (4.2) and (4.3) guarantee that the size of part (a) does not exceed $\Delta(\mathcal{G}_1) - r$ and that the size of part (c) does not exceed $\Delta(\mathcal{G}_2) - r$.

Kis et al. [15] show how to solve the $\mathcal{ILP}$ in polynomial time. They further show that $\lceil LP \rceil \le ILP \le \lceil LP \rceil + 1$, where LP is the value of an optimal solution to the LP-relaxation of $\mathcal{ILP}$, and conjecture that:

Conjecture 4.1 $ILP = \lceil LP \rceil$.

We prove this conjecture in this chapter. We follow the proof given in Kubiak [16]. Observe that $\mathcal{G}_1 = \emptyset$ or $\mathcal{G}_2 = \emptyset$ results in integral solutions with makespan $\Delta(\mathcal{G}_2) + \max\{\max_j\{\sum_h b_{jh}\}, \max_h\{\sum_j b_{jh}\}\}$ or $\Delta(\mathcal{G}_1)+\max\{\max_j\{\sum_h b_{jh}\}, \max_h\{\sum_j b_{jh}\}\}$, respectively. Thus the conjecture holds in this case and we assume non-empty $\mathcal{G}_1$ and non-empty $\mathcal{G}_2$ from now on in Sects. 4.3–4.9. We begin in the next section by focusing on those solutions to the LP-relaxation with the value of objective function $\lceil LP \rceil$ that minimize r. The goal will be to show that the minimum r must be integer. This will be shown in Sects. 4.3–4.9. A detailed outline of the proof will be given in Sect. 4.3.2 once necessary notation and preliminary concepts are introduced there.

4.3.1 LP Relaxation with Minimum r

Let $(\mathbf{y}^*, \mathbf{x}^*, , r^*, w^*)$ be an optimal solution to the LP-relaxation of $\mathcal{ILP}$. Let $w^* = \lfloor w^* \rfloor + \lambda_{w^*}$ and $r^* = \lfloor r^* \rfloor + \lambda_{r^*}$, where $0 \leq \lambda_{w^*} < 1$ and $0 \leq \lambda_{r^*} < 1$. Consider the following linear program ℓp:

$$lp = \min r.$$

Subject to

$$w - r = \lceil w^* - r^* \rceil \tag{4.13}$$

$$\lfloor r^* \rfloor \leq r \tag{4.14}$$

$$\sum_j b_{jh} - (\Delta(\mathcal{G}_2) - r) \leq \sum_j y_{jh} \leq w \qquad h \in \mathcal{G}_1 \tag{4.15}$$

$$\sum_j b_{jh} - (\Delta(\mathcal{G}_1) - r) \leq \sum_j y_{jh} \leq w \qquad h \in \mathcal{G}_2 \tag{4.16}$$

$$\sum_h y_{jh} \leq w \qquad j \in \mathcal{J} \tag{4.17}$$

$$0 \leq y_{jh} \leq b_{jh} \qquad h \in \mathcal{M} \qquad j \in \mathcal{J} \tag{4.18}$$

$$\sum_j x_{j1} = r \tag{4.19}$$

$$\sum_j x_{j2} = r \tag{4.20}$$

$$x_{j1} + x_{j2} \leq r \qquad j \in \mathcal{J} \tag{4.21}$$

$$0 \le x_{j\ell} \le a_{j\ell} \qquad j \in \mathcal{J} \qquad \ell = 1, 2 \tag{4.22}$$

$$\sum_{h \in \mathcal{G}_1} (b_{jh} - y_{jh}) + a_{j2} - x_{j2} \le \Delta(\mathcal{G}_2) - r \qquad j \in \mathcal{J} \tag{4.23}$$

$$\sum_{h \in \mathcal{G}_2} (b_{jh} - y_{jh}) + a_{j1} - x_{j1} \le \Delta(\mathcal{G}_1) - r \qquad j \in \mathcal{J}. \tag{4.24}$$

All entries in the constraint matrix of ℓp are 0, +1, or −1, thus ℓp can be solved by a strongly polynomial algorithm given in Tardos [19]. Let $(\mathbf{y}, \mathbf{x}, r, w)$ be an optimal solution to ℓp. The solution exists since $(\mathbf{y}^*, \mathbf{x}^*, r^*, \lfloor w^* \rfloor + \lambda_{r^*})$ is feasible for ℓp if $\lambda_{w^*} \le \lambda_{r^*}$, and $(\mathbf{y}^*, \mathbf{x}^*, r^*, \lceil w^* \rceil + \lambda_{r^*})$ is feasible for ℓp if $\lambda_{w^*} > \lambda_{r^*}$, thus ℓp is feasible and clearly it is also bounded. Observe that $\lfloor w^* \rfloor + \lambda_{r^*} - r^* = \lfloor w^* \rfloor - \lfloor r^* \rfloor = \lceil w^* - r^* \rceil$ for $\lambda_{w^*} \le \lambda_{r^*}$, and $\lceil w^* \rceil + \lambda_{r^*} - r^* = \lceil w^* \rceil - \lfloor r^* \rfloor = \lceil w^* - r^* \rceil$ for $\lambda_{w^*} > \lambda_{r^*}$.

We assume without loss of generality that the solution meets the machine *saturation condition*, i.e., the upper and lower bounds in (4.15) and (4.16) are equal. If the machine saturation is not met by the solution for some machine h, then a job $j(h)$ with $b_{j(h)h} = w - \sum_j b_{jh} + (\Delta(\mathcal{G}_2) - r)$, $a_{j(h)1} = a_{j(h)2} = 0$ should be added to the instance for each such machine to make the solution meet the saturation condition. Observe that by (4.13) $b_{j(h)h}$ is integral so the extended instance is a valid instance of the open shop with multiprocessor operations problem. We take $y_{j(h)h} = w - \sum_j y_{jh}$ in the extended solution. Observe that $n = |\mathcal{J}| \ge |\mathcal{G}_1| + |\mathcal{G}_2|$ for the solutions that meet the saturation condition.

An integral solution $(\mathbf{y}, \mathbf{x}, r, w)$ to ℓp is feasible for $\mathcal{ILP}$, and $w - r = \lceil w^* - r^* \rceil = \lceil LP \rceil$. Moreover this solution is optimal for $\mathcal{ILP}$ since by definition of LP-relaxation we have $LP \le ILP$ for any feasible solution to $\mathcal{ILP}$. This proves Conjecture 4.1. Therefore it suffices to prove that there is an integral solution to ℓp. To that end, we prove the following theorem in Sects. 4.3–4.9.

Theorem 4.3 *The r in an optimal solution to ℓp is integral. Moreover, there is optimal solution to ℓp that is integral.*

Proof Let $\mathbf{s} = (\mathbf{y}, \mathbf{x}, r, w)$ be an optimal solution to ℓp. Suppose for a contradiction that the r in $\mathbf{s}$ equals

$$r = \lfloor r \rfloor + \epsilon,$$

where $0 < \epsilon < 1$. Thus by (4.13)

$$w = \lfloor w \rfloor + \epsilon\ .$$

In Sects. 4.3–4.9 we show that such $\mathbf{s}$ cannot be optimal which leads to a contradiction and proves the first part of the theorem. We then show that an optimal solution that is integral can be found in polynomial time. An outline of the proof will be given at the end of the next section after we first introduce the necessary notations and definitions. □

4.3.2 Preliminaries

Consider the solution $\mathbf{s} = (\mathbf{y}, \mathbf{x}, r, w)$. Let B_1 be the set of all jobs j with fractional x_{j1}, and let B_2 be the set of all jobs j with fractional x_{j2}. Clearly both sets are non-empty because $\epsilon > 0$. By (4.19) and (4.20) the fractions in B_ℓ sum up to $i_\ell + \epsilon$ ($\sum_{j \in B_\ell} \varepsilon_j = i_\ell + \epsilon$), where i_ℓ is a non-negative integer, for $\ell = 1, 2$.

A job j is *d-tight* if

$$\sum_h y_{jh} = w.$$

Denote by D the set of all d-tight jobs.

A job j is *a-tight* if

$$\sum_{h \in \mathcal{G}_2} (b_{jh} - y_{jh}) + a_{j1} - x_{j1} = \Delta(\mathcal{G}_1) - r.$$

A job j is *c-tight* if

$$\sum_{h \in \mathcal{G}_1} (b_{jh} - y_{jh}) + a_{j2} - x_{j2} = \Delta(\mathcal{G}_2) - r.$$

For jobs g and k such that $x_{g1} > 0$ and $x_{k2} > 0$ define

$$\varepsilon_r(g,k) = \begin{cases} \min_{j \in (B_1 \cup B_2) \setminus \{g,k\}} \{r - (x_{j1} + x_{j2}), \epsilon\} & \text{if } (B_1 \cup B_2) \setminus \{g,k\} \neq \emptyset; \\ \epsilon & \text{if } B_1 \cup B_2 \subseteq \{g,k\}\ . \end{cases}$$

Observe that jobs g and k with $\varepsilon_r(g,k) > 0$ can potentially be used to obtain a solution to lp with smaller r since the reduction of both x_{g1} and x_{k2} by some small enough $\varepsilon > 0$ will leave the resulting constraint (4.21) satisfied. Moreover define

$$\varepsilon_c(k) = \sum_{h \in \mathcal{G}_1} y_{kh} - \left(\sum_{h \in \mathcal{G}_1} b_{kh} + a_{k2} - x_{k2} - \Delta(\mathcal{G}_2) + r \right),$$

$$\varepsilon_a(g) = \sum_{h \in \mathcal{G}_2} y_{gh} - \left(\sum_{h \in \mathcal{G}_2} b_{gh} + a_{g1} - x_{g1} - \Delta(\mathcal{G}_1) + r \right).$$

Let G be a job–machine bipartite graph such that there is an edge between machine $h \in \mathcal{M}$ and job $j \in \mathcal{J}$ if and only if $y_{jh} > 0$. The edge has multiplicity y_{jh} (the multiplicity may be fractional). A *column* $I = (M_I, \varepsilon_I)$ consists of a matching M_I in G that matches all m machines in $\mathcal{M}$ with a subset of exactly m jobs in $\mathcal{J}$ that are scheduled simultaneously in the solution **s**, and its multiplicity $\varepsilon_I > 0$ (the multiplicity may be fractional). That is the jobs matched in the column I are processed simultaneously for ε_I time units. Let $\mathcal{J}_I$ be the set of all jobs matched in M_I, i.e., $\mathcal{J}_I = \{j \in \mathcal{J} : (j, h) \in M_I \quad \text{for some} \quad h \in \mathcal{M}\}$. By definition of D we require that $D \subseteq \mathcal{J}_I$ for a column in **s**. By Gonzalez and Sahni [12], see also Gabow and Kariv [11] and Sect. 3.7.1 (Birkhoff–von Neumann theorem), part (d) can be represented by a set of columns $d(\mathbf{y}, w) = \{I_1, \ldots, I_p\}$. In the spirit of Birkhoff–von Neumann theorem, we can recast $d(\mathbf{y}, w)$ as follows. Let $\mathbb{Y}$ be an $n \times m$ matrix where the entry in row i and column h equals y_{ih}, and let $\mathbb{P}_I$ be an $n \times m$, 0-1 matrix corresponding to column $I = (M_I, \varepsilon_I) \in d(\mathbf{y}, w)$. The entry in row i and column h of $\mathbb{P}_I$ equals 1 if and only if job i is matched with machine h in M_I. We then can decompose $\mathbb{Y}$ as follows:

$$\mathbb{Y} = \varepsilon_{I_1}\mathbb{P}_{I_1} + \cdots + \varepsilon_{I_p}\mathbb{P}_{I_p}.$$

For a set X of columns let $l(X)$ denote the total multiplicity of all columns in X. We have $l(d(\mathbf{y}, w)) = w$ and $l(X_j) = \sum_h y_{jh} \leq w$ where X_j is the set of all columns that match job $j \in \mathcal{J}$. Let $I_1 = (M_{I_1}, \varepsilon_{I_1}), \ldots, I_q = (M_{I_q}, \varepsilon_{I_q})$ be a subset of $q \geq 1$ columns from $d(\mathbf{y}, w)$, the set of columns $Y = \{(M_{I_1}, \lambda_1), \ldots, (M_{I_q}, \lambda_q)\}$, where $0 \leq \lambda_1 \leq \varepsilon_{I_1}, \ldots, 0 \leq \lambda_q \leq \varepsilon_{I_q}$ and $\lambda_1 + \ldots + \lambda_q = \lambda$ is called the *interval* of length λ in $d(\mathbf{y}, w)$. Let $d(\mathbf{y}, w) \setminus Y$ be the set of all columns in $d(\mathbf{y}, w)$ with columns in Y removed. For each $j \in \mathcal{J}$ we have $l(Z_j) \leq l(d(\mathbf{y}, w) \setminus Y) = w - \lambda$ where Z_j is the set of all columns that match j in $d(\mathbf{y}, w) \setminus Y$.

Let $u_1, \ldots, u_p$ and $l_1, \ldots, l_q$ be different jobs from $\mathcal{J}$, and I be a column. We say that I is of type

$$\begin{pmatrix} *, u_1, \ldots, u_p \\ *, l_1, \ldots, l_q \end{pmatrix}$$

if $\{(u_1, h_1), \ldots, (u_p, h_p)\} \subseteq M_I$ for some machines $h_1, \ldots, h_p$ in $\mathcal{G}_1$, and $\{(l_1, H_1), \ldots, (l_q, H_q)\} \subseteq M_I$ for some machines $H_1, \ldots, H_q$ in $\mathcal{G}_2$. The asterisk denotes any matching for other jobs. For convenience, we sometimes use the following notation:

$$\begin{pmatrix} *, U \\ *, L \end{pmatrix},$$

where $U = \{u_1, \ldots, u_p\}$ and $L = \{l_1, \ldots, l_q\}$ instead. By definition if $p = 0$ or $q = 0$, then the asterisk alone denotes any matching on $\mathcal{G}_1$ or $\mathcal{G}_2$, respectively. We extend this notation for convenience as follows. Let u and l be different jobs from $\mathcal{J}$, and I be a column. We say that I is of type

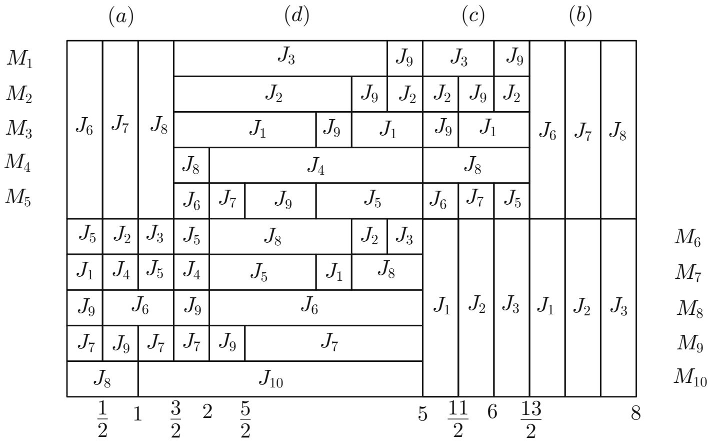


Fig. 4.4 An example of solution $\mathbf{s} = (\mathbf{y}, \mathbf{x}, \frac{3}{2}, \frac{7}{2})$ and its corresponding schedule S (parts (a), (d), and (c))

$$\begin{pmatrix} *, \overline{u} \\ *, \overline{l} \end{pmatrix}$$

if $(u, h) \notin M_I$ for any machine $h \in \mathcal{G}_1$, and $(l, H) \notin M_I$ for any machine $H \in \mathcal{G}_2$.

The concepts just introduced are illustrated in Fig. 4.4. The makespan of S equals 8, and the schedule S is clearly the shortest possible. The instance itself consists of $n = 10$ jobs and $m = 10$ machines, $\mathcal{G}_1 = \{M_1, M_2, M_3, M_4, M_5\}$ and $\mathcal{G}_2 = \{M_6, M_7, M_8, M_9, M_{10}\}$. The processing times of operations can easily be obtained from S, for example, for job J_1 we have $b_{13} = 4$, $b_{17} = 1$, $a_{12} = 1$ and all remaining operations have processing time 0, and for job J_9 we have $b_{91} = b_{92} = b_{93} = b_{95} = b_{98} = b_{99} = 1$ and all remaining operations have processing time 0. The solution $\mathbf{s}$ can also be easily obtained from S, for example, for job J_1 we have $y_{13} = 3$, $y_{17} = \frac{1}{2}$, $x_{12} = \frac{1}{2}$ and all remaining variables are set to 0, and for job J_9 we have $y_{91} = y_{92} = y_{93} = y_{98} = y_{99} = \frac{1}{2}$ and $y_{95} = 1$ all remaining variables are set 0. In S: $w = \frac{7}{2}$, $r = \frac{3}{2}$, $\epsilon = \frac{1}{2}$, $i_1 = i_2 = 1$, $B_1 = \{J_1, J_2, J_3\}$, $B_2 = \{J_6, J_7, J_8\}$, and $\varepsilon_r(g, k) = \frac{1}{2}$ for each pair $g \in B_1$ and $k \in B_2$. All jobs are d-tight; jobs J_1, J_2, J_3, J_8, and J_9 are c-tight; jobs J_6, J_7, and J_8 are a-tight. The matching $M = \{(M_1, J_3), (M_2, J_2), (M_3, J_1), (M_4, J_8), (M_5, J_6), (M_6, J_5), (M_7, J_4), (M_8, J_9), (M_9, J_7), (M_{10}, J_{10})\}$, and the multiplicity $\frac{1}{2}$ make up a column $(M, \frac{1}{2})$ which is the schedule S in the interval $[\frac{3}{2}, 2]$. All other details of $\mathbf{s}$ and S should now be clear from Fig. 4.4. We show later in Fig. 4.5 that $\mathbf{s}$ is not optimal for ℓp since ℓp admits solution with $r = 1$ and the same makespan 8.

4.3.3 Outline of the Proof

We now give a high level informal overview of the proof of the conjecture before moving to its details in the remaining sections. The proof is by contradiction. The solution **s** defines four open shops, one for each part (a), (b), (c), and (d). The bipartite graph G with the edge multiplicities y_{jh} obtained from the solution **s** defines an m-machine open shop with operation processing times equal y_{jh} for part (d), we call this part d-open shop. The groups $\mathcal{G}_1$ and $\mathcal{G}_2$ define a two-machine open shop with operation processing times x_{j1} and x_{j_2} for part (b), we call this part b-open shop. The group $\mathcal{G}_1$ and the individual machines in $\mathcal{G}_2$ make up a $(|\mathcal{G}_2| + 1)$-machine open shop with operation processing times $b_{jh} - y_{jh}$ on the individual machines in $\mathcal{G}_2$ and $a_{j1} - x_{j1}$ on the group $\mathcal{G}_1$ for part (a). Similarly, the group $\mathcal{G}_2$ and individual machines in $\mathcal{G}_1$ make up a $(|\mathcal{G}_1| + 1)$-machine open shop with operation processing times $b_{jh} - y_{jh}$ on the individual machines in $\mathcal{G}_1$ and $a_{j2} - x_{j2}$ on the group $\mathcal{G}_2$ for part (c). We call these two a-open shop and c-open shop, respectively. All four open shops are interrelated since they share jobs, individual machines, or groups, thus a local change to one affects the other open shops as well. Notice that all open shops are defined by the solution **s** rather than directly by the problem instance which normally is the case for open shops.The open shops for (a), (d), and (c) are shown in Fig. 4.4 for illustration. The makespan of each open shop is fractional, both r and w are fractional in **s**; however, the total makespan is integral since $w - r$ is integral in **s**.

Sections 4.3.4 and 4.4 give a matching-based approach to characterize those columns in d-open shop that cannot occur in **s** with $\epsilon > 0$ since their presence would contradict the optimality of r. Namely, those columns, if occurred in **s**, could be used along with the x_{j1}, x_{j2} to find another feasible solution with parts (b) and (d) shorter by ε, $0 < \varepsilon \leq \epsilon$, each, and parts (a) and (c) longer by ε, $0 < \varepsilon \leq \epsilon$, each so that the total makespan does not change. More precisely the approach uses the column matchings in d-open shop on $\mathcal{G}_1$ and $\mathcal{G}_2$ separately; this structure is reflected in the notation for the column type, in order to match the former with some x_{j2} and the latter with some x_{j1} so that we get a feasible solution with the same makespan yet d-open shop shorter by ε. The matching-based approach leads to the characterizations of the d-open shop given in Sects. 4.4.1 and 4.5, and the b-open shop in Sect. 4.4.2; however, it is insufficient to prove the conjecture. Nevertheless both characterizations are key for the subsequent sections.

Therefore we introduce a network flow-based approach to shorten the d open shop makespan from w to $\lfloor w \rfloor$ and the b open shop makespan from r to $\lfloor r \rfloor$ in order to obtain a feasible solution with the same total makespan. We show that this approach works by constructing two network flow problems for d-open shop, one for the case with $\sum_{j \in B_1} \varepsilon_j = \epsilon$ or $\sum_{j \in B_2} \varepsilon_j = \epsilon$ in Sect. 4.6, and the other for the case with $\sum_{j \in B_1} \varepsilon_j = i_1 + \epsilon$ and $\sum_{j \in B_2} \varepsilon_j = i_2 + \epsilon$ for some positive integers i_1 and i_2 in Sect. 4.7. The network flow problems have integral lower and upper bounds on the arc flows which means they admit integral flows provided that feasible flows *exist* at all. To that end we show how to construct a feasible flow for each network from the solution $\mathbf{s}$ in Sects. 4.6 and 4.8. The construction relies on the characteristics of d-open shop given in Sects. 4.4.1 and 4.5. The characteristics naturally focus on the sets B_1 and B_2 in $\mathbf{s}$ since any change to the four open shops needs to involve the changes to the jobs in $B_1 \cup B_2$. That is not all since the integral solutions to the network flow problems give integral solutions to the d-open shop only. Those solutions need to be subsequently extended to the other three open shops while preserving the whole solution feasibility and the total makespan. This is also done in Sects. 4.6 and 4.8. The extension relies on characteristics of the b-open shop proved in Sect. 4.4.2 where we prove that $B_1 \cap B_2 = \emptyset$ in $\mathbf{s}$, i.e., x_{j1} and x_{j2} cannot be both fractional, and Sect. 4.5 where we prove that the product $x_{j1}x_{j2} = 0$ for each job $j \in \mathcal{J}$ in $\mathbf{s}$ except for the case where $B_1 = \{j\}$ or $B_2 = \{j\}$. The characteristics make it possible to find integral feasible solutions for the b-, c-, and a-open shops consistent with the network-flow solutions to the d-open shop. Finally we show in Sect. 4.9 that the network-flow based approach leads to contradiction since it shortens r assumed to be the shortest possible. This proves the conjecture.

4.3.4 Columns Absent from $d(\mathbf{y}, w)$ in $\mathbf{s}$

In this section we show that for two different jobs g and k such that $x_{g1} > 0$ and $x_{k2} > 0$ certain columns or subsets of columns must be missing from $d(\mathbf{y}, w)$ if $\epsilon > 0$. Though these results are contingent on $\varepsilon_r(g, k) > 0$, we show that this condition often holds, for instance in Sect. 4.4.2 we show that this inequality holds for each pair $g \in B_1$ and $k \in B_2$.

Let g and k be two different jobs such that $x_{g1} > 0$ and $x_{k2} > 0$. A (g, k)-*feasible semi-matching* in G is a set of edges $E = E_1 \cup E_2$ of G of cardinality $m = |\mathcal{G}_1| + |\mathcal{G}_2|$ such that

1. $E_1 = \{(j, h) \in E : h \in \mathcal{G}_1\}$ and $E_2 = \{(j, h) \in E : h \in \mathcal{G}_2\}$ are matchings.
2. There are $h \in \mathcal{M}$ and $(j, h) \in E$ for each $j \in D$.
3. If $\varepsilon_a(g) = 0$, then $(g, h) \notin E_2$ for any $h \in \mathcal{G}_2$.
4. If $\varepsilon_c(k) = 0$, then $(k, h) \notin E_1$ for any $h \in \mathcal{G}_1$.

If E is a matching, then a (g, k)-feasible semi-matching in G is called a (g, k)-*feasible matching* in G.

We define solution $(\mathbf{y}(E), \mathbf{x}(g, k), r(g, k), w(g, k), \varepsilon)$ for jobs g, k, and a (g, k)-feasible semi-matching E, where

$$\varepsilon = \begin{cases} \varepsilon' & \text{if } \varepsilon_a(g) = 0 \text{ and } \varepsilon_c(k) = 0\,; \\ \min\{\varepsilon', \varepsilon_a(g)\} & \text{if } \varepsilon_a(g) > 0 \text{ and } \varepsilon_c(k) = 0\,; \\ \min\{\varepsilon', \varepsilon_c(k)\} & \text{if } \varepsilon_a(g) = 0 \text{ and } \varepsilon_c(k) > 0\,; \\ \min\{\varepsilon', \varepsilon_a(g), \varepsilon_c(k)\} & \text{if } \varepsilon_a(g) > 0 \text{ and } \varepsilon_c(k) > 0\,; \end{cases} \tag{4.25}$$

and

$$\varepsilon' = \min\{\varepsilon_r(g,k), x_{g1}, x_{k2}, \min_{(j,h)\in E}\{y_{jh}\}, \min_{j\in\mathcal{J}\setminus D}\{w - \sum_h y_{jh}\}\}\ , \tag{4.26}$$

as follows:

$$y_{jh}(E) = \begin{cases} y_{jh} - \varepsilon & \text{if } (j,h) \in E\ ; \\ y_{jh} & \text{otherwise}\ ; \end{cases} \tag{4.27}$$

$$x_{j1}(g,k) = \begin{cases} x_{g1} - \varepsilon & \text{if } j = g\ ; \\ x_{j1} & \text{if } j \neq g; \end{cases} \tag{4.28}$$

$$x_{j2}(g,k) = \begin{cases} x_{k2} - \varepsilon & \text{if } j = k\ ; \\ x_{j2} & \text{if } j \neq k; \end{cases} \tag{4.29}$$

$$r(g,k) = r - \varepsilon; \tag{4.30}$$

$$w(g,k) = w - \varepsilon\ . \tag{4.31}$$

We have the following lemma.

Lemma 4.1 *Let g and k be two different jobs such that $x_{g1} > 0$ and $x_{k2} > 0$. If $\varepsilon_r(g,k) > 0$, then no (g,k)-feasible semi-matching E in G exists.*

Proof Details can be found in Kubiak [16]. □

Lemma 4.2 *Let g and k be two different jobs such that $x_{g1} > 0$ and $x_{k2} > 0$. If $\varepsilon_r(g,k) > 0$, then no column of type $\binom{*,\bar{k}}{*,\bar{g}}$ exists in $d(\mathbf{y}, w)$.*

Proof If such a column $I = (M_I, \varepsilon_I)$ exists, then M_I is (g,k)-feasible semi-matching E in G which contradicts Lemma 4.1. □

We now consider another forbidden configuration of columns in $d(\mathbf{y}, w)$. Let $I_1 = (M_{I_1}, \varepsilon_{I_1})$ and $I_2 = (M_{I_2}, \varepsilon_{I_2})$ be two columns. Let g, k, a, and b be four different jobs such that $x_{g1} > 0$, $x_{k2} > 0$, $x_{a1} > 0$, and $x_{b2} > 0$. Define solution $(\mathbf{y}(I_1, I_2), \mathbf{x}', r', w', \varepsilon)$, where

$$\varepsilon = \min\{\varepsilon_r(g,k), \varepsilon_r(a,b), x_{g1}, x_{a1}, x_{b2}, x_{k2}, \varepsilon_{I_1}, \varepsilon_{I_2}, \min_{j\in\mathcal{J}\setminus D}\{w - \sum_h y_{jh}\}\} \tag{4.32}$$

as follows:

$$y_{jh}(I_1, I_2) = \begin{cases} y_{jh} - \varepsilon & \text{if } (j,h) \in M_{I_1} \text{ and } (j,h) \in M_{I_2}\ ; \\ y_{jh} - \varepsilon/2 & \text{if } (j,h) \in M_{I_1} \text{ and } (j,h) \notin M_{I_2}\ ; \\ y_{jh} - \varepsilon/2 & \text{if } (j,h) \notin M_{I_1} \text{ and } (j,h) \in M_{I_2}\ ; \\ y_{jh} & \text{otherwise}\ ; \end{cases} \tag{4.33}$$

$$x'_{j1} = \begin{cases} x_{j1} - \varepsilon/2 & \text{if } j = g \text{ or } j = a\ ; \\ x_{j1} & \text{otherwise}\ ; \end{cases} \tag{4.34}$$

$$x'_{j2} = \begin{cases} x_{j2} - \varepsilon/2 & \text{if } j = k \text{ or } j = b\ ; \\ x_{j2} & \text{otherwise}\ ; \end{cases} \tag{4.35}$$

$$r' = r - \varepsilon; \tag{4.36}$$

$$w' = w - \varepsilon\ . \tag{4.37}$$

We have the following lemma

Lemma 4.3 *Let g, k, a, and b be four different jobs such that $x_{g1} > 0$, $x_{k2} > 0$, $x_{a1} > 0$, and $x_{b2} > 0$. If $\varepsilon_r(g, k) > 0$ and $\varepsilon_r(a, b) > 0$, then a column of type $\binom{*,a,b,g,k}{*}$ does not exist in $d(\mathbf{y}, w)$ or a column of type $\binom{*}{*,a,b,k,g}$ does not exist in $d(\mathbf{y}, w)$.*

Proof Details can be found in Kubiak [16]. □

The following two corollaries follow immediately from the proof of Lemma 4.3.

Corollary 4.1 *Let g, k, and a be three different jobs such that $x_{g1} > 0$, $x_{k2} > 0$, and $x_{a1}x_{a2} > 0$. If $\varepsilon_r(g, a) > 0$ and $\varepsilon_r(a, k) > 0$, then a column of type $\binom{*,a,g,k}{*}$ does not exist in $d(\mathbf{y}, w)$ or a column of type $\binom{*}{*,a,k,g}$ does not exist in $d(\mathbf{y}, w)$.*

Corollary 4.2 *Let g and k be two different jobs such that $x_{g1}x_{g2} > 0$, and $x_{k1}x_{k2} > 0$. If $\varepsilon_r(g, k) > 0$, then a column of type $\binom{*,g,k}{*}$ does not exist in $d(\mathbf{y}, w)$ or a column of type $\binom{*}{*,k,g}$ does not exist in $d(\mathbf{y}, w)$.*

4.4 Pairs of Columns Absent from $d(\mathbf{y}, w)$ in s

Let g and k be two different jobs such that $x_{g1} > 0$, $x_{k2} > 0$. Let $I_k = (M_{I_k}, \varepsilon_{I_k})$ be a column of type $\binom{*,\bar{k}}{*}$, and $I_g = (M_{I_g}, \varepsilon_{I_g})$ a column of type $\binom{*}{*,\bar{g}}$. Without loss of generality we assume $\varepsilon_{I_k} = \varepsilon_{I_g} = \varepsilon$. Let $G(I_g, I_k) = (M_{I_g} \cup M_{I_k})$ be a job–machine bipartite multigraph, where an edge connects a machine h and a job j if and only if $(j, h) \in M_{I_g} \cup M_{I_k}$. The degree of each machine-vertex in $G(I_g, I_k)$ is exactly 2 and the degree of each job-vertex in $G(I_g, I_k)$ is either 1 or 2. Thus, $G(I_g, I_k)$ is a collection of connected components each of which is either a job–machine path or a job–machine cycle. We have the following lemma for I_k and I_g.

Lemma 4.4 *If $I_k, I_g \in d(\mathbf{y}, w)$, and $\varepsilon_r(g, k) > 0$, then I_k is of type $\binom{*}{*,k,g}$ and I_g is of type $\binom{*,k,g}{*}$ and both k and g belong to the same connected component of $G(I_g, I_k)$.*

Proof Column I_k either has no job k on any machine (we say I_k is *k-free*) or it is of type $\binom{*}{*,k}$. In the former case k is either missing from $G(I_g, I_k)$ or it is of degree 1 in $G(I_g, I_k)$. In the latter case I_k is either of type $\binom{*}{*,k,g}$ or of type $\binom{*,g}{*,k}$ or it is g-free. Since $\varepsilon_r(g, k) > 0$, by Lemma 4.2 I_k cannot be of type $\binom{*,g}{*,k}$ nor can I_k be g-free. Thus I_k is of type $\binom{*}{*,k,g}$ or I_k is k-free. In the latter case, by Lemma 4.2, I_g must be of type $\binom{*,k}{*,\overline{g}}$. A similar argument shows that I_g is of type $\binom{*,k,g}{*}$ or I_g is g-free. In the latter case, by Lemma 4.2, I_k must be of type $\binom{*,\overline{k}}{*,g}$. Thus we end up with the following four cases:

1. I_k is of type $\binom{*}{*,k,g}$, and I_g is of type $\binom{*,k,g}{*}$;
2. I_k is of type $\binom{*}{*,k,g}$, and I_g is g-free. By Lemma 4.2, I_g cannot be k-free. Hence k is of degree 2 and g is of degree 1 in $G(I_g, I_k)$;
3. I_k is k-free, and I_g is of type $\binom{*,k,g}{*}$. By Lemma 4.2, I_k cannot be g-free. Hence g is of degree 2 and k is of degree 1 in $G(I_g, I_k)$;
4. I_k is k-free and is of type $\binom{*}{*,g}$, and I_g is g-free and is of type $\binom{*,k}{*}$. Hence both g and k are of degree 1 in $G(I_g, I_k)$.

In Case (2), let g and k be in the same connected component P of $G(I_g, I_k)$. Then P is a job–machine path

$$g, h_1, j_1, \ldots h_i, k, h_{i+1}, j_{i+1}, \ldots, h_\ell, j_\ell,$$

where $h_1 \in \mathcal{G}_2$ and $\{h_i, h_{i+1}\} \cap \mathcal{G}_2 \neq \emptyset$. If $h_i \in \mathcal{G}_2$, then match the jobs with the machines as follows:

$$M = \{(h_1, j_1), \ldots, (h_{i-1}, j_{i-1}), (h_i, k), (h_{i+1}, j_{i+1}), \ldots, (h_\ell, j_\ell)\}$$

in the component P. If $h_i \in \mathcal{G}_1$, then there is a job $j_{i^*} \in \{j_1, \ldots, j_{i-1}\}$ such that $h_{i^*} \in \mathcal{G}_2$ and $h_{i^*+1} \in \mathcal{G}_1$. Then match the jobs with the machines as follows:

$$M = \{(h_1, j_1), \ldots, (h_{i^*-1}, j_{i^*-1}), (h_{i^*}, j_{i^*}), (h_{i^*+1}, j_{i^*}),$$
$$\ldots, (h_i, j_{i-1}), (h_{i+1}, k), \ldots, (h_\ell, j_{\ell-1})\}$$

in the component P. Thus each machine in P is matched exactly once, each job of degree 2 in P is matched at least once (actually each such job is matched exactly once except job j_{i^*} that is matched exactly twice: with $h_{i^*} \in \mathcal{G}_2$ and $h_{i^*+1} \in \mathcal{G}_1$), g is omitted from the matching, and k is matched with a machine in $\mathcal{G}_2$. The matching can easily be extended by adding matchings from the remaining connected components of $G(I_g, I_k)$. The result is a (g, k)-feasible semi-matching in $G(I_g, I_k)$. We proceed in a similar fashion in Case (3) if k and g are in the same component P of $G(I_g, I_k)$. In Case (4) if g and k are in the same connected component P of $G(I_g, I_k)$, then P is a job–machine path

$$g, h_1, j_1, \ldots h_i, k,$$

with $h_1 \in \mathcal{G}_2$ and $h_i \in \mathcal{G}_1$. Then there is job $j_{i^*} \in \{j_1, \ldots, j_{i-1}\}$ such that $h_{i^*} \in \mathcal{G}_2$ and $h_{i^*+1} \in \mathcal{G}_1$. Match the jobs with the machines as follows:

$$M = \{(h_1, j_1), \ldots, (h_{i^*-1}, j_{i^*-1}), (h_{i^*}, j_{i^*}), (h_{i^*+1}, j_{i^*}), \ldots, (h_i, j_{i-1})\}$$

in the component P. Thus each machine in P is matched exactly once, each job of degree 2 in P is matched at least once (actually each such job is matched exactly once except job j_{i^*} that is matched exactly twice: with $h_{i^*} \in \mathcal{G}_2$ and $h_{i^*+1} \in \mathcal{G}_1$), g and k are omitted from the matching. The matching can easily be extended by adding matchings from the remaining connected components of $G(I_g, I_k)$. The result is a (g, k)-feasible semi-matching in $G(I_g, I_k)$.

Let us now assume that k is in connected component C_k and g is in a connected component C_g and $C_k \neq C_g$. We have

1. In Case (1), k is of degree 2 and both on a machine in $\mathcal{G}_1$ and on a machine in $\mathcal{G}_2$ in C_k, and g is of degree 2 and both on a machine in $\mathcal{G}_1$ and on a machine in $\mathcal{G}_2$ in C_g.
2. In Case (2), k is of degree 2 and on $h \in \mathcal{G}_2$ in C_k, and g is of degree 1 in C_g.
3. In Case (3), g is of degree 2 and on $h \in \mathcal{G}_1$ in C_g, and k is of degree 1 in C_k.
4. In Case (4), g is of degree 1 in C_g, and k is of degree 1 in C_k.

A matching for C_k is selected so that k is matched with the machine in $\mathcal{G}_2$, if k is of degree 2, or omitted from the matching, if k is of degree 1. Similarly a matching for C_g is selected so that g is matched with the machine in $\mathcal{G}_1$, if g is of degree 2, or omitted from the matching if g is of degree 1. The matching can easily be extended by adding matchings from the remaining connected components of $G(I_g, I_k)$. The result is a (g, k)-feasible semi-matching in $G(I_g, I_k)$. Thus in all cases, except Case (1) with both k and g being in the same connected component of $G(I_g, I_k)$, we showed how to obtain (g, k)-feasible semi-matching M in $G(I_g, I_k)$. This however contradicts Lemma 4.1 since I_k, I_g in $d(\mathbf{y}, w)$ can be replaced by columns $I' = (M, \varepsilon)$ and $I'' = ((M_{I_g} \cup M_{I_k}) \setminus M, \varepsilon)$ resulting into another feasible solution to ℓp with the same value r of objective function but with a (g, k)-feasible semi-matching M. □

4.4.1 The a-, c-, and d-Tightness in s

We show that each job in B_1 is both a-tight and d-tight, and each job in B_2 is both c-tight and d-tight. We begin by showing the a- and c- tightness.

Lemma 4.5 *Each job $g \in B_1$ is a-tight and*

$$\sum_{h \in \mathcal{G}_2} y_{gh} < w, \tag{4.38}$$

and each job $k \in B_2$ is c-tight and

$$\sum_{h \in \mathcal{G}_1} y_{kh} < w. \tag{4.39}$$

Proof Details can be found in Kubiak [16]. □

We now prove d-tightness for each job in $B_1 \cup B_2$.

Theorem 4.4 *Each job in $B_1 \cup B_2$ is d-tight.*

Proof By (4.38) in Lemma 4.5, there is a column I_g of type $\binom{*}{*,\bar{g}}$ in $d(\mathbf{y}, w)$ for each $g \in B_1$. By (4.39) in Lemma 4.5, there is a column I_k of type $\binom{*,\bar{k}}{*}$ in $d(\mathbf{y}, w)$ for each $k \in B_2$.

Consider job g with the largest $x_{i1} + x_{i2}$ among the jobs $i \in B_1 \cup B_2$. Suppose $g \in B_1$. If $g \in B_2 \backslash B_1$, then the proof proceeds in a similar way and thus it will be omitted. Take any $k \in B_2 \setminus \{g\}$ or $k = g$ if $B_2 = \{g\}$. Observe that by our choice of g, if $x_{i1} + x_{i2} = r$ for some $i \in (B_1 \cup B_2) \setminus \{g, k\}$, then $x_{g1} + x_{g2} = r$. Therefore $\{k, i, g\} \subseteq B_1 \cup B_2$ which leads to a contradiction by (4.19) and (4.20) if $k \neq g$. Otherwise, if $k = g$, then by (4.20) $B_1 \cup B_2 = \{i, g\}$ and $g \in B_1 \cap B_2$. Thus $i \in B_1 \cap B_2$ and we get contradiction since $i \notin B_2$. Thus $\varepsilon_r(g, k) > 0$.

If k is not d-tight, then there is a column I of type $\binom{*,\bar{k}}{*,\bar{k}}$ in $d(\mathbf{y}, w)$. Thus, if $I \neq I_g$, then we get a contradiction with Lemma 4.4 applied to I and I_g. Otherwise, if $I = I_g$, then I is of type $\binom{*,\bar{k}}{*,\bar{g}}$ which contradicts Lemma 4.2. Similarly, if g is not d-tight, then there is a column I of type $\binom{*,\bar{g}}{*,\bar{g}}$ in $d(\mathbf{y}, w)$. Thus, if $I \neq I_k$, then we get a contradiction with Lemma 4.4 applied to I_k and I. Otherwise, if $I = I_k$, then I is of type $\binom{*,\bar{k}}{*,\bar{g}}$ which contradicts Lemma 4.2. Therefore the theorem holds for each job in $\{g\} \cup B_2$. Moreover, there is a column I'_g of type $\binom{*,\bar{g}}{*}$. Otherwise all columns in $d(\mathbf{y}, w)$ are of type $\binom{*,g}{*}$ and thus I_k is of type $\binom{*,\bar{k}}{*,\bar{g}}$ for any $k \in B_2$ which contradicts Lemma 4.2.

It remains to prove the theorem for each $a \in B_1 \setminus \{g\}$ whenever $B_1 \setminus \{g\} \neq \emptyset$. Observe that if $x_{g1} + x_{g2} = r$, then $x_{g2} > 0$. Otherwise $B_1 = \{g\}$ and we get a contradiction. Take a job $k = g$, if $x_{g1}+x_{g2} = r$, or any job $k \in B_2$, if $x_{g1}+x_{g2} < r$. W have $\varepsilon_r(a, k) > 0$. This holds since there is no $i \in (B_1 \cup B_2) \setminus \{a, k\}$ that meets $x_{i1} + x_{i2} = r$. Suppose for a contradiction that $x_{i1} + x_{i2} = r$ for some $i \in (B_1 \cup B_2) \setminus \{a, k\}$. Then $x_{k1} + x_{k2} = r$. Since $a \neq k$, we have $\{k, i, a\} \subseteq B_1 \cup B_2$ which leads to a contradiction by (4.19) and (4.20).

Thus if a is not d-tight, then there is a column I of type $\binom{*,\bar{a}}{*,\bar{a}}$ in $d(\mathbf{y}, w)$. Then, if $\varepsilon_r(a, k) > 0$ for $k \in B_2$, we have either $I \neq I_k$ which leads a contradiction with Lemma 4.4 applied to I_k and I or $I = I_k$ which implies that I is of type $\binom{*,\bar{k}}{*,\bar{a}}$ which contradicts Lemma 4.2. If $\varepsilon_r(a,k) > 0$ for $k \notin B_2$, then $k = g$. Thus, if $I \neq I'_g$, then we get a contradiction with Lemma 4.4 applied to I and I'_g. Otherwise, if $I = I'_g$, then I is of type $\binom{*,\overline{g}}{*,\overline{a}}$ which contradicts Lemma 4.2. □

For $j \in B_1 \cup B_2$ define

$$\alpha_j = \sum_{h \in \mathcal{G}_1} y_{jh} \quad \text{and} \quad \beta_j = \sum_{h \in \mathcal{G}_2} y_{jh}.$$

The following two lemmas relate the fractions of x_{j1}, x_{j2}, α_j, and β_j for $j \in B_1 \cup B_2$. The lemmas follow from Lemmas 4.5 and Theorem 4.4 and will prove useful in the remainder of the proof.

Lemma 4.6 *For $g \in B_1$, let*

$$x_{g1} = \lfloor x_{g1} \rfloor + \varepsilon_g \ , \ \ \beta_g = \lfloor \beta_g \rfloor + \lambda_g, \ \ \textit{and} \ \ \alpha_g = \lfloor \alpha_g \rfloor + \omega_g,$$

where $0 \le \lambda_g, \omega_g < 1$, $0 < \varepsilon_g < 1$ for $g \in B_1$. Then, $\omega_g = \varepsilon_g$, and $\lambda_g = \epsilon - \varepsilon_g$ for $\epsilon \ge \varepsilon_g$, and $\lambda_g = 1 - (\varepsilon_g - \epsilon)$ for $\epsilon < \varepsilon_g$.

Proof Details can be found in Kubiak [16]. □

Lemma 4.7 *For $k \in B_2$, let*

$$x_{k2} = \lfloor x_{k2} \rfloor + \varepsilon_k \ \ \textit{and} \ \ \beta_k = \lfloor \beta_k \rfloor + \lambda_k \ \ \textit{and} \ \ \alpha_k = \lfloor \alpha_k \rfloor + \omega_k,$$

where $0 \le \lambda_k, \omega_k < 1$, $0 < \varepsilon_k < 1$ for a job $k \in B_2$. Then, $\lambda_k = \varepsilon_k$, and $\omega_k = \epsilon - \varepsilon_k$ for $\epsilon \ge \varepsilon_k$, and $\lambda_k = 1 - (\varepsilon_k - \epsilon)$ for $\epsilon < \varepsilon_k$.

Proof The proof is similar to the proof of Lemma 4.6 and will be omitted. □

4.4.2 *The Absence of Crossing Jobs in* s

Each job $k \in B_1 \cap B_2$ is called *crossing*. We call a job $a \in B_1 \cup B_2$ an *e-crossing job*, if it meets the following conditions:

- $0 < x_{a2}$ and $0 < x_{a1}$.
- Both $B_1 \setminus \{a\}$ and $B_2 \setminus \{a\}$ are not empty.

We have the following.

Theorem 4.5 *Each crossing job is e-crossing.*

Proof Suppose for a contradiction that job a is crossing but not e-crossing. By Theorem 4.4 job a is d-tight and thus

$$\sum_{h \in \mathcal{G}_2} y_{ah} + \sum_{h \in \mathcal{G}_1} y_{ah} = w. \tag{4.40}$$

By Lemma 4.5 job a is both a-tight and c-tight, thus

$$a_{a1} - x_{a1} + \sum_{h \in \mathcal{G}_2} (b_{ah} - y_{ah}) = \Delta(\mathcal{G}_1) - r \tag{4.41}$$

and

$$a_{a2} - x_{a2} + \sum_{h \in \mathcal{G}_1} (b_{ah} - y_{ah}) = \Delta(\mathcal{G}_2) - r. \tag{4.42}$$

By summing up (4.40), (4.41), and (4.42) side by side we obtain

$$a_{a1} + a_{a2} + \sum_{h} b_{ah} - \Delta(\mathcal{G}_1) - \Delta(\mathcal{G}_2) + r - w = -r + x_{a1} + x_{a2} \,. \tag{4.43}$$

Since a is not e-crossing, $B_1 \setminus \{a\} = \emptyset$ or $B_2 \setminus \{a\} = \emptyset$. Thus, $x_{a1} = \lfloor x_{a1} \rfloor + \epsilon$ or $x_{a2} = \lfloor x_{a2} \rfloor + \epsilon$. Therefore, the left hand side of (4.43) is integral but its right hand side is fractional since both x_{a1} and x_{a2} are fractional. This leads to contradiction and thus the theorem holds. □

Theorem 4.6 *For each e-crossing job a we have* $x_{a1} + x_{a2} < r$.

Proof By contradiction. Let a be e-crossing with $x_{a1} + x_{a2} = r$. Let $g \in B_1 \setminus \{a\}$ and $k \in B_2 \setminus \{a\}$. By Theorem 4.4 and Lemma 4.5 there are columns I_k of type $\binom{*}{*,k}$ and I_g of type $\binom{*,g}{*}$ in $d(\mathbf{y}, w)$. By Theorem 4.4, I_k is either of type $\binom{*}{*,k,g}$ or of type $\binom{*,g}{*,k}$, and I_g is either of type $\binom{*,g,k}{*}$ or of type $\binom{*,g}{*,k}$. Suppose that I_k or I_g is of type $\binom{*,g}{*,k}$, then $g \neq k$. Since a is e-crossing, by Theorem 4.4 this column, say I, is either of type $\binom{*,a,g}{*,k}$ or of type $\binom{*,g}{*,a,k}$. The former is of type $\binom{*,\bar{k}}{*,\bar{a}}$ and the latter of type $\binom{*,\bar{a}}{*,\bar{g}}$. Since $g \neq k$, a is the only job i with $x_{i1} + x_{i2} = r$. Thus $\varepsilon_r(a, k) > 0$ and $\varepsilon_r(g, a) > 0$. Therefore we get a contradiction with Lemma 4.2 which implies that I_g is of type $\binom{*,g,k}{*}$ and I_k is of type $\binom{*}{*,k,g}$ (observe that we may now have $g = k$). Since a is e-crossing, by Theorem 4.4 we have I_g of type $\binom{*,a,g,k}{*}$ or of type $\binom{*,g,k}{*,a}$, and I_k is of type $\binom{*}{*,a,k,g}$ or of type $\binom{*,a}{*,k,g}$. The I_g of type $\binom{*,g,k}{*,a}$ is of type $\binom{*,\bar{a}}{*,\bar{g}}$, and the I_k of type $\binom{*,a}{*,k,g}$ is of type $\binom{*,\bar{k}}{*,\bar{a}}$. Moreover, if $g \neq k$, then a is the only job i with $x_{i1} + x_{i2} = r$, and if $k = g$, then either $x_{k1} + x_{k2} = r$ or a is the only job i with $x_{i1} + x_{i2} = r$. Thus $\varepsilon_r(a, k) > 0$ and $\varepsilon_r(g, a) > 0$. Therefore, I_g being of type $\binom{*,g,k}{*,a}$ or I_k being of type $\binom{*,a}{*,k,g}$ contradicts Lemma 4.2. Thus it remains to consider I_g of type $\binom{*,a,g,k}{*}$ and I_k is of type $\binom{*}{*,a,k,g}$. This leads to a contradiction by Corollaries 4.1 and 4.2 since $\varepsilon_r(g, a) >$ and $\varepsilon_r(a, k) > 0$. The last two inequalities clearly hold if a is the only job i with $x_{i1} + x_{i2} = r$, otherwise $g = k$ and k is the other job i with $x_{i1} + x_{i2} = r$. □

The following corollary follows immediately from the proof of Theorem 4.6 since the assumption $x_{i1} + x_{i2} < r$ for each $i \in B_1 \cup B_2$ implies $\varepsilon_r(g, k) > 0$ for each $g \in B_1$ and $k \in B_2$.

Corollary 4.3 *If $x_{i1} + x_{i2} < r$ for each $i \in B_1 \cup B_2$, then no job is e-crossing.*

We are now ready to prove two main results of this section.

Theorem 4.7 *No crossing job exists.*

Proof By contradiction. Suppose a is a crossing job. Take a crossing job with the largest $x_{a1}+x_{a2}$. By Theorem 4.5 a is e-crossing, and by Theorem 4.6 $x_{a1}+x_{a2} < r$. By Corollary 4.3 $x_{i1}+x_{i2} = r$ for some $i \in B_1 \cup B_2$. Thus $i \neq a$. By Theorem 4.6 i is not e-crossing. Thus ($x_{i1} = 0$ or $x_{i2} = 0$) which implies ($B_1 = \{i\}$ or $B_2 = \{i\}$). This leads to contradiction since $a \in B_1 \cap B_2$ and $a \neq i$. □

Theorem 4.8 *For each $g \in B_1$ and $k \in B_2$, $\varepsilon_r(g, k) > 0$.*

Proof Suppose for a contradiction that $\varepsilon_r(g, k) = 0$ for some $g \in B_1$ and $k \in B_2$. By Theorem 4.7, $g \neq k$. Then $r = x_{j1} + x_{j2}$ for some $j \in (B_1 \cup B_2) \setminus \{g, k\}$. By Theorem 4.7, j is not crossing thus $\{j, g\} \subseteq B_1$ and $j \notin B_2$, or $\{j, k\} \subseteq B_2$ and $j \notin B_1$. Suppose the former, the proof for the latter is similar and thus will be omitted. We have x_{j2} integral. However, by Theorem 4.6 j is not e-crossing. Hence $x_{j2} = 0$. Thus $r = x_{j1}$ and $B_1 = \{j\}$ which gives a contradiction. □

4.5 Characterization of $d(\mathbf{y}, w)$ in s

We give a characterization of $d(\mathbf{y}, w)$ that will be used in the remainder of the proof.

Lemma 4.8 *For each $g \in B_1$ and $k \in B_2$, any column I in $d(\mathbf{y}, w)$ is either of type $\binom{*,k}{*,g}$ or of type $\binom{*}{*,k,g}$ or of type $\binom{*,k,g}{*}$. Moreover, for each $g \in B_1$ and $k \in B_2$ there is I_k of type $\binom{*}{*,k,g}$, and there is I_g of type $\binom{*,k,g}{*}$ in $d(\mathbf{y}, w)$. Finally, if there is $i \in B_1 \cup B_2$ such that $x_{i1} + x_{i2} = r$, then either $B_1 = \{i\}$ or $B_2 = \{i\}$.*

Proof Let $g \in B_1$ and $k \in B_2$. By Lemma 4.5 and Theorem 4.4 each column I in $d(\mathbf{y}, w)$ is either of type $\binom{*,k}{*}$ or of type $\binom{*}{k,*}$. By Theorem 4.4 I is either of type $\binom{*,k,g}{*}$ or of type $\binom{*,k}{*,g}$, or of type $\binom{*}{*,g,k}$ or of type $\binom{*,g}{*,k}$. By Theorem 4.8 we have $\varepsilon_r(g, k) > 0$ and thus by Lemma 4.2 I is *not* of type $\binom{*,g}{*,k}$. This proves the first part of the lemma. Again, by Lemma 4.5 and Theorem 4.4 there are columns I_k of type $\binom{*}{*,k}$ and I_g of type $\binom{*,g}{*}$ in $d(\mathbf{y}, w)$. By Theorem 4.4 I_k is either of type $\binom{*}{*,k,g}$ or of type $\binom{*,g}{*,k}$, and I_g is either of type $\binom{*,g,k}{*}$ or of type $\binom{*,g}{*,k}$. By Theorem 4.8 we have $\varepsilon_r(g, k) > 0$ and thus by Lemma 4.2 neither I_k nor I_g is of type $\binom{*,g}{*,k}$. This proves the second part of the lemma.

If there is $i \in B_1 \cup B_2$ such that $x_{i1} + x_{x2} = r$. By Theorem 4.6 i is not e-crossing thus $x_{i1} = 0$ or $x_{i2} = 0$ or $B_1 \setminus \{i\} = \emptyset$ or $B_2 \setminus \{i\} = \emptyset$. In all the cases, either $B_1 = \{i\}$ or $B_2 = \{i\}$. This completes the proof. □

Theorem 4.9 *If there is a job j such that $x_{j1}x_{j2} > 0$, then $B_1 = \{j\}$ or $B_2 = \{j\}$.*

Proof Let $x_{j1}x_{j2} > 0$ for a job j. Without loss of generality let j be a job with the largest value of $x_{j1} + x_{j2}$ among jobs with $x_{j1}x_{j2} > 0$. Suppose for a contradiction that $B_1 \setminus \{j\} \neq \emptyset$ and $B_2 \setminus \{j\} \neq \emptyset$. Thus if $j \in B_1 \cup B_2$, then j is e-crossing. By Theorem 4.6, $x_{j1} + x_{j2} < r$. Take $g \in B_1 \setminus \{j\}$ and $k \in B_2 \setminus \{j\}$. If $j \notin B_1 \cup B_2$, then both x_{j1} and x_{j2} are integral. Thus $x_{j1} + x_{j2} < r$. Take $g \in B_1$ and $k \in B_2$. Thus we can pick three jobs $g \in B_1$, $k \in B_2$, and j such that $x_{j1} + x_{j2} < r$ and $g \neq j$ and $k \neq j$. Moreover, by Theorem 4.7 we have $g \neq k$. We now show that $\varepsilon_r(g, j) > 0$ and $\varepsilon_r(j, k) > 0$. To prove the former inequality we observe that by our choice of job j for any job $i \in B_1 \cup B_2$ different from g and j, and such that $x_{i1} + x_{i2} = r$ must be either $r = x_{i1}$ or $r = x_{i2}$. Otherwise $x_{i1}x_{i2} > 0$, thus i would have been chosen instead of j. The proof of the latter inequality follows by a similar argument. Thus by Corollary 4.1 a column of type $\binom{*,j,g,k}{*}$ does not exist in $d(\mathbf{y}, w)$ or a column of type $\binom{*}{*,j,k,g}$ does not exist in $d(\mathbf{y}, w)$. Suppose the former holds, then by Lemma 4.8 a column of type $\binom{*,g,k}{*}$ exists in $d(\mathbf{y}, w)$. This column is either of type $\binom{*,g,k}{*,j}$ or of type $\binom{*,\bar{j},g,k}{*,\bar{j}}$ which implies that the column is of type $\binom{*,\bar{j}}{*,\bar{g}}$. This however contradicts Lemma 4.2. For the latter, we prove in a similar fashion that a column of type $\binom{*,\bar{k}}{*,\bar{j}}$ exists in $d(\mathbf{y}, w)$ which contradicts Lemma 4.2. Therefore we get a contradiction which proves the theorem. □

4.5.1 The Overlap

An overlap of B_1 is a column $I = (M_I, \varepsilon) \in d(\mathbf{y}, w)$ that matches at least two different jobs from B_1 with machines in $\mathcal{G}_1$. Similarly, an overlap of B_2 is a column $I = (M_I, \varepsilon) \in d(\mathbf{y}, w)$ that matches at least two different jobs from B_2 with machines in $\mathcal{G}_2$.

Lemma 4.9 *An overlap of B_1 and an overlap of B_2 do not occur simultaneously.*

Proof Suppose for contradiction that both overlaps occur simultaneously. Then there are different jobs a and g both from B_1 done on $\mathcal{G}_1$ in a column $I_{a,g} \in d(\mathbf{y}, w)$ of type $\binom{*,a,g}{*}$, and different jobs b and k both from B_2 done on $\mathcal{G}_2$ in a column $I_{b,k} \in d(\mathbf{y}, w)$ of type $\binom{*}{*,b,k}$. By Theorem 4.7 there are no crossing jobs thus all four jobs a, g, b, and k are different. On the other hand for $g \in B_1$ and $k \in B_2$, by Lemma 4.8, any column I in $d(\mathbf{y}, w)$ is either of type $\binom{*,k}{*,g}$ or of type $\binom{*}{*,k,g}$ or of type $\binom{*,k,g}{*}$. Thus $I_{a,g}$ must be of type $\binom{*,a,k,g}{*}$. For $a \in B_1$ and $b \in B_2$, again by Lemma 4.8, any column I in $d(\mathbf{y}, w)$ is either of type $\binom{*,b}{*,a}$ or of type $\binom{*}{*,b,a}$ or of type $\binom{*,b,a}{*}$. Thus $I_{a,g}$ must be of type $\binom{*,a,b,g}{*}$. Therefore $I_{a,g}$ is of type $\binom{*,a,b,k,g}{*}$. We show similarly that $I_{b,k}$ is of type $\binom{*}{*,a,b,k,g}$. This, by Theorem 4.8, contradicts Lemma 4.3 and proves the lemma. □

4.6 Integral Optimal Solution to ℓp for $\sum_{j\in B_1}\varepsilon_j = \epsilon$ or $\sum_{j\in B_2}\varepsilon_j = \epsilon$

In this section we prove that an integral optimal solution for ℓp exists if $\epsilon > 0$ and $\sum_{j\in B_1}\varepsilon_j = \epsilon$ or $\sum_{j\in B_2}\varepsilon_j = \epsilon$. We first prove this assuming $\sum_{j\in B_2}\varepsilon_j = \epsilon$ in **s** throughout this section. The proof for $\sum_{j\in B_1}\varepsilon_j = \epsilon$ proceeds in a similar fashion and thus will be omitted.

Consider the following network-flow problem $\mathcal{F}$ with variables t_{jh} for j and $h \in \mathcal{G}_2$, and z_{jh} for j and $h \in \mathcal{G}_1$. The r, w, and $x_{j\ell}$ for $j \in \mathcal{J}$ and $\ell = 1, 2$ in $\mathcal{F}$ are constants obtained from the solution $\mathbf{s} = (\mathbf{y}, \mathbf{x}, r, w)$.

$$F = \max \sum_{j\in B_1}\sum_{h\in\mathcal{G}_2} t_{jh}.$$

Subject to

$$\sum_j t_{jh} = \lfloor w \rfloor \quad h \in \mathcal{G}_2 \tag{4.44}$$

$$\sum_{h\in\mathcal{G}_2} b_{jh} + a_{j1} - \Delta(\mathcal{G}_1) + \lfloor r\rfloor - x_{j1} \le \sum_{h\in\mathcal{G}_2} t_{jh} \quad j \in \mathcal{J}\setminus B_1 \tag{4.45}$$

$$\sum_{h\in\mathcal{G}_2} b_{jh} + a_{j1} - \Delta(\mathcal{G}_1) + \lfloor r\rfloor - \lceil x_{j1}\rceil \le \sum_{h\in\mathcal{G}_2} t_{jh}$$

$$\le \sum_{h\in\mathcal{G}_2} b_{jh} + a_{j1} - \Delta(\mathcal{G}_1) + \lfloor r\rfloor - \lfloor x_{j1}\rfloor \quad j \in B_1 \tag{4.46}$$

$$\sum_j z_{jh} = \lfloor w\rfloor \quad h \in \mathcal{G}_1 \tag{4.47}$$

$$\sum_{h\in\mathcal{G}_1} b_{jh} + a_{j2} - \Delta(\mathcal{G}_2) + \lfloor r\rfloor - \lfloor x_{j2}\rfloor \le \sum_{h\in\mathcal{G}_1} z_{jh} \quad j \in \mathcal{J} \tag{4.48}$$

$$0 \le t_{jh} \le b_{jh} \qquad h \in \mathcal{M} \qquad j \in \mathcal{J} \tag{4.49}$$

$$0 \le z_{jh} \le b_{jh} \qquad h \in \mathcal{M} \qquad j \in \mathcal{J} \tag{4.50}$$

$$\sum_{h\in\mathcal{G}_1} z_{jh}+\sum_{h\in\mathcal{G}_2} t_{jh}\le\lfloor w\rfloor \quad j\in\mathcal{J}. \tag{4.51}$$

Lemma 4.10 *There is a feasible solution to $\mathcal{F}$ with value*

$$\sum_{j\in B_1}\sum_{h\in\mathcal{G}_2} b_{jh}-\sum_{j\in\mathcal{J}\setminus B_1}(a_{j1}-x_{j1})-(|B_1|-1)(\Delta(\mathcal{G}_1)-\lfloor r\rfloor)-\epsilon. \tag{4.52}$$

Proof For **s**, consider the set Y_j of all columns of type $\binom{*}{*,j}$ in $d(\mathbf{y},w)$ for $j\in B_2$. By Lemma 4.7, $l(Y_j)=\beta_j=\lfloor\beta_j\rfloor+\varepsilon_j$. If there is no overlap of B_2 or $\sum_{j\in B_2}\lfloor\beta_j\rfloor>0$, then take an interval $Y\subseteq\bigcup_{j\in B_2}Y_j$ such that $l(Y)=\epsilon$, $l(Y\cap Y_j)\ge\varepsilon_j$ for $j\in B_2$. Otherwise, if there is overlap of B_2 and $\sum_{j\in B_2}\lfloor\beta_j\rfloor=0$, then take an interval $Y\subseteq(\bigcup_{j\in B_2}Y_j)\cup Z$ such that $l(Y)=\epsilon$, $l(Y\cap Y_j)\ge\varepsilon_j$ for $j\in B_2$. Here the Z is the set of all columns of type $\binom{*,B_2}{*,B_1}$ in $d(\mathbf{y},w)$. In order for such Y to exist we show that $l((\bigcup_{j\in B_2}Y_j)\cup Z)\ge 1$. By Lemma 4.9 there is no overlap of B_1, thus $l(\bigcup_{j\in B_1}W_j)=\sum_{j\in B_1}l(W_j)=\sum_{j\in B_1}\alpha_j$ where W_j is the set of all columns of type $\binom{*,j}{*}$ for $j\in B_1$ in $d(\mathbf{y},w)$. Hence by Lemma 4.6, $l(\bigcup_{j\in B_1}W_j)=\sum_{j\in B_1}\lfloor\alpha_j\rfloor+\sum_{j\in B_1}\varepsilon_j$. By definition $\sum_{j\in B_1}\varepsilon_j=i_1+\epsilon$ for some integer $i_1\ge 0$. Therefore $l(\bigcup_{j\in B_1}W_j)=i+\epsilon$ for some integer $i\ge 0$. Thus $l(d(\mathbf{y},w)\setminus\bigcup_{j\in B_1}W_j)$ is integral since $l(d(\mathbf{y},w))=w$, and positive. However $d(\mathbf{y},w)\setminus\bigcup_{j\in B_1}W_j=(\bigcup_{j\in B_2}Y_j)\cup Z$ by Theorem 4.4 and Lemma 4.8. This proves $l((\bigcup_{j\in B_2}Y_j)\cup Z)\ge 1$, and the required Y exists.

Let Y_{jh} be the set of columns $I\in Y$ such that $(j,h)\in M_I$, set $\gamma_{jh}:=l(Y_{jh})$. Informally, γ_{jh} is the amount of $j\in\mathcal{J}$ done on $h\in\mathcal{M}$ in the interval Y. We define a truncated solution as follows: $z^*_{jh}:=y_{jh}-\gamma_{jh}$ for $h\in\mathcal{G}_1$, and $t^*_{jh}:=y_{jh}-\gamma_{jh}$ for $h\in\mathcal{G}_2$. By Theorem 4.4 each $j\in B_2$ is d-tight thus

$$\sum_{h\in\mathcal{G}_1}\gamma_{jh}+\sum_{h\in\mathcal{G}_2}\gamma_{jh}=\epsilon \quad j\in B_2 \tag{4.53}$$

and

$$\sum_{h\in\mathcal{G}_2}\gamma_{jh}=\eta_j\ge\varepsilon_j \quad j\in B_2. \tag{4.54}$$

We prove that this truncated solution is feasible for $\mathcal{F}$ and meets (4.52). □

We first prove the following lemma.

Lemma 4.11 *If $\sum_{j\in B_2}\varepsilon_j=\epsilon$, then truncated solution meets (4.48).* □

Proof We have the following for the truncated solution:

$$\sum_{h\in\mathcal{G}_1} z^*_{jh} = \sum_{h\in\mathcal{G}_1} y_{jh} - (\epsilon - \eta_j) \quad j \in B_2. \tag{4.55}$$

By Lemma 4.5 each $j \in B_2$ is c-tight. Thus we get

$$\sum_{h\in\mathcal{G}_1} y_{jh} = \sum_{h\in\mathcal{G}_1} b_{jh} + a_{j2} - \Delta(\mathcal{G}_2) - \lfloor x_{j2}\rfloor + \lfloor r\rfloor + \epsilon - \varepsilon_j \quad j \in B_2. \tag{4.56}$$

Therefore by (4.55) and (4.56) we get

$$\sum_{h\in\mathcal{G}_1} z^*_{jh} + (\varepsilon_j - \eta_j) = \sum_{h\in\mathcal{G}_1} b_{jh} + a_{j2} - \Delta(\mathcal{G}_2) - \lfloor x_{j2}\rfloor + \lfloor r\rfloor \quad j \in B_2,$$

and by (4.54)

$$\sum_{h\in\mathcal{G}_1} z^*_{jh} \geq \sum_{h\in\mathcal{G}_1} b_{jh} + a_{j2} - \Delta(\mathcal{G}_2) - \lfloor x_{j2}\rfloor + \lfloor r\rfloor \quad j \in B_2,$$

which proves (4.48) holds for $j \in B_2$ in the truncated solution $\mathbf{t}^*$ and $\mathbf{z}^*$. For $j \in \mathcal{J} \setminus B_2$, x_{j2} is integral thus

$$\sum_{h\in\mathcal{G}_1} z^*_{jh} \geq \sum_{h\in\mathcal{G}_1} b_{jh} + a_{j2} - \Delta(\mathcal{G}_2) - \lfloor x_{j2}\rfloor + \lfloor r\rfloor + \epsilon - \sum_{h\in\mathcal{G}_1} \gamma_{jh} \quad j \in \mathcal{J} \setminus B_2,$$

since $\epsilon - \sum_{h\in\mathcal{G}_1} \gamma_{jh} \geq 0$ for $j \in \mathcal{J}$ we get

$$\sum_{h\in\mathcal{G}_1} z^*_{jh} \geq \sum_{h\in\mathcal{G}_1} b_{jh} + a_{j2} - \Delta(\mathcal{G}_2) - \lfloor x_{j2}\rfloor + \lfloor r\rfloor \quad j \in \mathcal{J} \setminus B_2.$$

Thus (4.48) holds for $j \in \mathcal{J}$. □

Let $\mathbf{t}^*$ and $\mathbf{z}^*$ be a solution of Lemma 4.11. The $\mathbf{t}^*$ and $\mathbf{z}^*$ clearly meet (4.44), (4.47), (4.49), (4.50), (4.51). By Lemma 4.11 (4.48) holds. Then (4.45) also holds for $\mathbf{t}^*$ and $\mathbf{z}^*$. To show that we observe that by feasibility of $\mathbf{s} = (\mathbf{y}, \mathbf{x}, r, w)$ we have

$$\sum_{h\in\mathcal{G}_2} b_{jh} + a_{j1} - x_{j1} - \Delta(\mathcal{G}_1) + r \leq \sum_{h\in\mathcal{G}_2} (y_{jh} - t^*_{jh}) + \sum_{h\in\mathcal{G}_2} t^*_{jh} \quad j \in \mathcal{J} \setminus B_1.$$

Since for $\mathbf{t}^*$ we have

$$0 \leq \sum_{h\in\mathcal{G}_2} (y_{jh} - t^*_{jh}) \leq \epsilon \quad j \in \mathcal{J},$$

and x_{j1} is integral for $\mathcal{J} \setminus B_1$, the $\mathbf{t}^*$ satisfies the (4.45).

To prove (4.46) we observe that by Lemma 4.5 each $j \in B_1$ is a-tight and thus

$$\sum_{h\in\mathcal{G}_2} b_{jh}+a_{j1}-x_{j1}-\Delta(\mathcal{G}_1)+r=\sum_{h\in\mathcal{G}_2}(y_{jh}-t^*_{jh})+\sum_{h\in\mathcal{G}_2} t^*_{jh} \quad j\in B_1. \tag{4.57}$$

By Theorem 4.4 $j \in B_1$ is d-tight. Thus by Lemma 4.2 and definition of truncated solution we have

$$\epsilon=\sum_{h\in\mathcal{G}_2}(y_{jh}-t^*_{jh}), \tag{4.58}$$

for $j \in B_1$.

Thus by (4.57) and (4.58)

$$\sum_{h\in\mathcal{G}_2} b_{jh}+a_{j1}-\Delta(\mathcal{G}_1)+\lfloor r\rfloor-\lfloor x_{j1}\rfloor+\epsilon-\epsilon-\varepsilon_j=\sum_{h\in\mathcal{G}_2} t^*_{jh} \quad j\in B_1.$$

Hence (4.46) is met by the truncated solution $\mathbf{t}^*$ and $\mathbf{z}^*$. Therefore the truncated solution $\mathbf{t}^*$ and $\mathbf{z}^*$ is feasible for $\mathcal{F}$.

To prove the lower bound on the value of objective function we observe that by (4.57) and (4.58)

$$\sum_{h\in\mathcal{G}_2} b_{jh}+a_{j1}-x_{j1}-\Delta(\mathcal{G}_1)+\lfloor r\rfloor+\epsilon-\epsilon=\sum_{h\in\mathcal{G}_2} t^*_{jh} \quad j\in B_1. \tag{4.59}$$

Summing up (4.59) side by side over all $j \in B_1$ we get by (4.19) for $(\mathbf{y},\mathbf{x},r,w)$

$$\sum_{j\in B_1}\Big(\sum_{h\in\mathcal{G}_2} b_{jh}+a_{j1}\Big)-(r-c)-|B_1|(\Delta(\mathcal{G}_1)-\lfloor r\rfloor)=\sum_{j\in B_1}\sum_{h\in\mathcal{G}_2} t^*_{jh},$$

where $c=\sum_{j\in\mathcal{J}\setminus B_1} x_{j1}$ is integral by definition of B_1. Thus

$$\sum_{j\in B_1}\sum_{h\in\mathcal{G}_2} b_{jh}+\Delta(\mathcal{G}_1)-\lfloor r\rfloor-\sum_{j\in\mathcal{J}\setminus B_1}(a_{j1}-x_{j1})-|B_1|(\Delta(\mathcal{G}_1)-\lfloor r\rfloor)-\epsilon=\sum_{j\in B_1}\sum_{h\in\mathcal{G}_2} t^*_{jh}$$

and

$$\sum_{j\in B_1}\sum_{h\in\mathcal{G}_2} b_{jh}-\sum_{j\in\mathcal{J}\setminus B_1}(a_{j1}-x_{j1})-(|B_1-1)|(\Delta(\mathcal{G}_1)-\lfloor r\rfloor)-\epsilon=\sum_{j\in B_1}\sum_{h\in\mathcal{G}_2} t^*_{jh}$$

as required.

Lemma 4.12 *If* $\sum_{j\in B_1}\varepsilon_j=\epsilon$*, then*

$$F = \sum_{j\in B_1}\sum_{h\in \mathcal{G}_2} b_{jh} + \sum_{j\in B_1}(a_{j1} - \lfloor x_{j1}\rfloor) - |B_1|(\Delta(\mathcal{G}_1) - \lfloor r\rfloor) \tag{4.60}$$

and

$$\sum_{h\in \mathcal{G}_2} t_{jh} = \sum_{h\in \mathcal{G}_2} b_{jh} + a_{j1} - \lfloor x_{j1}\rfloor - \Delta(\mathcal{G}_1) + \lfloor r\rfloor \quad j \in B_1. \tag{4.61}$$

Proof By (4.59)

$$\sum_{h\in \mathcal{G}_2} b_{jh} + a_{j1} - \Delta(\mathcal{G}_1) + \lfloor r\rfloor - \lfloor x_{j1}\rfloor - \varepsilon_j = \sum_{h\in \mathcal{G}_2} t^*_{jh} \quad j \in B_1, \tag{4.62}$$

summing up side by side for $j \in B_1$ and taking $\sum_{j\in B_1} \varepsilon_j = \epsilon$ we get

$$\sum_{j\in B_1}\sum_{h\in \mathcal{G}_2} b_{jh} + \sum_{j\in B_1}(a_{j1} - \lfloor x_{j1}\rfloor) - |B_1|(\Delta(\mathcal{G}_1) - \lfloor r\rfloor) - \epsilon = \sum_{j\in B_1}\sum_{h\in \mathcal{G}_2} t^*_{jh}, \tag{4.63}$$

for the truncated solution $\mathbf{t}^*$ and $\mathbf{z}^*$, which by Lemma 4.10 is feasible for $\mathcal{F}$. Let $\mathbf{t}$ and $\mathbf{z}$ be an optimal solution for $\mathcal{F}$. Since all upper and lower bounds in $\mathcal{F}$ are integral, we may assume both $\mathbf{t}$ and $\mathbf{z}$ integral by the Integral Circulation Theorem, see Lawler [17]. Thus by (4.63)

$$\sum_{j\in B_1}\sum_{h\in \mathcal{G}_2} b_{jh} + \sum_{j\in B_1}(a_{j1} - \lfloor x_{j1}\rfloor) - |B_1|(\Delta(\mathcal{G}_1) - \lfloor r\rfloor) \le \sum_{j\in B_1}\sum_{h\in \mathcal{G}_2} t_{jh}, \tag{4.64}$$

and the upper bounds in (4.46) give

$$\sum_{j\in B_1}\sum_{h\in \mathcal{G}_2} b_{jh} + \sum_{j\in B_1}(a_{j1} - \lfloor x_{j1}\rfloor) - |B_1|(\Delta(\mathcal{G}_1) - \lfloor r\rfloor) \ge \sum_{j\in B_1}\sum_{h\in \mathcal{G}_2} t_{jh}. \tag{4.65}$$

Hence by (4.64) and (4.65) we get

$$\sum_{j\in B_1}\sum_{h\in \mathcal{G}_2} b_{jh} + \sum_{j\in B_1}(a_{j1} - \lfloor x_{j1}\rfloor) - |B_1|(\Delta(\mathcal{G}_1) - \lfloor r\rfloor) = \sum_{j\in B_1}\sum_{h\in \mathcal{G}_2} t_{jh} = F,$$

which proves (4.60) in the lemma. Finally, in order to reach this optimal value all upper bounds in (4.46) must be reached, which proves (4.61). □

Theorem 4.10 *For $\sum_{j\in B_2} \varepsilon_j = \epsilon$, an optimal solution to $\mathcal{F}$ can be extended to an integral feasible solution to ℓp with $lp = \lfloor r\rfloor < r$.*

Proof Let $\mathbf{t}$ and $\mathbf{z}$ be an optimal solution to $\mathcal{F}$. This solution exists since by Lemma 4.10 there is a feasible solution to $\mathcal{F}$. Since all upper and lower bounds in $\mathcal{F}$ are integral, we may assume both $\mathbf{t}$ and $\mathbf{z}$ integral by the Integral Circulation

Theorem, see Lawler [17]. Thus by Lemma 4.10

$$\sum_{j\in B_1}\sum_{h\in\mathcal{G}_2} t_{jh} \geq \sum_{j\in B_1}\sum_{h\in\mathcal{G}_2} b_{jh} - \sum_{j\in\mathcal{J}\setminus B_1}(a_{j1}-x_{j1}) - (|B_1|-1)(\Delta(\mathcal{G}_1)-\lfloor r\rfloor). \tag{4.66}$$

For the partial solution $((\mathbf{t},\mathbf{z}), r'=\lfloor r\rfloor, w'=\lfloor w\rfloor)$ we have (4.51) implies (4.17), (4.49) and (4.50) imply (4.18), (4.44) implies (4.16), and (4.47) implies (4.15). To prove the last two implications we observe that

$$\sum_j b_{jh} - \Delta(\mathcal{G}_2) + \lfloor r\rfloor \leq \lfloor w\rfloor \qquad h\in\mathcal{G}_1$$

and

$$\sum_j b_{jh} - \Delta(\mathcal{G}_1) + \lfloor r\rfloor \leq \lfloor w\rfloor \qquad h\in\mathcal{G}_2$$

for $\mathbf{s}$. The (4.44) guarantees

$$\sum_j t_{jh} = \lfloor w\rfloor \quad h\in\mathcal{G}_2,$$

and (4.47) guarantees

$$\sum_j z_{jh} = \lfloor w\rfloor \quad h\in\mathcal{G}_1.$$

Therefore (4.16) and (4.15) are satisfied by the partial solution $((\mathbf{t},\mathbf{z}), r'=\lfloor r\rfloor, w'=\lfloor w\rfloor)$.

Let us now extend the solution $((\mathbf{t},\mathbf{z}), r'=\lfloor r\rfloor, w'=\lfloor w\rfloor)$ by setting $x^*_{j2}:=\lfloor x_{j2}\rfloor$, for $j\in B_2$ and $x^*_{j2}:=x_{j2}$ for $j\in\mathcal{J}\setminus B_2$. Since $\sum_{j\in B_2}\varepsilon_j=\epsilon$, (4.20) is met by this extension. Clearly (4.22) is also met for $\ell=2$. By (4.48) we have

$$\sum_{h\in\mathcal{G}_1} b_{jh} + a_{j2} - \lfloor x_{j2}\rfloor - \Delta(\mathcal{G}_2) + \lfloor r\rfloor \leq \sum_{h\in\mathcal{G}_1} z_{jh} \quad j\in\mathcal{J}.$$

Thus (4.23) is met for the extended solution $((\mathbf{t},\mathbf{z}), r'=\lfloor r\rfloor, w'=\lfloor w\rfloor)$, and x^*_{j2} for $j\in\mathcal{J}$.

We now extend this solution further by setting

$$x^*_{j1} := \sum_{h\in\mathcal{G}_2} b_{jh} + a_{j1} - \Delta(\mathcal{G}_1) + \lfloor r\rfloor - \sum_{h\in\mathcal{G}_2} t_{jh} \tag{4.67}$$

for $j \in B_1$ and $x^*_{j1} := x_{j1}$ for $j \in \mathcal{J} \setminus B_1$. To prove that (4.24) is met for the extended solution $((\mathbf{t}, \mathbf{z}), r' = \lfloor r \rfloor, w' = \lfloor w \rfloor)$, and x^*_{j2}, x^*_{j1} for $j \in \mathcal{J}$ we need to show that

$$\sum_{h \in \mathcal{G}_2} b_{jh} + a_{j1} - x^*_{j1} - \Delta(\mathcal{G}_1) + \lfloor r \rfloor \leq \sum_{h \in \mathcal{G}_2} t_{jh} \tag{4.68}$$

for each $j \in \mathcal{J}$. By the definition (4.67) this holds for $j \in B_1$. For $j \in \mathcal{J} \setminus B_1$ we have x_{j1} integral and thus (4.68) holds since (4.45) holds. Thus (4.24) is met for the extended solution $((\mathbf{t}, \mathbf{z}), r' = \lfloor r \rfloor, w' = \lfloor w \rfloor)$, and x^*_{j2}, x^*_{j1} for $j \in \mathcal{J}$. Moreover $a_{j1} \geq x^*_{j1} \geq 0$ for each $j \in \mathcal{J} \setminus B_1$ and thus (4.22) holds for $\ell = 1$ in this extended solution. It suffices to prove this for $j \in B_1$.

Then, since $\lfloor r \rfloor \geq \lfloor r \rfloor - \lfloor x_{j1} \rfloor$, $x^*_{j1} \geq 0$ by (4.67) and the right hand side inequality of (4.46). Moreover, $a_{j1} \geq \lceil x_{j1} \rceil$. Thus by the left hand side inequality of (4.46)

$$\sum_{h \in \mathcal{G}_2} b_{jh} - \Delta(\mathcal{G}_1) + \lfloor r \rfloor \leq \sum_{h \in \mathcal{G}_2} t_{jh}$$

and by (4.67)

$$x^*_{j1} = \sum_{h \in \mathcal{G}_2} b_{jh} - \Delta(\mathcal{G}_1) + \lfloor r \rfloor - \sum_{h \in \mathcal{G}_2} t_{jh} + a_{j1} \leq a_{j1}.$$

Therefore (4.22) holds for $\ell = 1$ for $j \in B_1$. For $j \in \mathcal{J} \setminus B_1$ the (4.22) for $\ell = 1$ in the extended solution $((\mathbf{t}, \mathbf{z}), r' = \lfloor r \rfloor, w' = \lfloor w \rfloor)$, and x^*_{j2}, x^*_{j1} follows from (4.22) for $\ell = 1$ in the solution $(\mathbf{y}, \mathbf{x}, r, w)$.

By definition of the extended solution $((\mathbf{t}, \mathbf{z}), r' = \lfloor r \rfloor, w' = \lfloor w \rfloor)$, and x^*_{j2}, x^*_{j1} for $j \in \mathcal{J}$, and since by Theorem 4.7 there are no crossing jobs we have

$$x^*_{j1} + x^*_{j2} \leq \lfloor r \rfloor \tag{4.69}$$

for $j \in \mathcal{J} \setminus B_1$. We now need to show this inequality for $j \in B_1$. For these jobs by the left hand side inequality of (4.46), and by (4.67) we get $x^*_{j1} - \lfloor r \rfloor + \lfloor r \rfloor - \lceil x_{j1} \rceil \leq 0$. Thus $x^*_{j1} \leq \lceil x_{j1} \rceil$ for each job $j \in B_1$. This unfortunately does not guarantee (4.69) for $j \in B_1$. However, we either have $\lceil x_{j1} \rceil + x_{j2} \leq \lfloor r \rfloor$ for each $j \in B_1$, in which case (4.69) holds for $j \in B_1$, or $\lceil x_{k1} \rceil + x_{k2} > \lfloor r \rfloor$ for some $k \in B_1$. The latter implies $\sum_{j \in B_1} \varepsilon_j = \epsilon$, which by Lemma 4.12, implies

$$\sum_{h \in \mathcal{G}_2} t_{jh} = \sum_{h \in \mathcal{G}_2} b_{jh} + a_{j1} - \Delta(\mathcal{G}_1) + \lfloor r \rfloor - \lfloor x_{j1} \rfloor \quad j \in B_1$$

in the optimal solution $\mathbf{t}$ and $\mathbf{z}$ to $\mathcal{F}$. Thus by definition (4.67), $x^*_{j1} = \lfloor x_{j1} \rfloor$ for $j \in B_1$. Since by Theorem 4.7 there are no crossing jobs the (4.69) is satisfied. Hence it remains to prove that if $\lceil x_{k1}\rceil + x_{k2} > \lfloor r\rfloor$ for some $k\in B_1$, then $\sum_{j\in B_1}\varepsilon_j=\epsilon$. For contradiction assume $\lceil x_{k1}\rceil + x_{k2} > \lfloor r\rfloor$ for some $k\in B_1$ and $\sum_{j\in B_1}\varepsilon_j>\epsilon$. If $x_{j1}x_{j2}=0$ for each $j\in\mathcal{J}$, then $x_{k2}=0$. Thus $\lceil x_{k1}\rceil > \lfloor r\rfloor$ which implies $\sum_{j\in B_1}\varepsilon_j=\epsilon$ and gives contradiction. Otherwise, if $x_{i1}x_{i2}>0$ for some $i\in\mathcal{J}$, then by Theorem 4.9 we have $B_1=\{i\}$ or $B_2=\{i\}$. If $B_1=\{i\}$, then $\sum_{j\in B_1}\varepsilon_j=\epsilon$ which gives contradiction. Hence $B_2=\{i\}$ and $x_{j2}=0$ for each $j\in B_1$. Since by Theorem 4.7 there are no crossing jobs and x_{i1} is integral and positive. Thus $x_{i1}\ge 1$, and $i\ne k$. By (4.19) $\sum_j x_{j1}=\sum_{j\ne i}x_{j1}+x_{i1}=r$. Hence $\sum_{j\ne i}x_{j1}\le r-1$ which gives $x_{k1}\le r-1$. Since $x_{k2}=0$, we get $x_{k1}+1+x_{k2}\le r$. Thus $\lceil x_{k1}\rceil+x_{k2}\le\lfloor r\rfloor$ which again gives contradiction. This proves that if $\lceil x_{k1}\rceil+x_{k2}>\lfloor r\rfloor$ for some $k\in B_1$, then $\sum_{j\in B_1}\varepsilon_j=\epsilon$ as required. Hence (4.21) holds for the extended solution $((\mathbf{t},\mathbf{z}), r'=\lfloor r\rfloor, w'=\lfloor w\rfloor)$, and x^*_{j2}, x^*_{j1}.

Finally we need to prove that (4.19) holds for an extended solution. By (4.67) and (4.66)

$$\sum_j x^*_{j1}\le\lfloor r\rfloor \tag{4.70}$$

for the extended solution $((\mathbf{t},\mathbf{z},\lfloor r\rfloor,\lfloor w\rfloor))$, and x^*_{j2}, x^*_{j1} for $j\in\mathcal{J}$. This solution satisfies all constraints (4.15)–(4.18) and (4.20)–(4.24) of ℓp. To complete the proof it suffices to modify the extension x^*_{j1} for $j\in\mathcal{J}$ in order to ensure the equality in (4.70) to satisfy (4.19), and to keep other constraints (4.15)–(4.18) and (4.20)–(4.24) of ℓp satisfied.

If $\sum\limits_j x^*_{j1}<\lfloor r\rfloor$, then take a $j\in B_1$ with a positive $d_j=\min\{\lceil x_{j1}\rceil-x^*_{j1},\lfloor r\rfloor-x^*_{j1}-x_{j2}\}$. Recall that by Theorem 4.7, x_{j2} is integral for each $j\in B_1$. Such j exists. To prove this existence define $X=\{j\in B_1:\lceil x_{j1}\rceil=x^*_{j1}\}$ and $Y=\{j\in B_1:x^*_{j1}=\lfloor x_{j1}\rfloor\}$. By definition (4.67) and (4.46) we have $B_1=X\cup Y$, and since

$$\sum_j x^*_{j1}<\lfloor r\rfloor<\sum_j\lceil x_{j1}\rceil \tag{4.71}$$

we have $Y\ne\emptyset$. Suppose for a contradiction that for each job $j\in Y$ we have $\lfloor r\rfloor=x^*_{j1}+x_{j2}$. Thus we have

$$\sum_j x^*_{j1}=\sum_{j\in\mathcal{J}\setminus B_1}x_{j1}+\sum_{j\in X}\lceil x_{j1}\rceil+\sum_{j\in Y}\lfloor x_{j1}\rfloor<\lfloor r\rfloor. \tag{4.72}$$

Since for each job $j\in Y$ we have $\lfloor r\rfloor=\lfloor x_{j1}\rfloor+x_{j2}$, we obtain

$$\sum_{j\in\mathcal{J}\setminus B_1}x_{j1}+\sum_{j\in X}\lceil x_{j1}\rceil+|Y|\lfloor r\rfloor-\sum_{j\in Y}x_{j2}<\lfloor r\rfloor,$$

and by (4.71) the set Y is not empty. Since $\sum_{j\in Y} x_{j2} \leq \lfloor r \rfloor$ by (4.20) we get

$$\sum_{j\in\mathcal{J}\setminus B_1} x_{j1} + \sum_{j\in X} \lceil x_{j1} \rceil + |Y| \lfloor r \rfloor < 2\lfloor r \rfloor,$$

and thus $|Y| \leq 1$, and since Y is not empty we have $|Y| = 1$. However

$$\lfloor r \rfloor = \lfloor \sum_j x_{j1} \rfloor = \sum_j \lfloor x_{j1} \rfloor + \lfloor \sum_{j\in B_1} \varepsilon_j \rfloor,$$

where

$$\lfloor \sum_{j\in B_1} \varepsilon_j \rfloor \leq |B_1| - 1.$$

Thus

$$\lfloor r \rfloor = \lfloor \sum_j x_{j1} \rfloor \leq \sum_{j\in\mathcal{J}\setminus B_1} x_{j1} + \sum_{j\in B_1} \lfloor x_{j1} \rfloor + |B_1| - 1 = \sum_{j\in\mathcal{J}\setminus B_1} x_{j1} + \sum_{j\in X} \lceil x_{j1} \rceil + \sum_{j\in Y} \lfloor x_{j1} \rfloor$$

since $|Y| = 1$ which contradicts (4.72) and proves that $j \in Y$ with $d_j = 1$ exists. Set $d := \min\{\min_{j,d_j>0}\{d_j\}, \lfloor r \rfloor - \sum_j x^*_{j1}\} = 1$. Then, set $x^*_{j1} := x^*_{j1} + 1$ for some $j \in Y$ with $d_j = 1$. We have $x^*_{j1} \leq \min\{\lceil x_{j1} \rceil, \lfloor r \rfloor - x_{j2}\}$ and $\sum_j x^*_{j1} \leq \lfloor r \rfloor$ for the new extended solution, which ensures that all constraints (4.15)–(4.18) and (4.20)–(4.24) of ℓp are met in the new extended solution. Since $d = 1$ the $\sum_j x^*_{j1}$ gets closer to but does not exceed $\lfloor r \rfloor$. Therefore by (4.71) we finally reach an extended solution **t**, **z**, and x^*_{j2}, x^*_{j1} for $j \in \mathcal{J}$ that meets all (4.15)–(4.24) of ℓp. The solution is integral with $w' = \lfloor w \rfloor$, and $r' = \lfloor r \rfloor$ which proves the lemma. □

4.7 The Projection

Consider the following system S that defines the set of feasible solutions to the LP-relaxation of $\mathcal{ILP}$,

$$\sum_j b_{jh} - (\Delta(\mathcal{G}_2) - r) \leq \sum_j y_{jh} \leq w \qquad h \in \mathcal{G}_1 \tag{4.73}$$

$$\sum_j b_{jh} - (\Delta(\mathcal{G}_1) - r) \le \sum_j y_{jh} \le w \qquad h \in \mathcal{G}_2 \tag{4.74}$$

$$\sum_h y_{jh} \le w \qquad j \in \mathcal{J} \tag{4.75}$$

$$0 \le y_{jh} \le b_{jh} \qquad h \in \mathcal{M} \qquad j \in \mathcal{J} \tag{4.76}$$

$$\sum_j x_{j1} = r \tag{4.77}$$

$$\sum_j x_{j2} = r \tag{4.78}$$

$$x_{j1} + x_{j2} \le r \qquad j \in \mathcal{J} \tag{4.79}$$

$$0 \le x_{j\ell} \le a_{j\ell} \qquad j \in \mathcal{J} \qquad \ell = 1, 2 \tag{4.80}$$

$$\sum_{h \in \mathcal{G}_1} (b_{jh} - y_{jh}) + a_{j2} - x_{j2} \le \Delta(\mathcal{G}_2) - r \qquad j \in \mathcal{J} \tag{4.81}$$

$$\sum_{h \in \mathcal{G}_2} (b_{jh} - y_{jh}) + a_{j1} - x_{j1} \le \Delta(\mathcal{G}_1) - r \qquad j \in \mathcal{J}. \tag{4.82}$$

Now consider the system S_r obtained from S by dropping (4.77) and (4.78) and adding the constraints (4.91), (4.92), and (4.93). We use $\alpha_{j1} = \sum_{h \in \mathcal{G}_2} (b_{jh} - y_{jh}) + a_{j1} - \Delta(\mathcal{G}_1)$ and $\alpha_{j2} = \sum_{h \in \mathcal{G}_1} (b_{jh} - y_{jh}) + a_{j2} - \Delta(\mathcal{G}_2)$ for $j \in \mathcal{J}$ for convenience.

$$\sum_j b_{jh} - (\Delta(\mathcal{G}_2) - r) \le \sum_j y_{jh} \le w \qquad h \in \mathcal{G}_1 \tag{4.83}$$

$$\sum_j b_{jh} - (\Delta(\mathcal{G}_1) - r) \le \sum_j y_{jh} \le w \qquad h \in \mathcal{G}_2 \tag{4.84}$$

$$\sum_h y_{jh} \le w \qquad j \in \mathcal{J} \tag{4.85}$$

$$0 \le y_{jh} \le b_{jh} \qquad h \in \mathcal{M} \qquad j \in \mathcal{J} \tag{4.86}$$

$$x_{j1} + x_{j2} \le r \qquad j \in \mathcal{J} \tag{4.87}$$

$$0 \le x_{j\ell} \le a_{j\ell} \qquad j \in \mathcal{J} \qquad \ell = 1, 2 \tag{4.88}$$

$$\sum_{h \in \mathcal{G}_1} (b_{jh} - y_{jh}) + a_{j2} - x_{j2} \leq \Delta(\mathcal{G}_2) - r \qquad j \in \mathcal{J} \tag{4.89}$$

$$\sum_{h \in \mathcal{G}_2} (b_{jh} - y_{jh}) + a_{j1} - x_{j1} \leq \Delta(\mathcal{G}_1) - r \qquad j \in \mathcal{J} \tag{4.90}$$

$$\sum_{j} \alpha_{j1} + (n-1)r \leq 0 \tag{4.91}$$

$$\sum_{j} \alpha_{j2} + (n-1)r \leq 0 \tag{4.92}$$

$$0 \leq r \leq \min\{\Delta(\mathcal{G}_1), \Delta(\mathcal{G}_2)\}. \tag{4.93}$$

Finally consider the following projection on $\mathbf{y}, w, r$.

Lemma 4.13 *Let $\mathcal{P}$ be the polyhedron that consists of feasible solutions to S_r. Then the projection of $\mathcal{P}$ on $\mathbf{y}, w, r$, denoted by $\mathcal{Q}$, is the set of solutions to the following system of inequalities Q:*

$$\sum_{j} b_{jh} - (\Delta(\mathcal{G}_2) - r) \leq \sum_{j} y_{jh} \leq w \qquad h \in \mathcal{G}_1 \tag{4.94}$$

$$\sum_{j} b_{jh} - (\Delta(\mathcal{G}_1) - r) \leq \sum_{j} y_{jh} \leq w \qquad h \in \mathcal{G}_2 \tag{4.95}$$

$$\sum_{h} y_{jh} \leq w \qquad j \in \mathcal{J} \tag{4.96}$$

$$0 \leq y_{jh} \leq b_{jh} \qquad h \in \mathcal{M} \qquad j \in \mathcal{J} \tag{4.97}$$

$$\sum_{h \in \mathcal{G}_2} (b_{jh} - y_{jh}) + a_{j1} - \Delta(\mathcal{G}_1) \leq 0 \qquad j \in \mathcal{J} \tag{4.98}$$

$$\sum_{h \in \mathcal{G}_1} (b_{jh} - y_{jh}) + a_{j2} - \Delta(\mathcal{G}_2) \leq 0 \qquad j \in \mathcal{J} \tag{4.99}$$

$$\sum_{h \in \mathcal{G}_2} (b_{jh} - y_{jh}) + r - \Delta(\mathcal{G}_1) \leq 0 \qquad j \in \mathcal{J} \tag{4.100}$$

$$\sum_{h \in \mathcal{G}_1} (b_{jh} - y_{jh}) + r - \Delta(\mathcal{G}_2) \leq 0 \qquad j \in \mathcal{J} \tag{4.101}$$

$$\sum_h (b_{jh} - y_{jh}) + a_{j1} + a_{j2} - \Delta(\mathcal{G}_1) - \Delta(\mathcal{G}_2) + r \leq 0 \qquad j \in \mathcal{J} \tag{4.102}$$

$$\sum_j \alpha_{j1} + (n-1)r \leq 0 \tag{4.103}$$

$$\sum_j \alpha_{j2} + (n-1)r \leq 0 \tag{4.104}$$

$$0 \leq r \leq \min\{\Delta(\mathcal{G}_1), \Delta(\mathcal{G}_2)\}. \tag{4.105}$$

Proof The lemma follows by the Fourier-Motzkin elimination, see Schrijver [18], of variables $x_{j\ell}$ from the system S_r. □

We summarize the results of this section in the following theorem and lemma.

Theorem 4.11 *Let* $(\mathbf{y}, r, w)$ *be feasible for* Q. *There exists* $\mathbf{x}$ *such that the solution* $(\mathbf{y}, \mathbf{x}, w, r)$ *is feasible for* S.

Proof Let $s = (\mathbf{y}, r, w)$ be a feasible solution for Q. By Lemma 4.13 there exist $\mathbf{x} = (x_{j\ell})$, where $j \in \mathcal{J}$ and $\ell = 1, 2$, such that $s = (\mathbf{y}, \mathbf{x}, w, r)$ is feasible for S_r. Let X be the set of all such $\mathbf{x}$. Take $\mathbf{x} \in X$ with minimum distance $d = |r - \sum_j x_{j1}| + |r - \sum_j x_{j2}|$. We show that $d = 0$ for $\mathbf{x}$. Suppose that $r < \sum_j x_{j1}$ or $r < \sum_j x_{j2}$. Let $r < \sum_j x_{j1}$. If there is k such that $\alpha_{k1} + r < x_{k1}$, then set $x_{k1} := x_{k1} - \lambda$ where $\lambda = \min\{x_{k1} - (\alpha_k + r), \sum_j x_{j1} - r\}$. The new solution is in X and reduces d which gives a contradiction. Thus we have $\alpha_{j1} + r = x_{j1}$ for each j. Therefore $\sum_j \alpha_{j1} + nr = \sum_j x_{j1} \leq r$ by the constraint (4.103) which contradicts this case assumption. The proof for $r < \sum_j x_{j2}$ is similar. Therefore we have $r \geq \sum_j x_{j1}$ and $r \geq \sum_j x_{j2}$ for the $\mathbf{x}$. Suppose that $r > \sum_j x_{j1}$ or $r > \sum_j x_{j2}$. If there is k such that $x_{k1} + x_{k2} < r$ and ($x_{k1} < a_{k1}$ or $x_{k2} < a_{k2}$), then set $x_{k1} + \lambda$, where $\lambda = \min\{r - (x_{k1} + x_{k2}), a_{k1} - x_{k1}, d\}$ provided $x_{k1} < a_{k1}$. Otherwise, if $x_{k1} = a_{k1}$ and $x_{k2} < a_{k2}$, set $x_{k2} + \lambda$, where $\lambda = \min\{r - (x_{k1} + x_{k2}), a_{k2} - x_{k2}, d\}$. The new solution is in X but has smaller d which gives a contradiction. Thus we have $x_{j1} + x_{j2} = r$ or ($x_{j1} = a_{j1}$ and $x_{j2} = a_{j2}$) for each j. We have at least one j with $x_{j1} + x_{j2} = r$. Otherwise, $r > \min\{\Delta(\mathcal{G}_1), \Delta(\mathcal{G}_2)\}$ which contradicts (4.105). On the other hand, we can have at most one j with $x_{j1} + x_{j2} = r$. Otherwise $\sum_j (x_{j1} + x_{j2}) \geq 2r$ and since $r \geq \sum_j x_{j1}$ and $r \geq \sum_j x_{j2}$ for the $\mathbf{x}$ we get $r = \sum_j x_{j1}$ and $r = \sum_j x_{j2}$ which contradicts the assumption. Therefore there is exactly one j such that $x_{j1} + x_{j2} = r$, and $x_{k1} = a_{k1}$, and $x_{k2} = a_{k2}$ for $k \in \mathcal{J} \setminus \{j\}$. Hence $\Delta(\mathcal{G}_1) - a_{j1} + x_{j1} < r$ or $\Delta(\mathcal{G}_2) - a_{j2} + x_{j2} < r$. Since $\Delta(\mathcal{G}_1) - a_{j1} + x_{j1} \leq r$ and $\Delta(\mathcal{G}_2) - a_{j2} + x_{j2} \leq r$, we have $\Delta(\mathcal{G}_1) + \Delta(\mathcal{G}_2) - a_{j2} + x_{j2} - a_{j1} + x_{j1} < 2r$. Hence $\Delta(\mathcal{G}_1) + \Delta(\mathcal{G}_2) - a_{j2} - a_{j1} < r$ since $x_{j1} + x_{j2} = r$. However by (4.102) and (4.97) we have $a_{j1} + a_{j2} + r \leq \Delta(\mathcal{G}_1) + \Delta(\mathcal{G}_2)$ which gives a contradiction. Thus we have $d = 0$ and the solution is feasible for S. □

We have the following lemma.

Lemma 4.14 *If* $(\mathbf{y}, \mathbf{x}, r, w)$ *is feasible for* S*, then* $(\mathbf{y}, r, w)$ *is feasible for* Q.

Proof If $(\mathbf{y}, \mathbf{x}, r, w)$ is feasible for S, then it is also feasible for S_r. Observe that (4.77), (4.78), and (4.80) in S imply (4.93) in S_r, the (4.77) and (4.82) in S imply (4.91) in S_r, and the (4.78) and (4.81) in S imply (4.92) in S_r. Finally, by Lemma 4.13 the $(\mathbf{y}, r, w)$ is feasible for Q. □

The system Q is a network-flow model with lower and upper bounds on the arcs for fixed w and r.

4.8 Integral Optimal Solution to ℓp for $\sum_{j \in B_i} \varepsilon_j > \epsilon$, $i = 1, 2$

Consider $\mathbf{s}$ with $\sum_{j \in B_i} \varepsilon_j > \epsilon$ for $i = 1, 2$. By Lemma 4.9 overlap of B_1 and of B_2 do not occur simultaneously. Without loss of generality let us assume no overlap of B_2.

Consider the set Y_j of all columns of type $\binom{*}{*,j}$ in $d(\mathbf{y}, w)$ for $j \in B_2$. By Lemma 4.7, $l(Y_j) = \beta_j = \lfloor \beta_j \rfloor + \varepsilon_j$. Take an interval $Y \subseteq \bigcup_{j \in B_2} Y_j$ such that $l(Y) = \epsilon$. Such Y exists since there is no overlap of B_2 and $\sum_{j \in B_2} \varepsilon_j > \epsilon$. Let Y_{jh} be the set of columns $I \in Y$ such that $(j, h) \in M_I$, set $\gamma_{jh} := l(Y_{jh})$. Informally, γ_{jh} is the amount of $j \in \mathcal{J}$ done on $h \in \mathcal{M}$ in the interval Y. We define a truncated solution as follows $z^*_{jh} := y_{jh} - \gamma_{jh}$ for $h \in \mathcal{G}_1$, and $t^*_{jh} := y_{jh} - \gamma_{jh}$ for $h \in \mathcal{G}_2$, and $\lfloor r \rfloor$, $\lfloor w \rfloor$. Thus

$$\sum_{h \in \mathcal{G}_1} \gamma_{jh} + \sum_{h \in \mathcal{G}_2} \gamma_{jh} \le \epsilon \qquad j \in \mathcal{J}.$$

Theorem 4.12 *For* $\sum_{j \in B_i} \varepsilon_j > \epsilon$, $i = 1, 2$*, there is a feasible integral solution to* ℓp *with* $lp = \lfloor r \rfloor < r$.

Proof We begin by proving that the truncated solution $(y^* = (z^*, t^*), \lfloor r \rfloor, \lfloor w \rfloor)$ is feasible for Q.

The constraints (4.98) and (4.99): For $\mathbf{s}$ we have

$$\sum_{h \in \mathcal{G}_1} b_{jh} + a_{j2} - \Delta(\mathcal{G}_2) + r - x_{j2} \le \sum_{h \in \mathcal{G}_1} y_{jh} \qquad j \in \mathcal{J}$$

$$\sum_{h \in \mathcal{G}_2} b_{jh} + a_{j1} - \Delta(\mathcal{G}_1) + r - x_{j1} \le \sum_{h \in \mathcal{G}_2} y_{jh} \qquad j \in \mathcal{J}.$$

If $r - x_{j1} \ge \epsilon$ and $r - x_{j2} \ge \epsilon$ for each $j \in \mathcal{J}$, then $\sum_{h \in \mathcal{G}_1} y_{jh} - (r - x_{j2}) \le \sum_{h \in \mathcal{G}_1} z^*_{jh}$ and $\sum_{h \in \mathcal{G}_2} y_{jh} - (r - x_{j1}) \le \sum_{h \in \mathcal{G}_2} t^*_{jh}$ for each j. Hence (4.98) and (4.99) hold for the truncated solution. Otherwise, if $r - x_{j1} < \epsilon$ or $r - x_{j2} < \epsilon$ for some $j \in \mathcal{J}$, then $\lfloor r\rfloor \le x_{j1}$ or $\lfloor r\rfloor \le x_{j2}$ for some j. This implies $\sum_{j\in B_1}\varepsilon_j = \epsilon$ or $\sum_{j\in B_2}\varepsilon_j = \epsilon$ which contradicts the theorem's assumption.

The constraints (4.100) and (4.101): For **s** we have

$$\sum_{h\in\mathcal{G}_2} b_{jh} + r - \Delta(\mathcal{G}_1) + a_{j1} - x_{j1} \le \sum_{h\in\mathcal{G}_2} y_{jh} \qquad j \in \mathcal{J},$$

and

$$\sum_{h\in\mathcal{G}_1} b_{jh} + r - \Delta(\mathcal{G}_2) + a_{j2} - x_{j2} \le \sum_{h\in\mathcal{G}_1} y_{jh} \qquad j \in \mathcal{J}.$$

By constraint (4.22) and definition of the truncated solution

$$\sum_{h\in\mathcal{G}_2} b_{jh} + \lfloor r\rfloor - \Delta(\mathcal{G}_1 \le \sum_{h\in\mathcal{G}_2} y_{jh} - \epsilon \le \sum_{h\in\mathcal{G}_2} t^*_{jh} \qquad j \in \mathcal{J},$$

and

$$\sum_{h\in\mathcal{G}_1} b_{jh} + \lfloor r\rfloor - \Delta(\mathcal{G}_2 \le \sum_{h\in\mathcal{G}_1} y_{jh} - \epsilon \le \sum_{h\in\mathcal{G}_1} z^*_{jh} \qquad j \in \mathcal{J}.$$

Hence (4.100) and (4.101) hold.

The constraints (4.102): For **s** by (4.23) and (4.24) we have

$$\sum_{h\in\mathcal{G}_1} (b_{jh} - y_{jh}) + a_{j2} - x_{j2} \le \Delta(\mathcal{G}_2) - r$$

and

$$\sum_{h\in\mathcal{G}_2} (b_{jh} - y_{jh}) + a_{j1} - x_{j1} \le \Delta(\mathcal{G}_1) - r,$$

by summing up the two side by side we get

$$\sum_h (b_{jh} - y_{jh}) + a_{j1} + a_{j2} - x_{j1} - x_{j2} \le \Delta(\mathcal{G}_1) + \Delta(\mathcal{G}_2) - 2r$$

or

$$\sum_h b_{jh} + a_{j1} + a_{j2} - \Delta(\mathcal{G}_1) - \Delta(\mathcal{G}_2) + \lfloor r\rfloor \le \sum_h y_{jh} - r + x_{j1} + x_{j2} - \epsilon.$$

Since $\sum_h y_{jh} - \epsilon \le \sum_{h\in\mathcal{G}_1} z^*_{jh} + \sum_{h\in\mathcal{G}_2} t^*_{jh}$, we have

$$\sum_h y_{jh} - r + x_{j1} + x_{j2} - \epsilon \leq \sum_{h \in \mathcal{G}_1} z^*_{jh} + \sum_{h \in \mathcal{G}_2} t^*_{jh} - r + x_{j1} + x_{j2}.$$

But $-r + x_{j1} + x_{j2} \leq 0$ by the constraint (4.21) and thus we get

$$\sum_h b_{jh} + a_{j1} + a_{j2} - \Delta(\mathcal{G}_1) - \Delta(\mathcal{G}_2) + \lfloor r \rfloor \leq \sum_{h \in \mathcal{G}_1} z^*_{jh} + \sum_{h \in \mathcal{G}_2} t^*_{jh}$$

which proves that (4.102) holds for $y^* = (z^*, t^*)$.

The constraints (4.94)–(4.95): For **s** by (4.15), and (4.16) we have

$$\sum_j b_{jh} - \Delta(\mathcal{G}_2) + \lfloor r \rfloor \leq \sum_j y_{jh} - \epsilon \leq \lfloor w \rfloor \qquad h \in \mathcal{G}_1,$$

and

$$\sum_j b_{jh} - \Delta(\mathcal{G}_1) + \lfloor r \rfloor \leq \sum_j y_{jh} - \epsilon \leq \lfloor w \rfloor \qquad h \in \mathcal{G}_2.$$

For the truncated solution we have $\sum_j y_{jh} = \sum_j z^*_{jh} + \sum_j \gamma_{jh}$ for $h \in \mathcal{G}_1$, and $\sum_j y_{jh} = \sum_j t^*_{jh} + \sum_j \gamma_{jh}$ for $h \in \mathcal{G}_2$. Because of the machine saturation $\sum_j \gamma_{jh} = \epsilon$ for $h \in \mathcal{G}_1 \cup \mathcal{G}_2$. Thus

$$\sum_j b_{jh} - \Delta(\mathcal{G}_2) + \lfloor r \rfloor \leq \sum_j z^*_{jh} \leq \lfloor w \rfloor \qquad h \in \mathcal{G}_1,$$

$$\sum_j b_{jh} - \Delta(\mathcal{G}_1) + \lfloor r \rfloor \leq \sum_j t^*_{jh} \leq \lfloor w \rfloor \qquad h \in \mathcal{G}_2,$$

and (4.94) and (4.95) are satisfied by the truncated solution. By the machine saturation we have $\sum_j z^*_{jh} = \lfloor w \rfloor$ for $h \in \mathcal{G}_1$, and $\sum_j t^*_{jh} = \lfloor w \rfloor$ for $h \in \mathcal{G}_2$.

The constraint (4.96): For **s** by (4.17) we have $l(X_j) \leq l(d(\mathbf{y}, w)) = w$ where X_j is the set of all columns in $d(\mathbf{y}, w)$ that match $j \in \mathcal{J}$. Since $l(Y) = \epsilon$, we get $l(Z_j) \leq l(d(\mathbf{y}, w) \setminus Y) = \lfloor w \rfloor$ where Z_j is the set of all columns in $d(\mathbf{y}, w) \setminus Y$ that match $j \in \mathcal{J}$. We have $l(X_j) = l((X_j \cap Y) \cup (X_j \setminus Y)) = l(X_j \cap Y) + l(X_j \setminus Y) = l(X_j \cap Y) + l(Z_j)$. Hence $l(Z_j) = l(X_j) - l(X_j \cap Y) = \sum_h y_{jh} - \sum_h \gamma_{jh} = \sum_h y^*_{jh}$. Thus $\sum_h y^*_{jh} \leq \lfloor w \rfloor$ and (4.96) is satisfied by the truncated solution.

Finally, the constraints (4.103) and (4.104). First we observe that $|\mathcal{G}_1| \leq n - 1$ and $|\mathcal{G}_2| \leq n - 1$. Otherwise $|\mathcal{G}_1| > n - 1$ or $|\mathcal{G}_2| > n - 1$ and since by the saturation $|\mathcal{G}_1| + |\mathcal{G}_2| \leq n$ we would have $|\mathcal{G}_1| = 0$ or $|\mathcal{G}_2| = 0$ which contradicts the assumption of non-empty groups. Second, by summing up (4.23) side by side for **s** over all jobs and doing the same for (4.24) we get

$$\sum_{h \in \mathcal{G}_2} \sum_j b_{jh} - |\mathcal{G}_2| w + (1-n)\Delta(\mathcal{G}_1) + (n-1) r \le 0$$

and

$$\sum_{h \in \mathcal{G}_1} \sum_j b_{jh} - |\mathcal{G}_1| w + (1-n)\Delta(\mathcal{G}_2) + (n-1) r \le 0,$$

respectively. Since $|\mathcal{G}_1| \le n-1$ and $|\mathcal{G}_2| \le n-1$, we get

$$\sum_{h \in \mathcal{G}_2} \sum_j b_{jh} - |\mathcal{G}_2| \lfloor w \rfloor + (1-n)\Delta(\mathcal{G}_1) + (n-1) \lfloor r \rfloor \le 0$$

and

$$\sum_{h \in \mathcal{G}_1} \sum_j b_{jh} - |\mathcal{G}_1| \lfloor w \rfloor + (1-n)\Delta(\mathcal{G}_2) + (n-1) \lfloor r \rfloor \le 0.$$

By the machine saturation we have $|\mathcal{G}_2| \lfloor w \rfloor = \sum_{h \in \mathcal{G}_2} \sum_j t^*_{jh}$ and $|\mathcal{G}_1| \lfloor w \rfloor = \sum_{h \in \mathcal{G}_1} \sum_j z^*_{jh}$ which proves that (4.103) and (4.104) are satisfied by the truncated solution.

Therefore the truncated solution $(y^* = (z^*, t^*), \lfloor r \rfloor, \lfloor w \rfloor)$ is feasible for Q, and by Theorem 4.11 there exists $\mathbf{x}^*$ such that $(y^* = (z^*, t^*), \mathbf{x}^*, \lfloor r \rfloor, \lfloor w \rfloor)$ is feasible for S. Moreover $\lfloor r^* \rfloor \le \lfloor r \rfloor$, and $\lfloor w \rfloor - \lfloor r \rfloor = \lceil w^* - r^* \rceil$ since $\mathbf{s} = (\mathbf{y}, \mathbf{x}, r, w)$ is feasible for ℓp. Thus the solution $(y^* = (z^*, t^*), \mathbf{x}^*, \lfloor r \rfloor, \lfloor w \rfloor)$ is feasible for ℓp and $lp = \lfloor r \rfloor$. For a feasible solution to Q with integral $\lfloor w \rfloor$ and $\lfloor r \rfloor$ all lower and upper bounds in the network Q are integral thus we can find in polynomial time an integral $\mathbf{y}^*$. Finally for given integral and fixed $\lfloor r \rfloor$, $\lfloor w \rfloor$, and $\mathbf{y}^*$ the S becomes a network-flow model with integral lower and upper bounds on the flows. Thus we can find in polynomial time an integral $\mathbf{x}^*$ such that the integer solution $(\mathbf{y}^*, \mathbf{x}^*, \lfloor r \rfloor, \lfloor w \rfloor)$ is feasible for lp and $lp = \lfloor r \rfloor$. □

Figure 4.5 gives an integral solution to $\mathcal{ILP}$ for the instance in Fig. 4.4. The solution has part (b) of size $\lfloor r \rfloor = 1$ that consists of job J_1 on $\mathcal{G}_2$ and job J_6 on $\mathcal{G}_1$. This part (b) is shorter than the part (b) in $\mathbf{s}$ which is of size $r = \frac{3}{2}$, see Fig. 4.4, and thus $\mathbf{s}$ cannot be an optimal solution to ℓp.

4.9 The Proof of the Conjecture

We are now ready to prove Theorem 4.3 which proves the conjecture.

Proof For contradiction suppose the optimal value for ℓp is fractional, $lp = r = \lfloor r \rfloor + \epsilon$, where $\epsilon > 0$. By Theorem 4.10 there is a feasible integral solution to ℓp

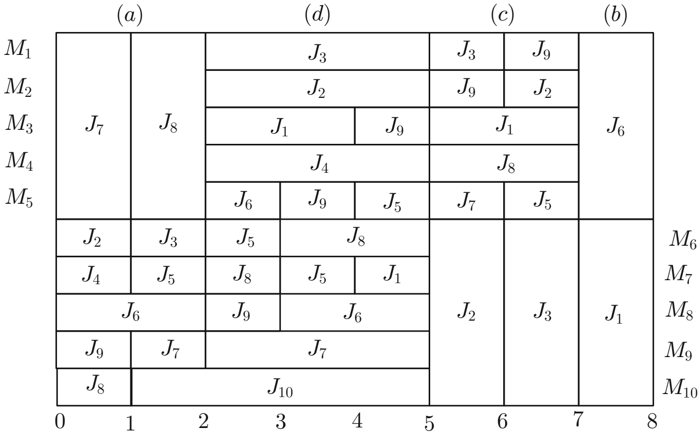


Fig. 4.5 An integral solution ($\mathbf{y}^*$, $\mathbf{x}^*$, $\lfloor r \rfloor = 1$, $\lfloor w \rfloor = 3$) for S in Fig. 4.4

with $lp = \lfloor r \rfloor$ for $\sum_{j \in B_1} \varepsilon_j = \epsilon$ or $\sum_{j \in B_2} \varepsilon_j = \epsilon$. By Theorem 4.12 there is a feasible integral solution to ℓp with $lp = \lfloor r \rfloor$ for $\sum_{j \in B_1} \varepsilon_j > \epsilon$ and $\sum_{j \in B_2} \varepsilon_j > \epsilon$. Thus there is a feasible integral solution for ℓp with $\lfloor r \rfloor < r$. Hence there is a feasible solution to ℓp which is smaller than optimal r which gives contradiction and proves the first part of the theorem. Thus optimal $\mathbf{s}$ has both r and w integer. The $\mathbf{s}$ is feasible for S and thus it is feasible for Q by Lemma 4.14. For a feasible solution to Q with integral w and r all lower and upper bounds in the network Q are integral thus we can find in polynomial time an integral $\mathbf{y}$. Finally for given integral and fixed r, w and $\mathbf{y}$ the S becomes a network with integral lower and upper bounds on the flows. Thus we can find in polynomial time an integral $\mathbf{x}$ such that the integer solution ($\mathbf{y}$, $\mathbf{x}$, r, w) is feasible for lp and $lp = r$. □

The question remains whether there is a simpler, perhaps more direct (not using LP- relaxations), approach that would result in the polynomial-time algorithm for two groups, also another natural question remains whether there is a shorter proof of the conjecture. These two remain challenging questions worthy further investigation.

4.10 Complexity of Open Shop Scheduling with Preemptions Allowed at Any Points

The idea of using a linear program to find a schedule that minimizes makespan for open shop with multiprocessor operations has been introduced in Sect. 4.3 for two groups, $p = 2$. This idea has been extended in Ittig [14] to any fixed $p > 2$. The extension is presented in this section. We begin with $p = 3$. Then any schedule S

Table 4.1 Possible intervals types in schedule $\mathcal{S}$; 0 and 1 in column $\mathcal{G}_\ell$, $\ell = 1, 2, 3$ denote individual and group operations on machines in $\mathcal{G}_\ell$, respectively

Interval type	Types of operations on machines in		Interval length	
	$\mathcal{G}_1$	$\mathcal{G}_2$	$\mathcal{G}_3$	
1	1	1	0	a_1
2	0	0	1	a_2
3	1	0	1	b_1
4	0	1	0	b_2
5	0	1	1	c_1
6	1	0	0	c_2
7	1	1	1	r
8	0	0	0	w

partitions the interval $[0, C_{\max}]$ into $2^p = 8$ disjoint interval types, some may be empty, listed in Table 4.1.

The interval of type (1) has group operations on both $\mathcal{G}_1$ and $\mathcal{G}_2$, thus 1 in the columns $\mathcal{G}_1$ and $\mathcal{G}_2$, and individual operations or idle time on $\mathcal{G}_3$, thus 0 in the column $\mathcal{G}_3$. The length of the interval of type (1) is denoted by a_1. Similarly the interval of type (2) has individual operations or idle time on both $\mathcal{G}_1$ and $\mathcal{G}_2$, thus 0 in the columns $\mathcal{G}_1$ and $\mathcal{G}_2$, and group operations on $\mathcal{G}_3$, thus 1 in the column $\mathcal{G}_3$. The length of the interval of type (2) is a_2. The other interval types should be clear from the table by now. Some of those interval types may be empty in $\mathcal{S}$, then their lengths equal 0. The interval types can be permuted in any possible way still giving the schedule with the same makespan as $\mathcal{S}$. In order to find the schedule that minimizes makespan we define variables as in Fig. 4.6, where the variables $x^i_{j\ell}$ and y^i_{jh}, for $J_j \in \mathcal{J}$, $\ell = 1, 2, 3$ and $M_h \in \mathcal{M}$, are introduced for pair $2i - 1$ and $2i$ of the interval types, $i = 1, 2, 3$. The two interval types in each pair complement one another; they partition the three groups into two disjoint sets. The variable $0 \le x^i_{j\ell}$ denotes the amount of job J_j group operation $\hat{O}_{j\ell}$ processed on $\mathcal{G}_\ell$ in the intervals of types $(2i - 1)$ and $(2i)$, $i = 1, 2, 3$, and the variable $0 \le y^i_{jh}$ denotes the amount of job J_j individual operation O_{jh} processed on M_h in the intervals of types $(2i-1)$ and $(2i)$, $i = 1, 2, 3$. The remaining amount $0 \le a_{j\ell} - (x^1_{j\ell} + x^2_{j\ell} + x^3_{j\ell})$ of job J_j group operation $\hat{O}_{j\ell}$ is left for the interval of type (7), and the remaining amount $0 \le b_{jh} - (y^1_{jh} + y^2_{jh} + y^3_{jh})$ of job J_j individual operation O_{jh} is left for the interval of type (8). The remaining non-negative variables a_1, a_2, b_1, b_2, c_1, c_2, r, and w denote the lengths of the intervals (1) – (8), respectively.

The constraints for each interval need to ensure that each job is processed in the interval for *not* longer than the length of the interval, and each machine is occupied for *not* longer than the length of the interval. Thus the constraints ensure that a feasible schedule can be obtained for each interval using the algorithms for $O|\text{pmtn}|C_{\max}$ discussed earlier in Sect. 3.7.1. For the interval type (1) of length a_1 we thus have the following constraints:

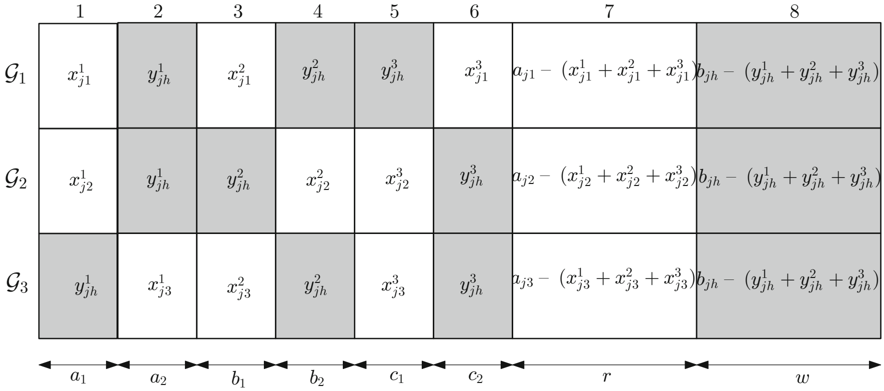


Fig. 4.6 The variables and interval types used in the linear program to minimize makespan

$$x_{j1}^1 + x_{j2}^1 + \sum_{h \in \mathcal{G}_3} y_{jh}^1 \leq a_1 \qquad j \in \mathcal{J}$$

for the jobs, and the following:

$$\sum_j x_{j1}^1 = a_1$$

$$\sum_j x_{j2}^1 = a_1$$

$$\sum_j y_{jh}^1 \leq a_1 \qquad h \in \mathcal{G}_3$$

for the machines. The constraints for the interval types (2)–(6) can be readily obtained in a similar fashion. The reader is encouraged to write them down, see Problem 4.2. For the interval type (7) we have

$$(a_{j1} + a_{j2} + a_{j3}) - \sum_{i=1}^{3} \sum_{\ell=1}^{3} x_{j\ell}^i \leq r \qquad j \in \mathcal{J}$$

for the jobs, and

$$\sum_j a_{j1} - \sum_{i=1}^{3} \sum_j x_{j\ell}^i = r \qquad \ell = 1, 2, 3$$

for the groups. Finally, for the interval type (8) we have

$$\sum_h b_{jh} - \sum_{i=1}^{3} \sum_h y^i_{jh} \le w \qquad j \in \mathcal{J}$$

for the jobs, and

$$\sum_j b_{jh} - \sum_{i=1}^{3} \sum_j y^i_{jh} \le w \qquad h \in \mathcal{M}$$

for the machines. The makespan equals $a_1 + a_2 + b_1 + b_2 + c_1 + c_2 + r + w$. However we have the following equalities:

$$\Delta(\mathcal{G}_1) = a_1 + b_1 + c_2 + r,$$

$$\Delta(\mathcal{G}_2) = a_1 + b_2 + c_1 + r,$$

$$\Delta(\mathcal{G}_3) = a_2 + b_1 + c_1 + r,$$

which can be used to reduce the number of variables in the linear program. By eliminating the variables a_2, b_2, and c_2 we obtain the following objective for the linear program:

$$\min(w - 2r - a_1 - b_1 - c_1).$$

Following the idea of interval types, linear programs can be obtained in polynomial time for any fixed number p of groups. All entries in the constraint matrix of those linear programs are 0, $+1$, or -1, thus the linear programs can be solved by a strongly polynomial algorithm. This proves that the makespan minimization for open shop scheduling problem with multiprocessor operations is polynomial for any fixed number of groups p. Observe that the number of interval types equals 2^p, and thus it is exponential when p is part of the input. Therefore problem complexity remains an open question for the case when p is part of the problem input. A polynomial-time algorithm, if any exists, that would produce optimal schedules needs to somehow limit the number of possible interval types so that the number can be bounded by a polynomial of the input size. The question whether such a bound exists remains open.

Problem 4.1 Is the problem of makespan minimization for preemptive scheduling of open shop with multiprocessor operations polynomial?

4.11 Integer Preemptions: Approximations

The solutions minimizing makespan for the open shop scheduling problem with multiprocessor operations and preemptions allowed at any points can be rounded in polynomial time to obtain optimal solutions with preemptions allowed at integer points only for $p = 2$. We presented this approach in Sects. 4.3–4.9 where we also proved that the minimum makespan for the latter problem equals $\lceil C_{\max} \rceil$, where $C_{\max}$ is the minimum makespan of the former. Though it may be tempting to think that the approach based on rounding in polynomial results in optimal solutions for other values of $p \geq 3$, this is unfortunately not the case. We showed in Sect. 4.2 that such rounding in polynomial time is impossible unless $NP = P$. However, the optimal solutions to the linear program for the problem with preemptions at any points can be rounded to provide *approximate* solutions to the problem with integral preemptions only for any fixed p. Ittig [14] has shown a polynomial-time rounding algorithm A that gives solutions within a constant absolute error for any fixed number of groups p.

Theorem 4.13 *Let C be the makespan of the optimal solution with preemptions at any points, and let C^A be the makespan of the solution with preemptions at integer points only obtained by the rounding algorithm A. We have*

$$C^A - C \leq 2p \cdot (2^{p-1} - 1) + 3.$$

Despite this constant absolute error obtained for the rounding algorithm we have the following implication of Theorem 4.1.

Theorem 4.14 *If $P \neq NP$, then no polynomial-time algorithm for University timetabling for $p \geq 3$ exists with the worst case ratio less than $\frac{4}{3}$.*

Proof Consider the set $\mathcal{I}$ of instances of University timetabling defined in the proof of Theorem 4.1. The problem Π defined by $\mathcal{I}$ and the question whether $I \in \mathcal{I}$ has a schedule with makespan not exceeding 3 or not is NP-complete which follows immediately from the proof of Theorem 4.1. Suppose for contradiction that there is a polynomial-time algorithm B such that $C^B_{\max}/C^*_{\max} < 4/3$ for any instance of University timetabling. Thus, in particular, $C^B_{\max}/C^*_{\max} < 4/3$ for any instance of Π. The algorithm B can be used to solve Π as follows. If $C^A_{\max} \leq 3$ for instance I, then the answer for I is affirmative. Otherwise, if $C^B_{\max} > 3$ for I, then, since all processing times in I are integer, we have $C^B_{\max} \geq 4$ and integer. Thus, since $C^*_{\max} > 3C^B_{\max}/4$, we get $C^*_{\max} > 3$ and the answer for I is negative. Since $C^B_{\max}$ can be computed in polynomial time for each $I \in \mathcal{I}$, we have Π in P. This implies $P = NP$ since Π is NP-complete and gives contradiction. □

These results indicate that the rounding algorithm A may give the ratios $\frac{C^A}{C^*}$, where C^* is the makespan of optimal schedule with preemptions at integer points only, close to 1 for the instances with large instance degree $\Delta \leq C^*$ and fixed p. However the worst case ratios are not smaller than $\frac{4}{3}$ for the instances with short schedules, thus small instance degree Δ, and arbitrary p. The inapproximability in Theorem 4.14 holds for open shops with 0-1 operations and no preemptions.

At the beginning of this chapter, we showed that the worst case ratio equals 2 for a simple decomposition algorithm. Asration and de Werra [1] give a polynomial-time algorithm with the worst case ratio $\frac{7}{6}$; however, their algorithm requires additional assumptions about the Δ's. Both approximations are for preemptive schedules with preemptions at integer points.

4.12 Other Models of Multiprocessor Operations

Brucker and Krämer [5] and Brucker [4] consider a different model of open shop with multiprocessor operations. Their model assumes the same subset of machines $\mathcal{M}_h$ for each operation $O_{i,h}$ regardless of the job J_i. The sets $\mathcal{M}_h$, $h = 1, \ldots, m$ may not be disjoint in which case they are called incompatible; disjoint sets are called compatible. They consider open shops with fixed m, which is called the number of stages. The stages form a compatibility graph with vertices corresponding to the stages and edges between the stages that are compatible. For unit-time operations they show that the open shop scheduling is polynomial for a number of objective functions including makespan, total weighted completion time, and weighted number of tardy jobs, see Brucker [4] for a complete list of results. The makespan minimization for three stages, $m = 3$, and arbitrary processing times reduces to either $O2||C_{\max}$ or to $O3||C_{\max}$ depending of the compatibility graph, see Brucker and Krämer [5].

Problems

4.1 Show that the open shop scheduling with multiprocessor operations is NP-hard in the strong sense for $p = 3$.

4.2 Write down complete linear program for $p = 3$, and for $p = 4$.

References

1. A.S. Asratian, D. de Werra, A generalized class-teacher model for some timtabling problems. Eur. J. Oper. Res. **143**, 531–542 (2002)
2. A.S. Asratian, R.R. Kamalian, Interval colorings of edges of a multigraph (in Russian). Appl Math (also arXiv:1401.8079v1) **5**, 25–34 (1987)
3. J.A. Bondy, U.S.R. Murty, *Graph Theory with Applications* (North-Holland Publishing, Amsterdam, 1976)
4. P. Brucker, *Scheduling Algorithms* (Springer, Berlin, 1995)

5. P. Brucker, A. Krämer, Schop scheduling with multiprocessor tasks on dedicated processors. Ann. Oper. Res. **57**, 13–27 (1995)
6. E. Cole, K. Ost, S. Schirra, Edge-coloring Bipartite Multigraphs in $O(E \log D)$. Combinatorica **21**, 5–12 (2001)
7. D. de Werra, An introduction to timetabling. Eur. J. Oper. Res. **19**, 151–162 (1985)
8. D. de Werra, A.S. Asratian, S. Durand, Complexity of some special types of timetabling problems. J. Sched. **5**, 171–183 (2002)
9. D. de Werra, T. Kis, W. Kubiak, Preemptive open shop scheduling with multiprocessors: polynomial cases and applications. J. Sched. **11**, 75–83 (2008)
10. S. Even, A. Itai, A. Shamir, On the complexity of timetable and multicomodity flow problems. SIAM J. Comput. **5**, 691–703 (1976)
11. H.N. Gabow, O. Kariv, Algorithms for edge coloring bipartite graphs and multigraphs. SIAM J. Comput. **11**, 117–129 (1982)
12. T. Gonzalez, S. Sahni, Open shop scheduling to minimize finish time. J. ACM **23**, 665–679 (1976)
13. C. Gotlieb, The construction of class-teacher time-tables, in *Proc. IFIP Congress 62, Munich* (North-Holland, Amsterdam, 1963)
14. O. Ittig, Scheduling with multiprocessors: a rounding network algorithm with a constant error. Master's thesis, Memorial University of Newfoundland and EPFL, 2005
15. T. Kis, D. de Werra, W. Kubiak, A projective algorithm for preemptive open shop scheduling with two multiprocessor groups. Oper. Res. Lett. **38**, 129–132 (2010)
16. W. Kubiak, On a conjecture for the university timetabling problem. Discrete Appl. Math. **299**, 26–49 (2021)
17. E.L. Lawler, *Combinatorial Optimization. Networks and Matroids* (Dover Publications, New York, 2001)
18. A. Schrijver, *Theory of Linear and Integer Programming* (Wiley, London, 1999)
19. É. Tardos, A strongly polynomial algorithm to solve combinatorial linear programs. Oper. Res. **34**, 250–256 (1986)