

Generalizing Horn's conditions for preemptive scheduling on identical parallel machines via network flow techniques

Akiyoshi Shioura¹ | Vitaly A. Strusevich² | Natalia V. Shakhlevich³

¹Department of Industrial Engineering and Economics, Tokyo Institute of Technology, Tokyo, Japan

²Sherwood Road, Welling, Kent, UK

³School of Computing, University of Leeds, Leeds, UK

Correspondence

Natalia V. Shakhlevich, University of Leeds, Leeds LS2 9JT, UK.

Email: N.Shakhlevich@leeds.ac.uk

Funding information

Engineering and Physical Sciences Research Council, Grant/Award Number: EP/T01461X/1

Abstract

We study the use of flow-based algorithmic and proof techniques applied to preemptive scheduling of jobs on parallel identical machines. For the classical problem in which the jobs have individual release dates and must be finished by a common deadline, we present and prove unified necessary and sufficient conditions for the existence of a feasible schedule by examining the structure of minimum cuts in a special network. We then study an enhanced model that allows the presence of additional resources, provided that some jobs at any time of their processing require one unit of a particular resource. We extend our argument developed for the classical case to this enhanced model. The generalized necessary and sufficient conditions for the existence of a feasible schedule are presented and proved using the max-flow/min-cut reasoning.

KEYWORDS

maximum-flow problem, minimum-cut problem, optimality conditions, parallel-machine scheduling, preemption, scheduling under resource constraints, scheduling with controllable processing times

1 | INTRODUCTION

Numerous modeling, algorithmic and proof techniques designed in machine scheduling are based on various network flow models and methods. According to Google Scholar, in scheduling literature the network flow based algorithms and reasoning appear in more than fifty thousand publications. One of the earliest examples of handling a scheduling problem by a network flow approach is the classical paper by Horn [10] written 50 years ago. Horn designs two sets of necessary and sufficient conditions for the existence of a feasible schedule for a basic scheduling problem on parallel identical machines. The purpose of this study is to demonstrate that the conditions developed by Horn, despite their differences, can be derived from the same principles that rely on the maximum-flow minimum-cut theorem. Moreover, we show how a similar proof technique can be extended to a more general scheduling model with resource constraints.

The classical machine scheduling models are traditionally formulated in terms of jobs to be processed on machines. In the main model that we consider the jobs have to be processed on identical parallel machines. Preemption is allowed, that is, the processing of any job can be interrupted and resumed later, possibly on a different machine. Problems of preemptive scheduling have numerous practical applications. One of the popular illustrations can be found in computing, when a number of files (jobs) have to be processed by programmers of similar skills (identical parallel machines). At a time, a file is only available to at most one programmer and each programmer can process at most one file. A programmer can interrupt the processing of a file which will be done later by another programmer.

This is an open access article under the terms of the [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2023 The Authors. *Networks* published by Wiley Periodicals LLC.

The classical paper by Horn [10] studies the feasibility problem in which the jobs have different release dates and must be completed by a common deadline. Horn presents necessary and sufficient conditions for the existing a feasible preemptive schedule. These conditions can be seen as one of the corner stones of scheduling theory, and they are present in essentially any scheduling book. One of these conditions is formulated as a set of inequalities, while the other is related to the maximum flow in a special network G_0 . The conditions as described in [10] show no apparent link between them. We present a unified condition which combines both previously known conditions, linking them in terms of the cuts in network G_0 .

The proof technique that we develop here appears to be useful for extending Horn's conditions to a more general problem that requires the use of additional resources. Scheduling under Resource Constraints is an established research area within machine scheduling. There are multiple publications in this area which have been reviewed in a number of surveys, from the earliest [2] till the most recent [3]. It has been introduced to handle various practical situations that cannot be adequately covered by the classical models; some of such examples are given in Section 5.1. The scheduling problem with resource constraints that we consider in this paper can be illustrated by extending the example given above. Suppose that some of the files for their processing require a special software (an additional resource). Programmers have that software installed on their computers. However, the software provider insists that the software can be run on one computer only at a time. Notice that similar licenses are applied to many software products, for example, to a popular video editor.

The remainder of this article is organized as follows. We start with the feasibility problem with no resource constraints, provided that jobs have different release dates and a common deadline. Section 2 contains some introductory comments, definitions and notation. Section 3 reproduces two forms of the classical Horn's conditions that are necessary and sufficient for the existence of a feasible schedule. The unified conditions are formulated in Section 4. The proof shows the equivalence of four statements, two of which are the classical Horn's conditions and their equivalence is due to their link to the minimum cuts in a special network.

We then turn to the extended model, in which the jobs require additional resources. Section 5.1 provides some preliminary information on resource constrained scheduling. The model with no release dates is considered in Section 5.2, where a polynomial-time algorithm is provided. For the problem with different release dates, the necessary and sufficient conditions for the existence of a feasible schedule, including the unified condition, are formulated and proved in Section 5.3. Conclusions are discussed in Section 6.

2 | PRELIMINARIES

In this section, we formally define the basic scheduling models under study and briefly review the results on computational complexity.

In the basic problem, the jobs J_1, J_2, \dots, J_n have to be processed with preemptions on m identical parallel machines M_1, \dots, M_m . Denote $N = \{1, 2, \dots, n\}$. The processing time of job $J_j, j \in N$, is equal to $p(j)$. For job J_j , its release date is given and is denoted by $r(j)$. The completion time of job J_j in a feasible schedule is denoted by $C(j)$. We consider the problems, in which it is required to verify whether there exists a preemptive schedule such that all jobs are completed by a given common deadline D . More formally, in a feasible schedule S_0 a job J_j is not allowed to be processed outside the time interval $[r(j), D]$.

In order to refer to a scheduling problem in a clear and compact way, the three-field classification scheme of the form $\alpha | \beta | \gamma$ is widely accepted, where α describes the machine environment, β presents the processing conditions and γ is the objective function. The main focus of this paper is on scheduling models with identical parallel machines, in which case in the field α we write "P" if the number of machines is variable (part of the input). If the number of machines is fixed and equal to m , then we write "Pm" rather than "P". In the review parts of this paper we may refer to more general models, with uniform (or uniformly related) machines that differ in their speeds and with unrelated machines. For these machine environments, in the field α we write "Q" and "R", respectively.

The classical problem with no resource constraints of finding a feasible nonpreemptive schedule on m parallel identical machines in which all jobs are completed by a common deadline D can be denoted either by $P|C(j) \leq D|$ — if all jobs are released at time zero or by $P|r(j), C(j) \leq D|$ — if the jobs have individual release dates $r(j)$. A similar problem with preemption allowed is denoted by $P|pmtn, C(j) \leq D|$ — and by $Pm|r(j), pmtn, C(j) \leq D|$ —, respectively.

In each of these problems, it is required to determine whether a required feasible schedule exists and, if it does, to find it. These decision problems are closely related to their optimization counterparts, in which the goal is to find a schedule that minimizes the makespan $C_{\max}^* = \max \{C(j) | j \in N\}$. In the corresponding three-field notation, the third field γ becomes C_{\max} , while the string " $C(j) \leq D$ " is removed from the middle field β ; for instance, the problem of finding a makespan-optimal preemptive schedule for the jobs with release dates is denoted by $P|r(j), pmtn|C_{\max}$. It is clear that solving an optimization problem can be reduced to a sequence of the corresponding decision problems. On the other hand, if an optimal makespan C_{\max}^* is found, then the corresponding decision problem is feasible for all $D \geq C_{\max}^*$ and infeasible for any smaller deadline.

We briefly review the results on the computational complexity of the problems with no resource constraints; more details and appropriate references can be found in a focused survey [4]. If preemption is not allowed, then already problem $P2|C_{\max}$, the simplest of the outlined range, is NP-hard. Moreover, problem $P|C_{\max}$ is NP-hard in the strong sense. If preemption is allowed, then problem $P|pmtn|C_{\max}$ with no release dates is solvable in $O(n)$ time as proved by McNaughton [18]. This algorithm will be part of several algorithms presented in this paper, therefore, we outline it below.

If all jobs arrive at time zero, for any schedule S_0 that is feasible for problem $P|pmtn|C_{\max}$ the makespan $C_{\max}(S_0)$ cannot be smaller than the processing time of an individual job. Additionally, in any feasible schedule there must be a machine with a total load of at least $p(N)/m$, which is the average machine load. Thus,

$$C_{\max}(S_0) \geq T_0,$$

where

$$T_0 = \max \left\{ \max \{p(j) \mid j \in N\}, \frac{1}{m}p(N) \right\}.$$

For problem $P|pmtn|C_{\max}$, McNaughton's algorithm delivers an optimal schedule with the makespan equal to T_0 . It is convenient to represent the algorithm as the following procedure. Consider an artificial schedule S_A in which the jobs are placed on an auxiliary single machine next to each other. Schedule S_A is cut into pieces of length T_0 and each such piece is allocated to the next one of the m original machines. In the resulting schedule, let m' be such a number of machines, $m' \leq m$, that the total load of each machine M_i , $1 \leq i \leq m'$, is equal to T_0 and is less than T_0 for each machine M_i , $m' < i \leq m$. More formally, for $T_0 = \frac{1}{m}p(N)$ define $m' = m$. If $T_0 = \max \{p(j) \mid j \in N\} > \frac{1}{m}p(N)$, identify the machine $M_{m'}$ such that $m' < m$ and $m'T_0 < p(N) \leq (m' + 1)T_0$. In any case, assign a segment of schedule S_A in the time interval $[(i - 1)T_0, iT_0]$ to be processed on machine M_i , $i = 1, 2, \dots, m'$, from time zero. If $m' < m$, assign the last segment in the interval $[m'T_0, p(N)]$ to machine $M_{m'+1}$ and in the case $m' + 2 \leq m$ assign no jobs to any of the machines M_i , $m' + 2 \leq i \leq m$. The resulting schedule has at most $m - 1$ preemptions. No preempted job is processed by two or more machines simultaneously, so that the resulting schedule is feasible. The algorithm takes $O(n)$ time.

3 | RELATED WORK

In this section, we formulate the original Horn's conditions for the existence of a feasible schedule for problem $P|r(j), pmtn, C(j) \leq D|-$, in which the jobs have individual release dates $r(j)$, $j \in N$. Further in this paper, for a problem with release dates it is assumed that there are h distinct release dates $R^{(1)}, R^{(2)}, \dots, R^{(h)}$ such that

$$R^{(1)} < R^{(2)} < \dots < R^{(h)},$$

and for completeness we add $R^{(h+1)} = D$. We also use notation $p(Q) = \sum_{j \in Q} p(j)$ for the total processing time of the jobs from a nonempty set $Q \subseteq N$. For completeness, define $p(\emptyset) = 0$.

Problem $P|r(j), pmtn, C(j) \leq D|-$ is independently studied in [7] and [10]. In fact, Horn [10] formulates his findings for the "dual" problem of the existence of a feasible schedule, provided that all jobs are available at time zero and have individual due dates. Below, we interpret his results in terms of problem $P|r(j), pmtn, C(j) \leq D|-$.

Following [10], given an instance of problem $P|r(j), pmtn, C(j) \leq D|-$, let $g^{(z)}(j)$ denote the *minimum amount of processing* of job J_j , $j \in N$, which *must* be completed in the interval $[R^{(z)}, R^{(h+1)}]$, $1 \leq z \leq h$, in any feasible schedule. Clearly, if job J_j arrives no earlier than $R^{(z)}$, then to be completed by time $R^{(h+1)} = D$ it must be fully processed in the interval $[R^{(z)}, R^{(h+1)}]$. Alternatively, if $r(j) < R^{(z)}$, then job J_j cannot be processed before time $R^{(z)}$ for more than $R^{(z)} - r(j)$ time units. Thus, for any $j \in N$ and any z , $1 \leq z \leq h$, we have that

$$g^{(z)}(j) = \begin{cases} p(j), & \text{if } r(j) \geq R^{(z)}, \\ \max \{0, p(j) - (R^{(z)} - r(j))\}, & \text{if } r(j) < R^{(z)}. \end{cases} \quad (1)$$

Horn [10] uses the quantities (1) to formulate the conditions presented below in Theorem 1.

Theorem 1. For problem $P|r(j), pmtn, C(j) \leq D|-$, a feasible schedule exists if and only if

- the inequality

$$p(j) \leq R^{(h+1)} - r(j), \quad (2)$$

holds for every job J_j , $j \in N$,

- and the inequality

$$\sum_{j \in N} g^{(z)}(j) \leq m(R^{(h+1)} - R^{(z)}) \quad (3)$$

holds for every interval $[R^{(z)}, R^{(h+1)}]$, $1 \leq z \leq h$.

Another approach employed in [10] links problem $P|r(j), pmtn, C(j) \leq D|-$ to finding the maximum flow in a special network.

Consider a generic network $G = (V, A)$ with the set of vertices V and the set of arcs A , in which the set V includes the source s and the sink t . The capacity of arc (v, v') is denoted by $\mu(v, v')$, which can be infinite for some arcs. Recall that a flow f is a function $f : A \rightarrow \mathbb{R}$ that assigns real numbers to arcs. A flow $f : A \rightarrow \mathbb{R}$ is said to be *feasible* if it satisfies the *capacity constraints*

$$0 \leq f(v, v') \leq \mu(v, v'), \quad (v, v') \in A, \quad (4)$$

and the *flow balance constraints*

$$\sum_{v' \in V, (v, v') \in A} f(v, v') = \sum_{v' \in V, (v', v) \in A} f(v', v), \quad v \in V \setminus \{s, t\}. \quad (5)$$

In the *maximum flow problem*, it is required to find a feasible flow of the maximum value, where the *value* of a flow f is the total flow on the arcs that leave the source (or, equivalently, enter the sink):

$$\text{the value of flow } f = \sum_{v' \in V, (s, v') \in A} f(s, v') = \sum_{v \in V, (v, t) \in A} f(v, t).$$

A partition (S, T) of the node set V such that $s \in S$ and $t \in T$ is called an *s-t cut*. For an *s-t cut* $(S, V \setminus S)$, we define the arc set $A(S, V \setminus S)$ by

$$A(S, V \setminus S) = \{(u, v) \in A \mid u \in S, v \in V \setminus S\},$$

and the capacity $\kappa(S) \in \mathbb{R} \cup \{+\infty\}$ of the *s-t cut* by

$$\kappa(S) = \sum_{(u, v) \in A(S, V \setminus S)} \mu(u, v).$$

Thus, the capacity $\kappa(S)$ of an *s-t cut* (S, T) is defined as the total capacity of the arcs that go from the nodes of set S to the nodes of set T .

An *s-t cut* (S, T) is called a *minimum s-t cut* if its capacity $\kappa(S)$ is the minimum among all *s-t cuts* in G . The maximum-flow minimum-cut theorem, the most well-known statement of network optimization, asserts that the value of the maximum flow is equal to the capacity of a minimum *s-t cut*.

Following [10], given an instance of problem $P|r(j), pmtn, C(j) \leq D|-$, define a network $G_0 = (V, A)$ in which the vertex set V contains

- the source s and the sink t ;
- n job nodes $j = 1, 2, \dots, n$;
- h interval nodes $I^{(z)}$, each associated with the interval $[R^{(z)}, R^{(z+1)}]$, $1 \leq z \leq h$.

Throughout this paper, we denote the length of an interval $[R^{(z)}, R^{(z+1)}]$ by $\ell^{(z)}$, $1 \leq z \leq h$. In network G_0 , the set A of arcs is $A = A^s \cup A' \cup A^t$ and their capacities are as follows:

$$\begin{aligned} A^s &= \{(s, j) \mid 1 \leq j \leq n\}, & \mu(s, j) &= p(j); \\ A' &= \{(j, I^{(z)}) \mid 1 \leq j \leq n, 1 \leq z \leq h, r(j) \leq R^{(z)}\}, & \mu(j, I^{(z)}) &= \ell^{(z)}; \\ A^t &= \{(I^{(z)}, t) \mid 1 \leq z \leq h\}, & \mu(I^{(z)}, t) &= m\ell^{(z)}. \end{aligned}$$

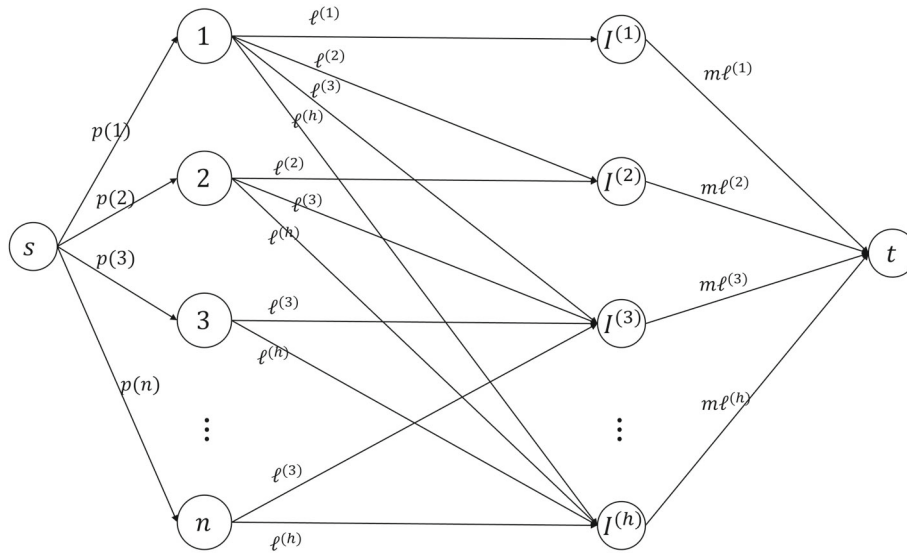
The structure of network G_0 is shown in Figure 1. The following statement comes from [10].

Theorem 2. *For problem $P|r(j), pmtn, C(j) \leq D|-$, a feasible schedule exists if and only if a maximum flow in network G_0 has the value $p(N)$.*

The two types of conditions in Theorems 1 and 2 look very different and they are treated in [10] from different principles. The proof of sufficiency of the conditions in Theorem 1 is a technical inductive argument, while the formal proof of Theorem 2 is not presented in [10]. There is no apparent link between the two types of conditions, while they both are aimed at checking feasibility of the same problem.

4 | A NEW UNIFIED NETWORK-FLOW-BASED THEOREM FOR HORN'S FEASIBILITY CONDITIONS

In this section, we prove a statement that establishes the equivalence of both original Horn's conditions.

FIGURE 1 Network $G_0 = (V, A)$.

For our purpose it is convenient to reformulate Theorem 2 in terms of a minimum cut as follows. For $S \subseteq V$ with $s \in S$, $t \notin S$, let $\kappa(S)$ be the capacity of the s - t cut $(S, V \setminus S)$.

Theorem 3. For problem $P|r(j), pmtn, C(j) \leq D|$ -, a feasible schedule exists if and only if in network $G_0 = (V, A)$ the inequality

$$p(N) \leq \kappa(S) \quad (6)$$

holds for every $S \subseteq V$ with $s \in S$, $t \notin S$.

We link the inequalities in Theorem 1 with the special cuts in network G_0 . In what follows, we often denote the minimum cut in network G_0 by $(S^*, V \setminus S^*)$.

For each z , $1 \leq z \leq h$, define the index set

$$N_+^{(z)} = \{j \in N \mid r(j) < R^{(z)}, p(j) \geq R^{(z)} - r(j)\}$$

such that each job $J_j, j \in N_+^{(z)}$, arrives before time $R^{(z)}$ but cannot be completed by time $R^{(z)}$, and the index set

$$N_-^{(z)} = \{j \in N \mid r(j) \geq R^{(z)}\}$$

such that each job $J_j, j \in N_-^{(z)}$, arrives no earlier than time $R^{(z)}$.

Notice that for each $j \in N$ such that $p(j) + r(j) < R^{(z)}$ it follows from (1) that $g^{(z)}(j) = 0$, so that

$$\sum_{j \in N} g^{(z)}(j) = \sum_{j \in N_-^{(z)}} g^{(z)}(j) + \sum_{j \in N_+^{(z)}} g^{(z)}(j) = \sum_{j \in N_-^{(z)}} p(j) + \sum_{j \in N_+^{(z)}} (p(j) + r(j) - R^{(z)}). \quad (7)$$

Define

$$S_j = \{s, j\}, \quad j = 1, 2, \dots, n;$$

$$S^{(z)} = \{s\} \cup N_+^{(z)} \cup N_-^{(z)} \cup \{I^{(z')} \mid z \leq z' \leq h\}.$$

For illustration, consider an instance with $n = 4$ jobs that have distinct release dates, so that $R^{(z)} = r(z)$, $1 \leq z \leq 4$. Set $S_3 = \{s, 3\}$ is shown in Figure 2. Assuming that $p(1) + r(1) < R^{(3)}$, set $S^{(3)}$ is $\{s\} \cup \{2\} \cup \{3, 4\} \cup \{I^{(3)}, I^{(4)}\}$ and it is shown in Figure 3.

The statement that presents the unified conditions that link Theorems 1 and 2 is given below.

Theorem 4. For problem $P|r(j), pmtn, C(j) \leq D|$ -, the following statements are equivalent:

- a feasible schedule exists;
- the inequality (2) holds for every job $J_j, j \in N$, and the inequality (3) holds for every interval $[R^{(z)}, R^{(h+1)}]$, $1 \leq z \leq h$;
- for each $j \in N$, the inequality

$$p(N) \leq \kappa(S_j) \quad (8)$$

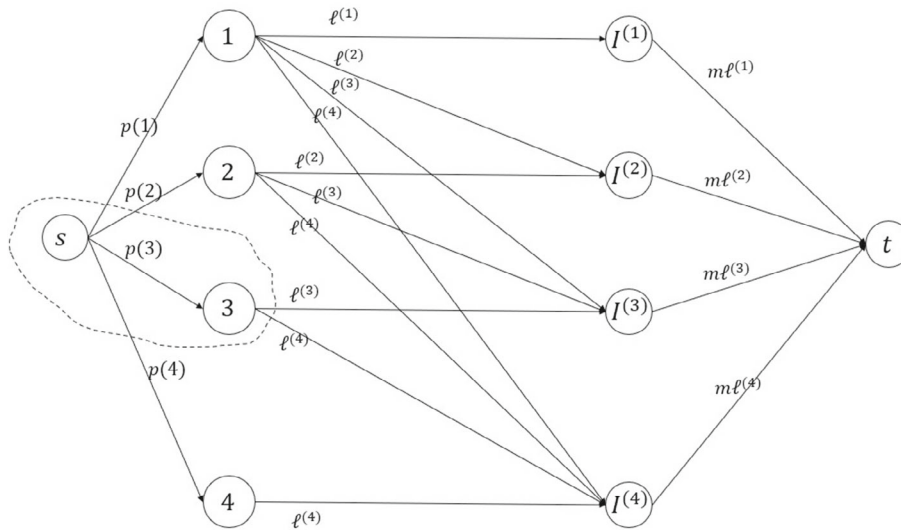


FIGURE 2 Vertex set S_3 .

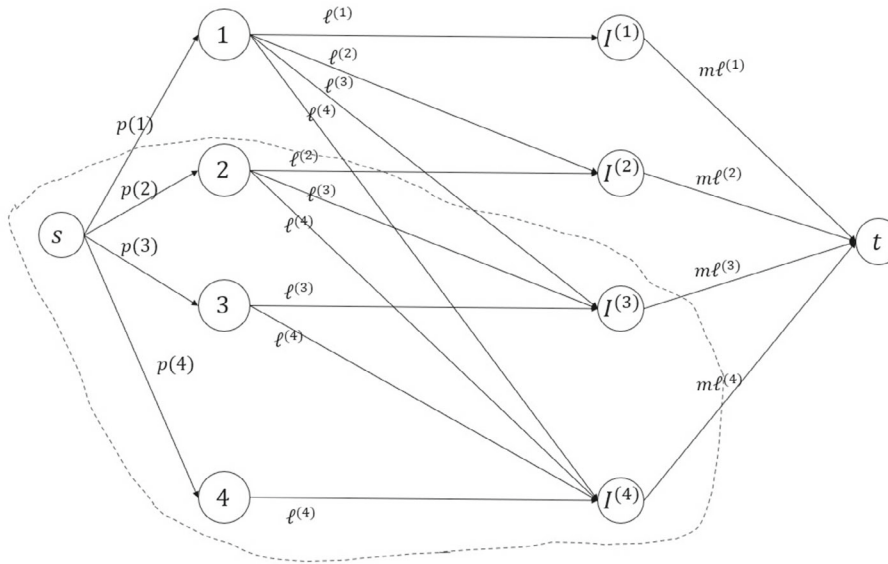


FIGURE 3 Vertex set $S^{(3)}$.

holds, and for each $z, 1 \leq z \leq h$, the inequality

$$p(N) \leq \kappa(S^{(z)}) \tag{9}$$

holds;

(d) for network G_0 the inequality (6) holds for every $S \subseteq V$ with $s \in S, t \notin S$.

We split the proof of Theorem 4 into several lemmas.

Lemma 1. In Theorem 4, statement (a) implies statement (b).

Proof. In a feasible schedule, any job $J_j, j \in N$, must be fully processed in the interval between its arrival time and the deadline D ; hence, the condition (2) follows. Additionally, in a feasible schedule the processing capacity must be sufficient to perform the minimum processing of all jobs that must be done after time $R^{(z)}, 1 \leq z \leq h$; hence, the condition (3) holds. ■

Lemma 2. In Theorem 4, statement (b) implies statement (c).

Proof. In fact, we prove a more general property: For each $j \in N$, condition (2) is equivalent to the inequality (8) and for each $z, 1 \leq z \leq h$, condition (3) is equivalent to the inequality (9).

For a fixed $j \in N$, by the definition of network G_0 , the cut $(S_j, V \setminus S_j)$ consists of all arcs that leave the source except the arc (s, j) and also of all arcs (j, I^z) with $r(j) \leq R^{(z)}$. Thus,

$$\begin{aligned} \kappa(S_j) - p(N) &= \sum_{j' \in N \setminus \{j\}} \mu(s, j') + \sum_{z: R^{(z)} \geq r(j)} \mu(j, I^{(z)}) - p(N) \\ &= p(N \setminus \{j\}) + \sum_{z: R^{(z)} \geq r(j)} (R^{(z+1)} - R^{(z)}) - p(N) \\ &= R^{(h+1)} - r(j) - p(j). \end{aligned}$$

Hence, the inequality (8) is equivalent to the condition (2).

Fix a z , $1 \leq z \leq h$. In network G_0 , a node $j \in N^{(z)}$ is not connected to any of the nodes of set $\{I^{(z')} \mid 1 \leq z' < z\}$ and the cut $(S^{(z)}, V \setminus S^{(z)})$ contains no arcs to the nodes of set $\{I^{(z')} \mid z \leq z' \leq h\}$; thus, the cut $(S^{(z)}, V \setminus S^{(z)})$ contains no arcs leaving the nodes of set $N^{(z)}$.

On the other hand, the cut $(S^{(z)}, V \setminus S^{(z)})$ consists of the arcs (s, j) for each $j \in N \setminus (N_+^{(z)} \cup N_-^{(z)})$, of the arcs that link each node $j \in N_+^{(z)}$ to the interval nodes of the set $\{I^{(z')} \mid 1 \leq z' < z, R^{(z)} \geq r(j)\}$, and of the arcs that link each node of the set $\{I^{(z')} \mid z \leq z' \leq h\}$ to the sink t . Thus, we have

$$\begin{aligned} \kappa(S^{(z)}) - p(N) &= \sum_{j \in N \setminus (N_+^{(z)} \cup N_-^{(z)})} \mu(s, j) + \sum_{j \in N_+^{(z)}} \sum_{z': z' < z, R^{(z')} \geq r(j)} \mu(j, I^{(z')}) + \sum_{z'=z}^h \mu(I^{(z')}, t) - p(N) \\ &= \sum_{j \in N \setminus (N_+^{(z)} \cup N_-^{(z)})} p(j) + \sum_{j \in N_+^{(z)}} \sum_{z': z' < z, R^{(z')} \geq r(j)} \ell^{(z')} + \sum_{z'=z}^h m \ell^{(z')} - p(N) \\ &= p(N \setminus (N_+^{(z)} \cup N_-^{(z)})) + \sum_{j \in N_+^{(z)}} (R^{(z)} - r(j)) + m(R^{(h+1)} - R^{(z)}) - p(N) \\ &= -p(N_+^{(z)} \cup N_-^{(z)}) + \sum_{j \in N_+^{(z)}} (R^{(z)} - r(j)) + m(R^{(h+1)} - R^{(z)}) \\ &= -\sum_{j \in N_-^{(z)}} p(j) - \sum_{j \in N_+^{(z)}} [p(j) - (R^{(z)} - r(j))] + m(R^{(h+1)} - R^{(z)}). \end{aligned}$$

Due to (7), we deduce

$$\kappa(S^{(z)}) - p(N) = -\sum_{j \in N} g^{(z)}(j) + m(R^{(h+1)} - R^{(z)}),$$

that is, the inequality (9) is equivalent to the condition (3). \blacksquare

Lemma 3. *In Theorem 4, statement (c) implies statement (d).*

Proof. We demonstrate that conditions (8) and (9) imply that for any cut $(S, V \setminus S)$ inequality $p(N) \leq \kappa(S)$ holds. We prove this by examining the structure of the possible cuts.

Case 1. Let $(S, V \setminus S)$ be an s - t cut such that $S \cap \{I^{(z)} \mid 1 \leq z \leq h\} = \emptyset$, that is, S does not include any interval nodes. We deduce

$$\begin{aligned} \kappa(S) - p(N) &= \sum_{j' \in N \setminus S} \mu(s, j') + \sum_{j' \in N \cap S} \sum_{z: R^{(z)} \geq r(j')} \mu(j', z) - p(N) \\ &= \sum_{j' \in N \setminus S} p(j') + \sum_{j' \in N \cap S} (R^{(h+1)} - r(j')) - p(N) \\ &= -p(N \cap S) + \sum_{j' \in N \cap S} (R^{(h+1)} - r(j')) \\ &= \sum_{j' \in N \cap S} (R^{(h+1)} - r(j') - p(j')). \end{aligned}$$

Recall that conditions (8) hold and that they are equivalent to conditions (2), as proved in Lemma 2. Thus, inequality $r(j) + p(j) \leq R^{(h+1)}$ holds for any $j \in N$, and hence (6) holds.

Case 2. Let $(S, V \setminus S)$ be an s - t cut such that $S \cap N = \emptyset$, that is, S does not include any job nodes. Then

$$\begin{aligned} \kappa(S) - p(N) &= \sum_{j \in N} \mu(s, j) + \sum_{z: I^{(z)} \in S} \mu(I^{(z)}, t) - p(N) \\ &= \sum_{j \in N} p(j) + \sum_{z: I^{(z)} \in S} m(R^{(z+1)} - R^{(z)}) - p(N) = \sum_{z: I^{(z)} \in S} m\ell^{(z)} > 0, \end{aligned}$$

so that (6) holds as a strict inequality.

Case 3. Let $(S, V \setminus S)$ be an s - t cut such that S contains at least one job node and at least one interval node. Suppose $I^{(z)} \in S$. Let k denote the number of jobs $J_j, j \in S \cap N$, with $r(j) \leq R^{(z)}$. First, we prove that in this case the equality

$$\kappa(S) - \kappa(S \setminus \{I^{(z)}\}) = (m - k)(R^{(z+1)} - R^{(z)})$$

holds. Moreover, we show that the inequality $k \geq m$ holds if $(S, V \setminus S)$ is a minimum s - t cut.

Compare the capacities of the cuts $(S, V \setminus S)$ and $(S \setminus \{I^{(z)}\}, V \setminus (S \setminus \{I^{(z)}\}))$. We have

$$\begin{aligned} \kappa(S) - \kappa(S \setminus \{I^{(z)}\}) &= \mu(I^{(z)}, t) - \sum_{j \in S \cap N: r(j) \leq R^{(z)}} \mu(j, I^{(z)}) \\ &= m(R^{(z+1)} - R^{(z)}) - k(R^{(z+1)} - R^{(z)}) \\ &= (m - k)(R^{(z+1)} - R^{(z)}), \end{aligned} \quad (10)$$

as required. If $(S, V \setminus S)$ is a minimum s - t cut, then we have

$$0 \geq \kappa(S) - \kappa(S \setminus \{I^{(z)}\}) = (m - k)(R^{(z+1)} - R^{(z)}), \quad (11)$$

which, together with $R^{(z+1)} - R^{(z)} > 0$, implies that $k \geq m$.

In what follows, we consider a minimum s - t cut $(S^*, V \setminus S^*)$ which satisfies the conditions of Case 3 stated above. Splitting our consideration into two subcases, we show that the total capacity of any minimum s - t cut $(S^*, V \setminus S^*)$ is equal to $\kappa(S^{(z)})$, and thus $p(N) \leq \kappa(S^{(z)})$.

Case 3.1. Let $I^{(z)} \in S^*$ and $I^{(z+1)} \notin S^*$ for some $z, 1 \leq z < h$. We show that in this case, that is, if set S^* does not contain a pair of consecutively numbered interval nodes, the inequality

$$\kappa(S^* \cup \{I^{(z+1)}\}) \leq \kappa(S^*)$$

holds, that is, $I^{(z+1)}$ can be added to S^* without increasing the capacity of the resulting cut.

Let k (respectively, \tilde{k}) be the number of jobs $J_j, j \in S^* \cap N$, with $r(j) \leq R^{(z)}$ (respectively, with $r(j) \leq R^{(z+1)}$). Then, $k \leq \tilde{k}$ holds. Since $(S^*, V \setminus S^*)$ is a minimum s - t cut with $I^{(z)} \in S^*$, it follows from (11) that $k \geq m$. Applying (10) to sets $S^* \cup \{I^{(z+1)}\}$ and S^* , we obtain

$$\kappa(S^* \cup \{I^{(z+1)}\}) - \kappa(S^*) = (m - \tilde{k})(R^{(z+2)} - R^{(z+1)}) \leq 0,$$

where the inequality is due to $\tilde{k} \geq k \geq m$ and $R^{(z+2)} - R^{(z+1)} > 0$.

Case 3.2. Let $(S^*, V \setminus S^*)$ be a minimum s - t cut such that $I^{(z)}, I^{(z+1)}, \dots, I^{(h)} \in S^*$ and $I^{(1)}, I^{(2)}, \dots, I^{(z-1)} \notin S^*$ for some $z, 1 \leq z \leq h$.

Assume that $(S^{(z)} \setminus S^*) \cap N \neq \emptyset$. Then there exists some j that belongs to $(S^{(z)} \setminus S^*) \cap N = (N_+^{(z)} \cup N_-^{(z)}) \setminus S^*$. If $j \in N_+^{(z)} \setminus S^*$, then $r(j) < R^{(z)}$ and $p(j) \geq R^{(z)} - r(j)$ hold, and therefore

$$\kappa(S^* \cup \{j\}) - \kappa(S^*) = \sum_{z': r(j) \leq R^{(z')} < R^{(z)}} \mu(j, I^{(z')}) - \mu(s, j) = (R^{(z)} - r(j)) - p(j) \leq 0.$$

If $j \in N_-^{(z)} \setminus S^*$, that is, $r(j) \geq R^{(z)}$, then in network G_0 there is no arc from node j to the nodes of the set $\{I^{(1)}, I^{(2)}, \dots, I^{(z-1)}\}$, and therefore

$$\kappa(S^* \cup \{j\}) - \kappa(S^*) = -\mu(s, j) = -p(j) \leq 0.$$

In either case, we have that $\kappa(S^* \cup \{j\}) \leq \kappa(S^*)$, so that $(S^* \cup \{j\}, V \setminus (S^* \cup \{j\}))$ is also a minimum s - t cut. Thus, for the minimum cut $(S^*, V \setminus S^*)$ we can add all nodes $j \in (S^{(z)} \setminus S^*) \cap N$ to set S^* without increasing the capacity of the cut.

Assume now that $(S^{(z)} \setminus S^*) \cap N = \emptyset$ but there exists some $j \in (S^* \setminus S^{(z)}) \cap N$. Then, $j \notin N_+^{(z)} \cup N_-^{(z)}$ so that $p(j) + r(j) < R^{(z)}$. Therefore, we obtain

$$\begin{aligned} \kappa(S^* \setminus \{j\}) - \kappa(S^*) &= \mu(s, j) - \sum_{z': z' < z, R^{(z')} \geq r(j)} \mu(j, I^{(z')}) \\ &= p(j) - (R^{(z)} - r(j)) < 0, \end{aligned}$$

a contradiction to the minimality of the s - t cut $(S^*, V \setminus S^*)$. Hence, we have that $(S^* \setminus S^{(z)}) \cap N = \emptyset$, which along with $(S^{(z)} \setminus S^*) \cap N = \emptyset$ implies that $S^* = S^{(z)}$, so that $(S^{(z)}, V \setminus S^{(z)})$ is a minimum s - t cut.

Thus, we have proved that for any s - t cut its total capacity is no less than $p(N)$, that is, (6) holds for any set $S \subseteq V$, as required. ■

Lemma 4. *In Theorem 4, statement (d) implies statement (a).*

Proof. We know that the capacity of any cut in network G_0 is no less than $p(N)$. It is clear that $S^* = \{s\}$ defines a minimum s - t cut $(S^*, V \setminus S^*)$ with $\kappa(S^*) = p(N)$, that is, there exists a maximum flow f of value $p(N)$. For $j \in N$, let the flow through the arc $(j, I^{(z)})$ such that $r(j) \leq I^{(z)}$ be equal to $p^{(z)}(j)$. In terms of the original scheduling problem, $p^{(z)}(j)$ is the duration of processing of job J_j in the interval $[R^{(z)}, R^{(z+1)}]$.

The actual feasible schedule S_0 can be found by solving a sequence of auxiliary problems $P^{(z)}$, $1 \leq z \leq h$. Each problem $P^{(z)}$ is in fact problem $P|pmtn|C_{\max}$ of processing jobs J_j with $p^{(z)}(j) > 0$ in the interval $[R^{(z)}, R^{(z+1)}]$ and can be solved by McNaughton's algorithm outlined in Section 2.

Since the flow through an arc (s, j) is equal to $p(j)$, $j \in N$, it follows that in schedule S_0 each job will be processed in full. For each interval $[R^{(z)}, R^{(z+1)}]$, the value $p^{(z)}(j)$, that is, the duration of processing of each job J_j in that interval, does not exceed the length $\ell^{(z)}$ of the interval. Additionally, the total processing amount assigned to an interval $[R^{(z)}, R^{(z+1)}]$, equal to $\sum_{j: r(j) \geq R^{(z)}} p^{(z)}(j)$, does not exceed $m\ell^{(z)}$, the total processing capacity of the interval. Thus, schedule S_0 is feasible. ■

Notice that in network G_0 there are $O(n)$ vertices, therefore the maximum flow in G_0 can be found in $O(n^3)$ time, for example, by the algorithm in [15]. In [10], in order to prove the sufficiency of the conditions (2) and (3) an iterative construction procedure is given that finds a feasible schedule, provided that the conditions hold. The running time of that procedure is not estimated in [10]. However, due to the equivalency between both types of conditions presented in [10], we deduce that finding a feasible schedule based on any of these conditions requires $O(n^3)$ time. This, however, does not mean that problem $P|r(j), pmtn, C(j) \leq D|$ — and even a more general problem $Q|r(j), pmtn, C(j) \leq D|$ — with uniform machines cannot be solved faster; see, for example, a very efficient algorithms by Sahni [20] and by Labetoulle et al. [16], respectively. Notice also that in [16] for problem $Q|r(j), pmtn, C(j) \leq D|$ — the generalization of the parameter denoted in our paper by $g^{(z)}(j)$ is introduced, as the minimum amount of processing of a job which must be completed in the interval $[R^{(z)}, R^{(h+1)}]$. However, that paper gives preference to the algorithmic aspects of finding a feasible schedule rather than formulating and proving the necessary and sufficient conditions similar to those in Theorem 1.

5 | EXTENSION OF HORN'S CONDITIONS TO PREEMPTIVE SCHEDULING UNDER RESOURCE CONSTRAINTS

We now turn to an enhanced model of parallel machine scheduling with resource constraints. We start with some preliminary information.

5.1 | Definitions and preliminaries

Let us extend the model of preemptive scheduling on parallel machines in the following way. There are Q renewable resources and at any time moment exactly one unit of each resource is available. Some jobs at any time of their processing must consume one unit of exactly one of these Q resources. An instance may also contain n_0 jobs which do not require these resources; however, it is convenient to think of any such a job as a job that consumes one unit of a unique additional resource k , $Q + 1 \leq k \leq Q + n_0$. Such an interpretation allows us to denote $q = Q + n_0$ as the total number of resources. Therefore, for the set of jobs the index set N can be seen as partitioned into q disjoint sets N_1, \dots, N_q such that each job of set N_k at any time of its processing requires one unit of resource k , $1 \leq k \leq q$. No two jobs of set N_k , $1 \leq k \leq q$, that is, the jobs that require the same resource, cannot be processed simultaneously.

For the scheduling problems under the resource constraints, the notation that has become standard since its first appearance in [2] places into the middle field β of the $\alpha | \beta | \gamma$ notation described in Section 2 a string that specifies the rules of resource usage by addressing three parameters.

Widely accepted, that notation, however, does not provide enough details in order to distinguish between various versions of the resource-constrained scheduling problems. Following [24], we adopt the extended scheme that adds to the middle field β a string of the form “res $\rho_1 \rho_2 \rho_3 \rho_4$ ”, where

- ρ_1 is the number of available renewable resources;
- ρ_2 is an upper bound on the number of resources a job may need;
- ρ_3 is an upper bound on the number of units of any resource available at a time;
- ρ_4 is an upper bound on the number of units of any resource that can be consumed by a job at a time.

The value of each of these parameters is either a known constant or the symbol “.”; in the latter case the value of the parameter is variable (part of the input). In accordance with this updated scheme, for the resource-constrained problems considered in this paper the string “res · 111” is used. This notation implies that

- $\rho_1 = \text{“.”}$, that is, there are several renewable resources (we use the variable q to denote their number);
- $\rho_2 = 1$, that is, each job needs at most one resource at any time of its processing;
- $\rho_3 = 1$, that is, one unit of each resource is available at a time;
- $\rho_4 = 1$, that is, a job that requires a resource consumes one unit of that resource at any time of its processing.

In the traditional scheme the parameter ρ_2 was missing. Therefore, in most of the previously considered models with resource constraints it was assumed that a job might need any number of resources.

In fact, there is a considerable interest, both from the theoretical and practical perspective, in scheduling models in which a job may only need a fixed number of resources, in particular in those models that are described by the string “res · 111”.

A natural meaningful interpretation of problem $P|\text{res} \cdot 111|C_{\max}$ is related to human resource management. The projects (jobs) can be performed by any of the available m teams (machines). However, to be able to perform certain projects a team must additionally include an extra employee with a special skill. For each skill k , $1 \leq k \leq q$, there is exactly one employee who possesses this skill, that is, such an employee can be seen as a renewable resource k . See [24].

Hebrard et al. [8] motivate their study of problem $P|\text{res} \cdot 111|C_{\max}$ by the problem that arises in satellite data download management. Janssen et al. [14] study the problem with the same environment but with the objective of the sum of the completion times and present a motivation scenario of using photolithography equipment in the microelectronic industry.

The problems with resources can be considered as special cases of multiprocessor task scheduling, where jobs have prescribed processing modes, with specific subsets of machines associated with each job; see, for example, [11, 12].

A review of resource-constrained scheduling on parallel machines can be found in [5]. For the purposes of this paper, we only mention the results on the models with preemption allowed. The most general problem $Rm|pmtn, \text{res} \cdot \dots \cdot|C_{\max}$ with a fixed number m of unrelated machines reduces to a linear programming problem with $O(n^m)$ variables and is therefore polynomially solvable. If the number of machines is variable then problem $Q|pmtn, \text{res} 1111|C_{\max}$ with uniform machines and a single additional resource is solvable in $O(n + m \log m)$ time; see the survey [2] and the references within. The approximability results for problem $P|pmtn, \text{res} \cdot \dots \cdot|C_{\max}$ are presented in [13].

In the remainder of this paper, we present a polynomial-time algorithm for problem $P|pmtn, \text{res} \cdot 111, C(j) \leq D|-$ and the necessary and sufficient conditions for the existence of a feasible schedule for problem $P|r(j), pmtn, \text{res} \cdot 111, C(j) \leq D|-$, similar to those discussed in Section 4.

5.2 | Zero release dates

In this subsection, we show that problem $P|pmtn, \text{res} \cdot 111, C(j) \leq D|-$ and its optimization counterpart, problem $P|pmtn, \text{res} \cdot 111|C_{\max}$, can be solved by a simple adaptation of McNaughton’s algorithm for the classical problem $P|pmtn|C_{\max}$ with no resource constraints, presented in Section 2.

Similarly to problem $P|pmtn|C_{\max}$, the makespan of any schedule S that is feasible for problem $P|pmtn, \text{res} \cdot 111|C_{\max}$ cannot be smaller than the average machine load $p(N)/m$. Additionally, for any k , $1 \leq k \leq q$, no two jobs that require the same resource k can be processed in parallel, that is,

$$C_{\max}(S) \geq \max \left\{ \frac{1}{m}p(N), \max \{p(N_k) \mid 1 \leq k \leq q\} \right\}. \quad (12)$$

In our consideration, we often rely on the so-called group technology approach. Given an instance of a problem with the “res · 111” constraints, associate each set N_k , the index set of jobs that require resource k , with a composite job V_k , $1 \leq k \leq q$. The processing times of these composite jobs are defined by

$$p(V_k) = p(N_k), \quad 1 \leq k \leq q.$$

Replace the original instance I of problem $P|pmtn, \text{res} \cdot 111|C_{\max}$ by instance I_G that consists of the composite jobs V_k , $1 \leq k \leq q$, described above. Solve problem $P|pmtn|C_{\max}$ for instance I_G by McNaughton’s algorithm. Let S_G^* be the resulting schedule of the composite jobs. We can convert schedule S_G^* into a schedule of the original jobs J_1, \dots, J_n that is optimal for

problem $P|pmtn, res \cdot 111|C_{\max}$ by replacing a composite job V_k (or its segment if it is processed with preemption) by the block of jobs of set $\{J_j | j \in N_k\}$ (one job maybe preempted). Clearly,

$$C_{\max}(S_G^*) = \max \left\{ \frac{1}{m}p(N), \max \{p(V_k) | 1 \leq k \leq q\} \right\}.$$

It follows from (12) that such a schedule is optimal for problem $P|pmtn, res \cdot 111|C_{\max}$ since its makespan is the same as $C_{\max}(S_G^*)$.

Thus, the following statement holds.

Theorem 5. *Problem $P|pmtn, res \cdot 111|C_{\max}$ is solvable in $O(n)$ time.*

It immediately follows that problem $P|pmtn, res \cdot 111, C(j) \leq D|-$ has a feasible solution if and only if $D \geq C_{\max}(S_G^*)$.

5.3 | Nonzero release dates

In this subsection, we consider problem $P|r(j), pmtn, res \cdot 111, C(j) \leq D|-$. We formulate the necessary and sufficient conditions for the existence of a feasible schedule. We start with the conditions that are similar to those in Theorem 1 developed in [10] for problem $P|r(j), pmtn, C(j) \leq D|-$ with no resource constraints. We also link problem $P|r(j), pmtn, res \cdot 111, C(j) \leq D|-$ with finding the maximum flow in a special network. Then we formulate and prove the unified necessary and sufficient feasibility conditions, thereby extending Theorem 4 to the problem with resource constraints.

Throughout this subsection, given a set Y and a logical condition λ , denote by $Y[\lambda]$ the subset of the elements of Y that satisfy λ . For example, $N_k[r(j) \leq R^{(z)}]$ denotes the index set of jobs $J_j, j \in N_k$, that require resource k and arrive no later than time $R^{(z)}, 1 \leq k \leq q, 1 \leq z \leq h$. We also use notation $N_k^{(z)} = N_k[r(j) = R^{(z)}]$ for the index set of jobs $J_j, j \in N_k$, that require resource k and arrive at time $R^{(z)}, 1 \leq k \leq q, 1 \leq z \leq h$.

We start with formulating an extension of Horn's conditions, that is, the necessary and sufficient conditions for the existence of a feasible schedule for problem $P|r(j), pmtn, res \cdot 111, C(j) \leq D|-$.

Let $g^{(z)}(N_k)$ denote the *minimum amount of processing* of the jobs of set $N_k, 1 \leq k \leq q$, which *must* be completed in the interval $[R^{(z)}, R^{(h+1)}], 1 \leq z \leq h$.

Lemma 5. *In any schedule that is feasible for problem $P|r(j), pmtn, res \cdot 111, C(j) \leq D|-$, the total processing time of the jobs of set N_k after time $R^{(z)}, 1 \leq k \leq q, 1 \leq z \leq h$, is at least*

$$g^{(z)}(N_k) = \max_{1 \leq z' \leq z} \left\{ p\left(N_k[r(j) \geq R^{(z')}]\right) - (R^{(z)} - R^{(z')}) \right\}.$$

Proof. For each $k, 1 \leq k \leq q$, and each $z, 1 \leq z \leq h$, set $N_k[r(j) \geq R^{(z)}]$ is a subset of jobs of set $\{J_j | j \in N_k\}$ that arrive no earlier than $R^{(z)}$. All these jobs must be fully processed in the interval $[R^{(z)}, R^{(h+1)}]$ with no overlap, so that

$$g^{(z)}(N_k) \geq p(N_k[r(j) \geq R^{(z)}]) = p\left(N_k^{(z)} \cup N_k^{(z+1)} \cup \dots \cup N_k^{(h)}\right).$$

We also need to take into account those jobs of set N_k that for some $z', 1 \leq z' < z$, arrive at or after $R^{(z')}$ but are not fully completed before $R^{(z)}$ and therefore must be processed in the interval $[R^{(z)}, R^{(h+1)}]$. Since these jobs can be processed in the interval $[R^{(z')}, R^{(z)}]$ for at most $R^{(z)} - R^{(z')}$ time units, it follows that the total processing time of the jobs of set $\{J_j | j \in N_k[r(j) \geq R^{(z')}]\}$ after time $R^{(z)}$ is at least $p\left(N_k[r(j) \geq R^{(z')}]\right) - (R^{(z)} - R^{(z')})$. ■

Based on the determined values of $g^{(z)}(N_k)$, we can formulate the following statement, which is an extension of Theorem 1.

Theorem 6. *For problem $P|r(j), pmtn, res \cdot 111, C(j) \leq D|-$, a feasible schedule exists if and only if for each $k, 1 \leq k \leq q$, the inequality*

$$g^{(z)}(N_k) \leq R^{(h+1)} - R^{(z)} \quad (13)$$

holds for each $z, 1 \leq z \leq h$, and the inequality

$$\sum_{k=1}^q g^{(z)}(N_k) \leq m(R^{(h+1)} - R^{(z)}) \quad (14)$$

holds for each $z, 1 \leq z \leq h$.

We refer to the conditions in Theorem 6 as *extended Horn's conditions*. They are part of the unified conditions formulated and proved in Theorem 8.

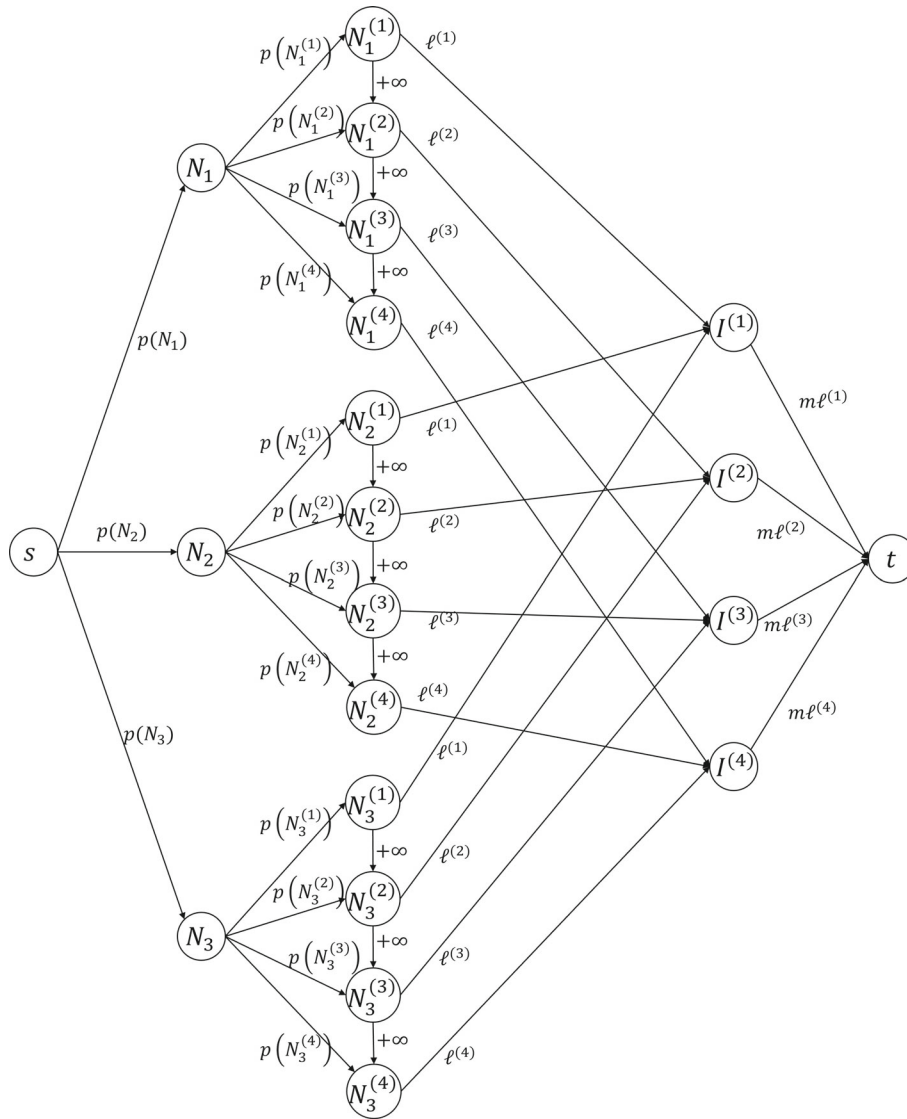


FIGURE 4 Network $G_{Res} = (V, A)$ with $q = 3$ and $h = 4$.

As in Section 3, define $\ell^{(z)} = R^{(z+1)} - R^{(z)}$, $z = 1, 2, \dots, h$. Create the network $G_{Res} = (V, A)$ with a set of vertices V and the set of arcs A . The set V includes

- the source s and the sink t ;
- q resource nodes N_k , each associated with the set of jobs that require resource k , $1 \leq k \leq q$;
- $q \times h$ resource-interval nodes $N_k^{(z)}$, each associated with the set of jobs that require resource k , $1 \leq k \leq q$, and have release date $r(j) = R^{(z)}$, $1 \leq z \leq h$;
- h interval nodes $I^{(z)}$, each associated with the interval $[R^{(z)}, R^{(z+1)}]$, $1 \leq z \leq h$.

The set A of arcs is $A = A^s \cup A_1 \cup A_2 \cup A_3 \cup A^t$ and the arc capacities are as follows:

$$\begin{aligned}
 A^s &= \{(s, N_k) \mid 1 \leq k \leq q\}, & \mu(s, N_k) &= p(N_k); \\
 A_1 &= \{(N_k, N_k^{(z)}) \mid 1 \leq k \leq q, 1 \leq z \leq h\}, & \mu(N_k, N_k^{(z)}) &= p(N_k^{(z)}); \\
 A_2 &= \{(N_k^{(z)}, N_k^{(z+1)}) \mid 1 \leq k \leq q, 1 \leq z \leq h-1\}, & \mu(N_k^{(z)}, N_k^{(z+1)}) &= +\infty; \\
 A_3 &= \{(N_k^{(z)}, I^{(z)}) \mid 1 \leq k \leq q, 1 \leq z \leq h\}, & \mu(N_k^{(z)}, I^{(z)}) &= \ell^{(z)}; \\
 A^t &= \{(I^{(z)}, t) \mid 1 \leq z \leq h\}, & \mu(I^{(z)}, t) &= m \ell^{(z)}.
 \end{aligned}$$

The structure of network G_{Res} is shown in Figure 4. The following statement is an extension of Theorem 2.

Theorem 7. For problem $P \mid r(j), pmtn, res \cdot 111, C(j) \leq D \mid -$, a feasible schedule exists if and only if a maximum flow in network G_{Res} has the value $p(N)$.

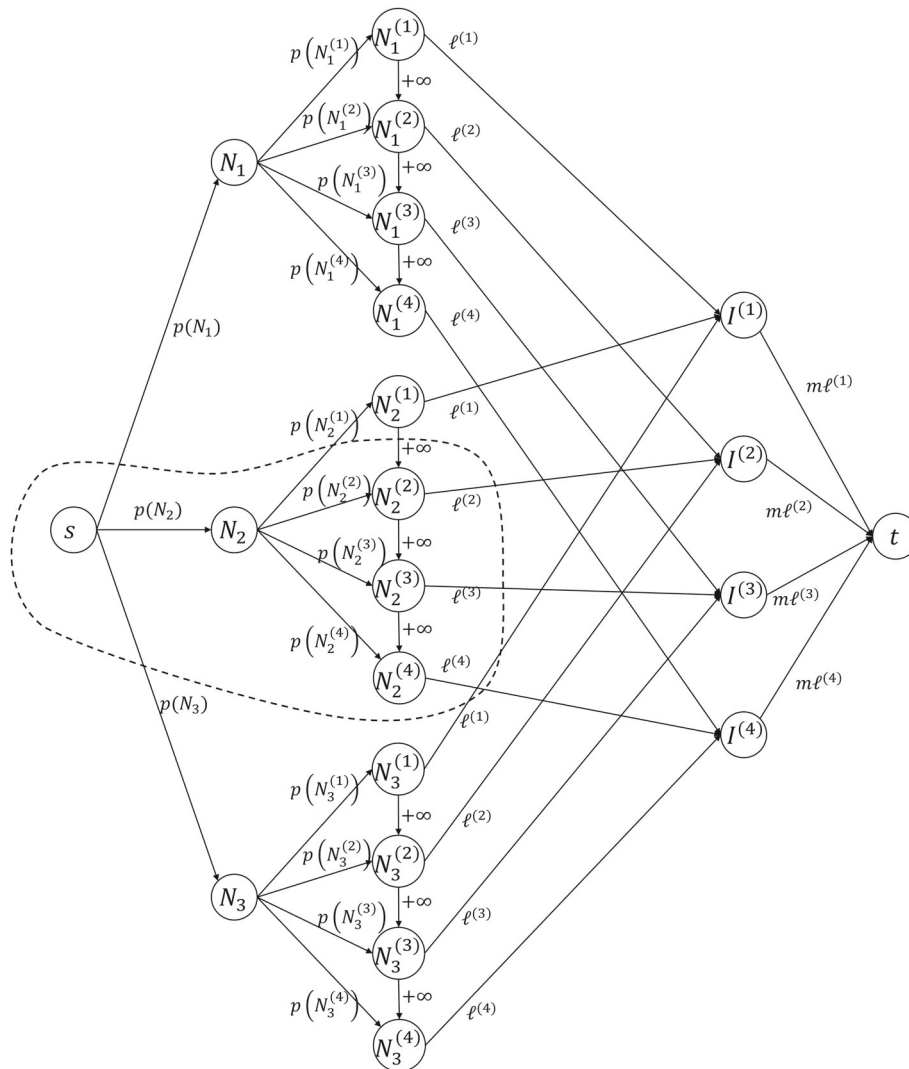


FIGURE 5 Vertex set $S_k^{(z)}$ with $k = 2, z = 2$.

Notice that under the conditions of Theorem 7 an analogue of Theorem 3 holds, which guarantees that a maximum flow in G_{Res} is equal to $p(N)$ if and only if the inequality (6) holds for any s - t cut $(S, V \setminus S)$.

Instead of proving Theorems 6 and 7 directly, we formulate and prove a statement similar to Theorem 4 that presents the unified conditions.

For each $z, 1 \leq z \leq h$, and each $k, 1 \leq k \leq q$, define a vertex set $S_k^{(z)}$ by

$$S_k^{(z)} = \{s\} \cup \{N_k\} \cup \{N_k^{(z)}, N_k^{(z+1)}, \dots, N_k^{(h)}\}.$$

For each $z, 1 \leq z \leq h$, and each vector (z_1, \dots, z_q) with $z_1, \dots, z_q \in \{1, 2, \dots, z\}$, we define a vertex set $S^{(z)}(z_1, \dots, z_q)$ by

$$S^{(z)}(z_1, \dots, z_q) = \{s\} \cup \{N_1, \dots, N_q\} \cup \bigcup_{k=1}^q \{N_k^{(z_k)}, N_k^{(z_k+1)}, \dots, N_k^{(h)}\} \cup \{I^{(z)}, \dots, I^{(h)}\}.$$

Sets $S_k^{(z)}$ and $S^{(z)}(z_1, \dots, z_q)$ for a small value of z are illustrated in Figures 5 and 6, respectively.

The statement that presents the unified conditions that link Theorems 6 and 7 is given below.

Theorem 8. For problem $P | r(j), p, m, n, res \cdot 111, C(j) \leq D | -$, the following statements are equivalent:

- (a) a feasible schedule exists;
- (b) for each $k, 1 \leq k \leq q$, and each $z, 1 \leq z \leq h$, the inequalities (13) and (14) hold;
- (c) for each $k, 1 \leq k \leq q$, and each $z, 1 \leq z \leq h$, the inequality

$$p(N) \leq \kappa \left(S_k^{(z)} \right) \tag{15}$$

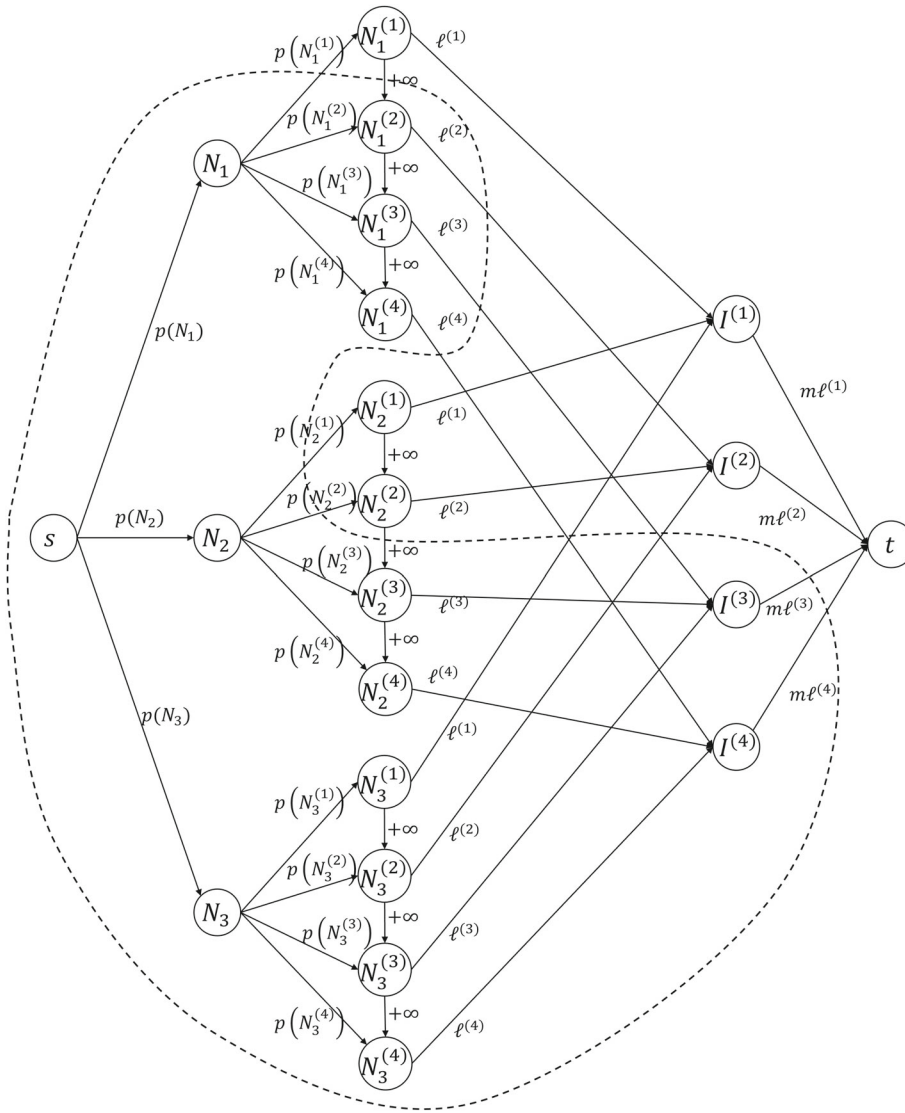


FIGURE 6 Vertex set $S^{(z)}(z_1, \dots, z_q)$ with $z_1 = 2, z_2 = 3, z_3 = 1, z = 3$.

holds, and for each $z, 1 \leq z \leq h$, and each vector (z_1, \dots, z_q) with $z_1, \dots, z_q \in \{1, 2, \dots, z\}$ the inequality

$$p(N) \leq \kappa(S^{(z)}(z_1, \dots, z_q)) \tag{16}$$

holds;

(d) for network G_{Res} the inequality (6) holds for every $S \subseteq V$ with $s \in S, t \notin S$.

As in Section 4, we split the proof into several lemmas.

Lemma 6. In Theorem 8, statement (a) implies statement (b).

Proof. By Lemma 5, in any feasible schedule the total processing time of the jobs of set N_k after time $R^{(z)}, 1 \leq k \leq q, 1 \leq z \leq h$, is at least $g^{(z)}(N_k)$. This amount must be fully processed in the interval $[R^{(z)}, R^{(h+1)}]$; hence, the condition (13) holds. Additionally, in a feasible schedule the processing capacity must be sufficient to perform the minimum processing of all jobs that must be done after time $R^{(z)}, 1 \leq z \leq h$; this implies the condition (14). ■

Lemma 7. In Theorem 4, statement (b) implies statement (c).

Proof. As in Section 4, we actually prove a more general property: For each $k, 1 \leq k \leq q$, and each $z, 1 \leq z \leq h$, condition (13) is equivalent to the inequality (15) and for each $z, 1 \leq z \leq h$, condition (14) is equivalent to the inequality (16) for each vector (z_1, \dots, z_q) with $z_1, \dots, z_q \in \{1, 2, \dots, z\}$.

Let us fix a resource $k \in \{1, 2, \dots, q\}$. The inequalities (13) can be rewritten as

$$\max_{1 \leq z' \leq z} \left\{ p\left(N_k^{(z')} \cup N_k^{(z'+1)} \cup \dots \cup N_k^{(h)}\right) - (R^{(z)} - R^{(z')}) \right\} \leq R^{(h+1)} - R^{(z)}, \quad 1 \leq z \leq h,$$

which is equivalent to

$$\sum_{u=z'}^h p(N_k^{(u)}) \leq R^{(h+1)} - R^{(z')}, \quad 1 \leq z' \leq z \leq h.$$

Changing the variables, we deduce the equivalent inequalities

$$\sum_{z'=z}^h p\left(N_k^{(z')}\right) \leq R^{(h+1)} - R^{(z)}, \quad 1 \leq z \leq h. \quad (17)$$

The s - t cut $(S_k^{(z)}, V \setminus S_k^{(z)})$ consists of all arcs $(s, N_{k'})$, $1 \leq k' \leq q$, $k' \neq k$, all arcs $(N_k, N_k^{(z')})$ for $1 \leq z' < z$ and all arcs $(N_k^{(z')}, I^{(z')})$ for $z \leq z' \leq h$. Thus,

$$\kappa(S_k^{(z)}) = (p(N) - p(N_k)) + \sum_{z'=1}^{z-1} p\left(N_k^{(z')}\right) + \sum_{z'=z}^h \ell_{z'}.$$

To prove (15) compute

$$\begin{aligned} \kappa(S_k^{(z)}) - p(N) &= -p(N_k) + \sum_{z'=1}^{z-1} p\left(N_k^{(z')}\right) + (R^{(h+1)} - R^{(z)}) \\ &= -\sum_{z'=1}^h p\left(N_k^{(z')}\right) + \sum_{z'=1}^{z-1} p\left(N_k^{(z')}\right) + (R^{(h+1)} - R^{(z)}) \\ &= -\sum_{z'=z}^h p\left(N_k^{(z')}\right) + (R^{(h+1)} - R^{(z)}), \end{aligned} \quad (18)$$

so that (17) is equivalent to the desired inequality (15).

Now we fix a $z \in \{1, 2, \dots, h\}$. The inequality (14) can be explicitly written as

$$\sum_{k=1}^q \max_{1 \leq z_k \leq z} \left\{ p\left(N_k^{(z_k)} \cup N_k^{(z_k+1)} \cup \dots \cup N_k^{(h)}\right) - (R^{(z)} - R^{(z_k)}) \right\} \leq m(R^{(h+1)} - R^{(z)}). \quad (19)$$

Select a vector (z_1, \dots, z_q) with $z_k \in \{1, 2, \dots, z\}$, $1 \leq k \leq q$. The s - t cut $(S^{(z)}(z_1, \dots, z_q), V \setminus S^{(z)}(z_1, \dots, z_q))$ includes all arcs $(N_k, N_k^{(z')})$ with $1 \leq z' < z_k$ and all arcs $(N_k^{(z')}, I^{(z')})$ for $z_k \leq z' < z$; recall that the nodes $I^{(z)}, \dots, I^{(h)}$ belong to $S^{(z)}(z_1, \dots, z_q)$. Additionally, the cut also includes all arcs $(I^{(z')}, t)$ for $z \leq z' \leq h$. Thus,

$$\begin{aligned} \kappa(S^{(z)}(z_1, \dots, z_q)) - p(N) &= \sum_{k=1}^q \left(\sum_{z' < z_k} p\left(N_k^{(z')}\right) + \sum_{z_k \leq z' < z} \ell_{z'} \right) + \sum_{z' \geq z} m \ell_{z'} - p(N) \\ &= \sum_{k=1}^q \left(\left(\sum_{z' < z_k} p\left(N_k^{(z')}\right) - p(N_k) \right) + (R^{(z)} - R^{(z_k)}) \right) + m(R^{(h+1)} - R^{(z)}) \\ &= -\sum_{k=1}^q \left(p\left(N_k^{(z_k)} \cup N_k^{(z_k+1)} \cup \dots \cup N_k^{(h)}\right) - (R^{(z)} - R^{(z_k)}) \right) + m(R^{(h+1)} - R^{(z)}), \end{aligned}$$

which implies the equivalence of (19) and (16). \blacksquare

Lemma 8. *In Theorem 4, statement (c) implies statement (d).*

Proof. We prove the lemma by showing that if the conditions (15) and (16) (or, equivalently, the conditions (13) and (14), respectively) hold, then the capacity of a minimum s - t cut in network G_{Res} is equal to $p(N)$.

Consider the s - t cut $(S', V \setminus S')$ with $S' = \{s\}$. The capacity $\kappa(S')$ of $(S', V \setminus S')$ is equal to $p(N)$. Hence, it suffices to show that the inequality

$$p(N) \leq \kappa(S^*) \quad (20)$$

holds for a minimum s - t cut $(S^*, V \setminus S^*)$. To prove that we show that there exists a minimum s - t cut with some nice structure.

First, observe that if for a minimum s - t cut $(S^*, V \setminus S^*)$ we have that $N_k^{(z)} \in S^*$ for some k , $1 \leq k \leq q$, and z , $1 \leq z \leq h - 1$, then $N_k^{(z+1)} \in S^*$. Indeed, if $N_k^{(z+1)} \notin S^*$ then $(S^*, V \setminus S^*)$ is not a minimum cut, since the arc set $A(S^*, V \setminus S^*)$ would contain the arc $(N_k^{(z)}, N_k^{(z+1)})$ of infinite capacity, implying that $\kappa(S^*) = +\infty$.

Thus, in a possible minimum s - t cut $(S^*, V \setminus S^*)$ set S^* for each k , $1 \leq k \leq q$, either does not contain the vertices of set $\{N_k^{(1)}, N_k^{(2)}, \dots, N_k^{(h)}\}$ or for some z_k contains all vertices of set $\{N_k^{(z_k)}, N_k^{(z_k+1)}, \dots, N_k^{(h)}\}$ and none of the vertices of set $\{N_k^{(1)}, \dots, N_k^{(z_k-1)}\}$. More formally,

$$S^* \cap \{N_k^{(1)}, N_k^{(2)}, \dots, N_k^{(h)}\} = \{N_k^{(z_k)}, N_k^{(z_k+1)}, \dots, N_k^{(h)}\}, \quad 1 \leq k \leq q, \tag{21}$$

for some $z_1, z_2, \dots, z_q \in \{1, 2, \dots, h, h + 1\}$, where $z_k = h + 1$ means that $S^* \cap \{N_k^{(1)}, N_k^{(2)}, \dots, N_k^{(h)}\} = \emptyset$.

In the rest of the proof, we assume that $(S^*, V \setminus S^*)$ is a minimum s - t cut that possesses this structure.

Let $(S, V \setminus S)$ be an s - t cut of a finite total capacity, $\kappa(S) < +\infty$. We show that if for some k , $1 \leq k \leq q$, vertex N_k does not belong to S , then it can be added to S without increasing the capacity of the resulting cut, that is, $\kappa(S \cup \{N_k\}) \leq \kappa(S)$.

To see this, suppose that we add node N_k to S . Then, the arc set $A(S \cup \{N_k\}, V \setminus (S \cup \{N_k\}))$ is obtained from the arc set $A(S, V \setminus S)$ by deleting the arc (s, N_k) of capacity $p(N_k)$ followed by adding some of the arcs $(N_k, N_k^{(z)})$, $1 \leq z \leq h$. Since the capacity of arc $(N_k, N_k^{(z)})$ is equal to $p(N_k^{(z)})$ and $\sum_{z=1}^h p(N_k^{(z)}) = p(N_k)$, we have that

$$\kappa(S \cup \{N_k\}) - \kappa(S) \leq -\mu(s, N_k) + \sum_{z=1}^h \mu(N_k, N_k^{(z)}) = -p(N_k) + \sum_{z=1}^h p(N_k^{(z)}) = 0,$$

as required. Thus, for the minimum cut $(S^*, V \setminus S^*)$ we can add all nodes N_k to set S^* without increasing the capacity of the cut, so that

$$\{N_1, \dots, N_q\} \subseteq S^*. \tag{22}$$

Due to (21) and (22) we deduce that the vertex set S^* contains the nodes

$$\{s\} \cup \{N_1, \dots, N_q\} \cup \bigcup_{k=1}^q \{N_k^{(z_k)}, N_k^{(z_k+1)}, \dots, N_k^{(h)}\},$$

and possibly some vertices in $\{I^{(1)}, \dots, I^{(h)}\}$. We consider the following two possible cases and show that (20) holds in each case.

Case 1. For the minimum cut $(S^*, V \setminus S^*)$ set S^* does not contain any of the interval nodes, that is, $S^* \cap \{I^{(1)}, \dots, I^{(h)}\} = \emptyset$.

In this case, we have

$$S^* = \{s\} \cup \{N_1, \dots, N_q\} \cup \bigcup_{k=1}^q \{N_k^{(z_k)}, N_k^{(z_k+1)}, \dots, N_k^{(h)}\}.$$

See Figure 7 for an illustration of a cut of such a structure.

Similarly to the proof of Lemma 7, we obtain

$$\begin{aligned} \kappa(S^*) - p(N) &= \sum_{k=1}^q \left(\sum_{z'=1}^{z_k-1} p(N_k^{(z')}) + \sum_{z'=z_k}^h \ell_{z'} \right) - p(N) \\ &= \sum_{k=1}^q \left(\sum_{z'=1}^{z_k-1} p(N_k^{(z')}) - p(N_k) + (R^{(h+1)} - R^{(z_k)}) \right) \\ &= \sum_{k=1}^q \left(- \sum_{z'=z_k}^h p(N_k^{(z')}) + (R^{(h+1)} - R^{(z_k)}) \right). \end{aligned}$$

For each k , $1 \leq k \leq q$, such that $z_k = h + 1$, we have that

$$- \sum_{z'=z_k}^h p(N_k^{(z')}) + (R^{(h+1)} - R^{(z_k)}) = R^{(h+1)} - R^{(h+1)} = 0,$$

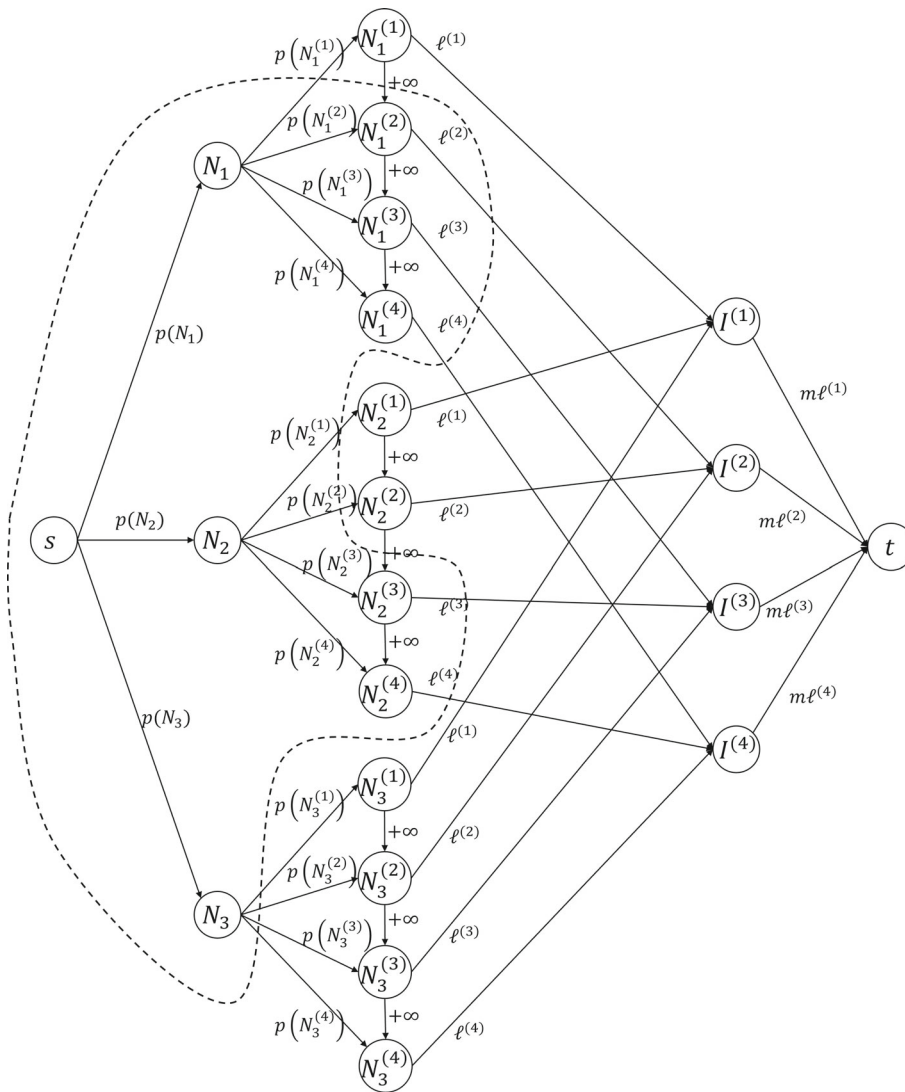


FIGURE 7 A minimum s - t cut $(S^*, V \setminus S^*)$ in Case 1.

while for each each $k, 1 \leq k \leq q$, such that $z_k \leq h$, we have that

$$-\sum_{z'=z_k}^h p(N_k^{z'}) + (R^{(h+1)} - R^{(z_k)}) = \kappa(S_k^{(z_k)}) - p(N) \geq 0,$$

where in the latter expression the equality is by (18) and the inequality is by (15). Hence, the inequality (20) follows.

Case 2. For the minimum cut $(S^*, V \setminus S^*)$ set S^* contains some of the interval nodes, that is, $S^* \cap \{I^{(1)}, \dots, I^{(h)}\} \neq \emptyset$. For illustration, see Figure 8.

For each $z, 1 \leq z \leq h$, compute

$$\eta_z = |S^* \cap \{N_1^{(z)}, N_2^{(z)}, \dots, N_q^{(z)}\}|,$$

the number of the resource-interval nodes contained in S^* . Due to (21), we have $\eta_1 \leq \eta_2 \leq \dots \leq \eta_h$, which implies that there exists some $z' \in \{1, 2, \dots, h, h + 1\}$ such that

$$\eta_1 \leq \dots \leq \eta_{z'-1} < m \leq \eta_{z'} \leq \dots \leq \eta_h;$$

if $z' = h + 1$ then we have $\eta_z < m$ for all z .

For each $z, 1 \leq z \leq h$, such that $I^{(z)} \in S^*$ and $\eta_z < m$ it follows that

$$\kappa(S^* \setminus \{I^{(z)}\}) - \kappa(S^*) = \eta_z \ell_z - m \ell_z < 0,$$

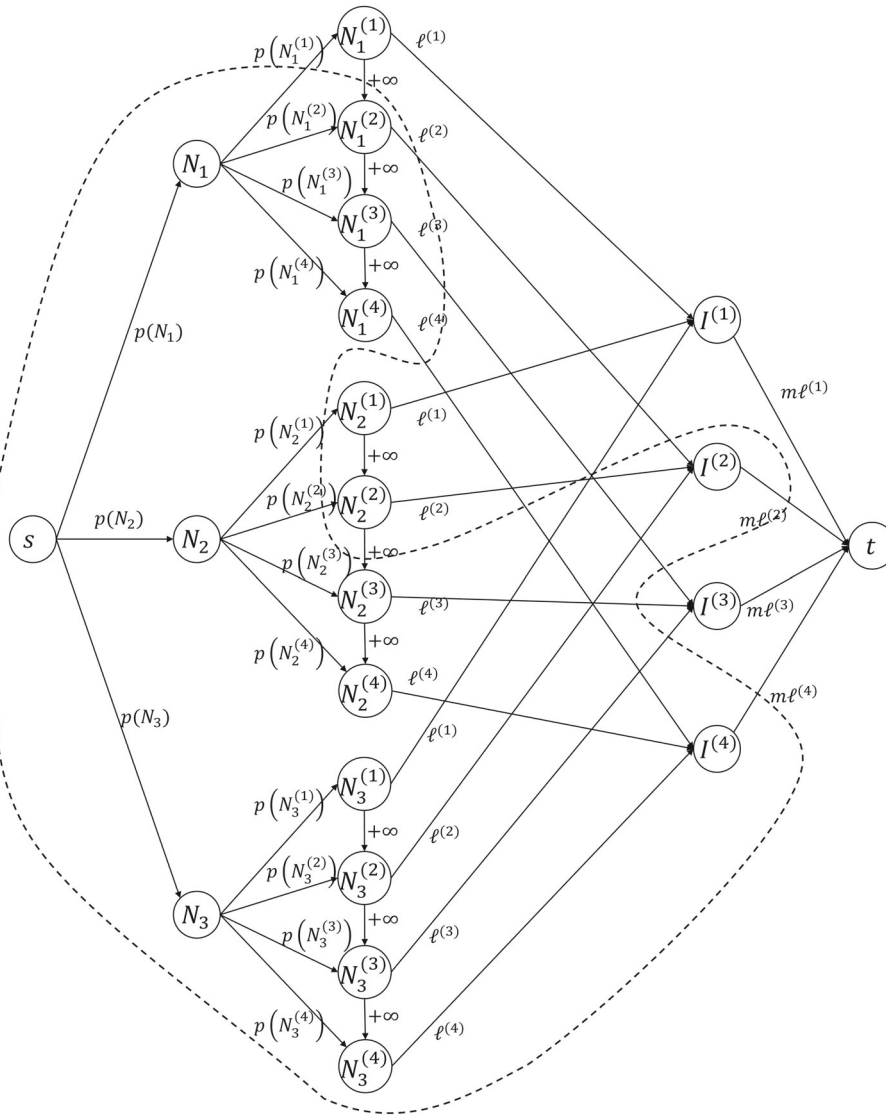


FIGURE 8 A minimum $s-t$ cut $(S^*, V \setminus S^*)$ in Case 2.

a contradiction to the minimality of the $s-t$ cut $(S^*, V \setminus S^*)$. Hence, for each z , $1 \leq z \leq h$, if $I^{(z)} \in S^*$ then $\eta_z \geq m$ holds. This observation, together with the assumption $S^* \cap \{I^{(1)}, \dots, I^{(h)}\} \neq \emptyset$, implies that $z' \leq h$, that is, there exists some z with $\eta_z \geq m$.

We then suppose that $I^{(z)} \notin S^*$ for some z with $z' \leq z \leq h$. Since $\eta_z \geq m$, it follows that

$$\kappa(S^* \cup \{I^{(z)}\}) - \kappa(S^*) = m\ell_z - \eta_z \ell_z \leq 0,$$

which implies that $(S^* \cup \{I^{(z)}\}, V \setminus (S^* \cup \{I^{(z)}\}))$ is also a minimum $s-t$ cut. Hence, we may assume that

$$\{I^{(z')}, I^{(z'+1)}, \dots, I^{(h)}\} \subseteq S^*. \tag{23}$$

We then assume that $N_k^{(z)} \notin S^*$ for some k , $1 \leq k \leq q$, and z is the largest index in $\{z', z'+1, \dots, h\}$. Since $N_k \in S^*$ by the assumption (22), it follows that

$$\kappa(S^* \cup \{N_k^{(z)}\}) - \kappa(S^*) = -p(N_k^{(z)}) \leq 0,$$

which implies that $(S^* \cup \{N_k^{(z)}\}, V \setminus (S^* \cup \{N_k^{(z)}\}))$ is also a minimum $s-t$ cut. Hence, we may assume that

$$\bigcup_{k=1}^q \{N_k^{(z')}, N_k^{(z'+1)}, \dots, N_k^{(h)}\} \subseteq S^*. \tag{24}$$

This assumption implies, in particular, that $z_k \leq z'$ by the assumption (21).

From (21), (22), (23), and (24), we deduce that $S^* = S^{(z)}(z_1, \dots, z_q)$. Hence, the desired inequality (20) follows from (16). This concludes the proof of the lemma. ■

Lemma 9. *In Theorem 8, statement (d) implies statement (a).*

Proof. We know that the capacity of any cut in network G_{Res} is no less than $p(N)$. It is clear that $S^* = \{s\}$ defines a minimum s - t cut with $\kappa(S^*) = p(N)$, that is, there exists a maximum flow f of value $p(N)$. For k , $1 \leq k \leq q$, let the flow through the arc $(N_k^{(z)}, I^{(z)})$ be equal to $p_k^{(z)}$. In terms of the composite job V_k , $p_k^{(z)}$ is the duration of processing of that job in the interval $[R^{(z)}, R^{(z+1)}]$.

The actual feasible schedule S_0 can be found by solving a sequence of auxiliary problems $P^{(z)}$, $1 \leq z \leq h$. Each problem $P^{(z)}$ is in fact problem $P|pmtn, C(j) \leq D|$ — of processing composite jobs V_k with $p_k^{(z)} > 0$ in the interval $[R^{(z)}, R^{(z+1)}]$ and can be solved by McNaughton's algorithm outlined in Section 2. For each interval $[R^{(z)}, R^{(z+1)}]$ the composite job V_k (or its segments) is replaced by the original jobs (or their segments) J_j , $j \in N_k$, with $r(j) \geq R^{(z)}$.

Similarly to the proof of Lemma 4, it can be verified that in schedule S_0 each original job will be processed in full. Additionally, for each interval $[R^{(z)}, R^{(z+1)}]$ the total duration of the jobs that require the same resource does not exceed its length $\ell^{(z)}$, and total processing amount assigned to the interval does not exceed its total processing capacity $m\ell^{(z)}$. Thus, schedule S_0 is feasible. ■

To estimate the running time needed for finding a feasible schedule based on Theorem 8, notice that there are $2 + q + qh + h$ nodes in $G_{\text{Res}} = (V, A)$. There are q arcs leaving the source, at most three distinctive arcs coming to and leaving each node $N_k^{(z)}$, $1 \leq k \leq q$, $1 \leq z \leq h$, and h arcs coming to the sink, that is, at most $q + 3qh + h$ arcs all together. Thus, $|A| = O(|V|)$ and using the algorithm in [19] the maximum flow in G_{Res} can be found in $O\left(\frac{|V|^2}{\log |V|}\right) = O\left(\frac{q^2 h^2}{\log(qh)}\right) = O\left(\frac{q^2 n^2}{\log(qn)}\right)$ time.

6 | CONCLUSIONS

The necessary and sufficient conditions for the existence of a feasible solution for problem $P|r(j), pmtn, C(j) \leq D|$ — have been known and widely used for 50 years. One of these conditions is in the form of a system of inequalities, while the other is related to finding the maximum flow in a special network. In this paper, we have strengthened the understanding of the connection between these two conditions by establishing their link to the minimum cuts in such a network. The elaborated proof technique is essentially based on the underlying network model and on the maximum-flow minimum-cut theorem.

To illustrate further applications of the proof technique, we study the enhanced version of the problem, in which jobs require additional resources, namely problem $P|r(j), pmtn, \text{res} \cdot 111, C(j) \leq D|$ —. For that problem we develop the network model, formulate the feasibility conditions, and present a formal proof of their correctness. The conditions can be used not only for verifying feasibility, but also for optimization versions of the problem, for example, for minimizing the makespan if a common deadline D is treated as a parameter, or for minimizing the total compression cost if job processing times are controllable in the sense of [23].

Another generalization of the problem under study, problem $Q|r(j), pmtn, C(j) \leq D|$ — with parallel machines having different speeds, can be a subject of future research. While the correctness of the associated network is proved in [6] and feasibility can be verified via solving a network flow problem, it seems interesting to derive conditions in an analytical form, with a polynomial number of inequalities. Notice that the makespan minimization problem $Q|r(j), pmtn|C_{\text{max}}$ is studied in [16]. Although the necessary and sufficient optimality conditions for problem $Q|r(j), pmtn, C(j) \leq D|$ — are not derived in [16], the algorithm for the makespan minimization and its justification are essentially based on an analogue of the parameter denoted in our paper by $g^{(z)}(j)$.

Overall, network models play an important role in scheduling research, and there is a potential for future application of our methodology. Possible problems to be addressed are, for example, parallel machine problems with job splitting [22], parallel machine problems with additional restrictions on machines' availability ([1] and subsequent papers), machines' availability and eligibility [17], open-shop scheduling with preemption [21], and just-in-time scheduling [9], to name a few.

ACKNOWLEDGMENTS

This work is supported by the project EP/T01461X/1 “Algorithmic Support for Massive Scale Distributed Systems” funded by the Engineering and Physical Sciences Research Council (EPSRC).

DATA AVAILABILITY STATEMENT

Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

ORCID

Akiyoshi Shioura  <https://orcid.org/0000-0002-5216-7190>

Vitaly A. Strusevich  <https://orcid.org/0000-0002-4602-8573>

Natalia V. Shakhlevich  <https://orcid.org/0000-0002-5225-4008>

REFERENCES

- [1] J. Błażewicz, M. Drozdowski, P. Formanowicz, W. Kubiak, and G. Schmidt, *Scheduling preemptable tasks on parallel processors with limited availability*, *Parallel Comput.* **26** (2000), 1195–1211.
- [2] J. Błażewicz, J. K. Lenstra, and A. H. G. Rinnooy Kan, *Scheduling subject to resource constraints: classification and complexity*, *Discr. Appl. Math.* **5** (1983), 11–24.
- [3] J. Błażewicz, K. H. Ecker, E. Pesch, G. Schmidt, M. Sterna, and J. Weglarz, “*Scheduling under resource constraints*,” *Handbook of scheduling. From theory to practice*, Springer, Berlin, 2019, pp. 475–525.
- [4] B. Chen, “*Parallel machine scheduling for early completion*,” *Handbook of scheduling: Algorithms, models and performance analysis*, J. Y.-T. Leung (ed.), Chapman & Hall/CRC, Boca Raton, 2004, pp. 9–175–9–184.
- [5] E. B. Edis, C. Oguz, and I. Ozkarahan, *Parallel machine scheduling with additional resources: Notation, classification, models and solution methods*, *Eur. J. Oper. Res.* **230** (2013), 449–463.
- [6] A. Federgruen and H. Groenevelt, *Preemptive scheduling of uniform machines by ordinary network flow techniques*, *Manage. Sci.* **32** (1986), 341–349.
- [7] V. S. Gordon and V. S. Tanaev, “*Deadlines in single-stage deterministic scheduling*,” *Optimization of Systems for Collecting, Transfer and Processing of Analogous and Discrete Data in Local Information Computing Systems, Materials of the 1st Joint Soviet-Bulgarian seminar*, Institute of Engineering Cybernetics of Academy of Sciences of BSSR – Institute of Engineering Cybernetics of Bulgarian Academy of Sciences, Minsk, 1973, pp. 53–58 (in Russian).
- [8] E. Hebrard, M.-J. Hugueta, N. Jozefowicz, A. Maillard, C. Pralet, and G. Verfaillie, *Approximation of the parallel machine scheduling problem with additional unit resources*, *Discr. Appl. Math.* **215** (2016), 126–135.
- [9] K. Hiraishi, E. Levner, and M. Vlach, *Scheduling of parallel identical machines to maximize the weighted number of just-in-time jobs*, *Comput. Oper. Res.* **29** (2002), 841–848.
- [10] W. Horn, *Some simple scheduling algorithms*, *Naval Res. Logist. Quart.* **21** (1974), 177–185.
- [11] K. Jansen and L. Porkolab, *Preemptive scheduling with dedicated processors: applications of fractional graph coloring*, *J. Sched.* **7** (2004), 35–48.
- [12] K. Jansen and L. Porkolab, *General multiprocessor task scheduling: Approximate solutions in linear time*, *SIAM J. Comp.* **35** (2005), 519–530.
- [13] K. Jansen and L. Porkolab, *On preemptive resource constrained scheduling: Polynomial-time approximation schemes*, *SIAM J. Discr. Math.* **20** (2006), 545–563.
- [14] T. Janssen, C. Swennenhuis, A. Bitar, T. Bosman, D. Gijswijt, L. van Iersel, S. Dausère-Pérèz, and C. Yugma, *Parallel machine scheduling with a single resource per job*. arXiv preprint, arXiv:1809.05009v3 2018.
- [15] A. V. Karzanov, *Determining the maximal flow in a network by the method of preflows*, *Sov. Math. Doklady* **15** (1974), 434–437.
- [16] J. Labetoulle, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan, “*Preemptive scheduling of uniform machines subject to release dates*,” *Progress in combinatorial optimization*, W. R. Pulleyblank (ed.), Academic Press, Cambridge MA, 1984, pp. 245–261.
- [17] L.-W. Liao and G.-J. Sheen, *Parallel machine scheduling with machine availability and eligibility constraints*, *Eur. J. Oper. Res.* **184** (2008), 458–467.
- [18] R. McNaughton, *Scheduling with deadlines and loss functions*, *Manag. Sci.* **12** (1959), 1–12.
- [19] J. B. Orlin, *Max flows in $O(nm)$ time or better*, *STOC '13, Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, ACM, New York, 2013, pp. 765–774.
- [20] S. Sahni, *Preemptive scheduling with due dates*, *Oper. Res.* **27** (1979), 925–934.
- [21] A. Sedeño-Noda, D. Alcaide, and C. González-Martín, *Network flow approaches to pre-emptive open-shop scheduling problems with time-windows*, *Eur. J. Oper. Res.* **174** (2006), 1501–1518.
- [22] P. Serafini, *Scheduling jobs on several machines with the job splitting property*, *Oper. Res.* **44** (1996), 617–628.
- [23] A. Shioura, N. V. Shakhlevich, and V. A. Strusevich, *Preemptive models of scheduling with controllable processing times and of scheduling with imprecise computation: A review of solution approaches*, *Eur. J. Oper. Res.* **266** (2018), 795–818.
- [24] V. A. Strusevich, *Approximation algorithms for makespan minimization on identical parallel machines under resource constraints*, *J. Oper. Res. Soc.* **72** (2021), 2135–2146.

How to cite this article: A. Shioura, V. A. Strusevich, and N. V. Shakhlevich, *Generalizing Horn’s conditions for preemptive scheduling on identical parallel machines via network flow techniques*, *Networks*. **83** (2024), 527–546. <https://doi.org/10.1002/net.22202>