IS A UNIT-TIME JOB SHOP NOT EASIER THAN IDENTICAL PARALLEL MACHINES?†

VADIM G. TIMKOVSKY

Star Data Systems Inc., 900-1 Toronto St., Toronto,
Ontario M5C 2V6, E-mail: vtimkovs@stardata.ca and
Department of Computer Science and Systems, McMaster University
Hamilton, Ontario L8S 4K1, E-mail: timko@dcss.mcmaster.ca

October 31, 1997

ABSTRACT. This paper gives the positive answer to the question posed in the title for a wide class of minimization criteria including the maximum completion time, maximum lateness, total completion time, total weighted completion time, total tardiness, total weighted tardiness, number of late jobs and the weighted number of late jobs. That is any scheduling problem for m identical parallel machines to minimize a criterion of the class reduces to a scheduling problem for an m-machine unit-time job shop to minimize the same criterion. Employing this general reduction we prove the NP-hardness of unit-time job-shop scheduling problems which had unknown complexity status before. The paper also presents a comprehensive picture of complexity results attained in unit-time job-shop scheduling and related open problems.

Key words and phrases. Job shop, identical parallel machines, scheduling, unit-time operations, polynomial-time reduction, NP-hardness.

†This work was partially supported by the Grant from the American Mathematical Society's (fSU) Aid Fund and by the Grant from the Soros Foundation.

INTRODUCTION

Job-shop scheduling and *identical-parallel-machines* scheduling to minimize a criterion are problems with different types of the machine environment. In the former problem, each job consists of a chain of operations which have to be processed on specified machines. In the latter problem, each job consists of a single operation which can be processed on any machine. Evidently, the former problem is harder, and even with *unit-processing-time operations* it appears to have at least the same complexity as the latter problem with *arbitrary job processing times* [T93].

This paper makes an attempt to confirm this observation theoretically by describing for a wide class of minimization criteria a reduction from any m-identical-parallel-machines problem to an m-machine unit-time job-shop problem.

It is interesting to note that in earlier works there have been described inverse reductions in the two-machine case. Probably, Kubiak [K89] was the first to find out such a

Typeset by $\mathcal{AMS}$-TEX

reduction. He proposed a pseudopolynomial-time algorithm finding a *no-wait* schedule to minimize the maximum completion time in a two-machine unit-time job shop employing a pseudopolynomial-time algorithm finding a *nonpreemptive* schedule to minimize the same criterion on two identical machines with arbitrary job processing times. As it was shown by Kubiak, Sethi and Sriskandarajah [KSS95] the same approach works out in developing a polynomial-time algorithm for the relaxed version of the former problem without the no-wait constraint employing a polynomial-time algorithm for the relaxed version of the latter problem with preemptions. In other words, these works describe polynomial-time reductions:

$$J2|no\,wait, p_{ij}=1|C_{\max} \quad \propto \quad P2||C_{\max}$$
$$J2|p_{ij}=1|C_{\max} \quad \propto \quad P2|pmtn|C_{\max}$$

In what follow, we use the well-known three-field classification $\alpha|\beta|\gamma$ for scheduling problems described in the surveys [GLLRK79], [LLRK82] and [LLRKS93], where α, β and γ specify the *machine environment*, *job characteristics* and the *minimization criterion*, respectively.

The above two reductions allow to suggest that under the unit-processing-time constraint an m-machine job shop might even be easier than m identical parallel machines in a similar correspondence. This paper shows that this is very unlikely because it describes the inverse polynomial-time reductions taking place in a more general situation involving two, three or more machines, precedence relations or release dates of jobs and the following minimization criteria:

$C_{\max}$, *the maximum completion time*;
$L_{\max}$, *the maximum lateness*;
ΣC_j, *the total completion time*;
$\Sigma w_j C_j$, *the total weighted completion time*;
ΣT_j, *the total tardiness*;
$\Sigma w_j T_j$, *the total weighted tardiness*;
ΣU_j, *the number of late jobs*;
$\Sigma w_j U_j$, *the weighted number of late jobs.*

That is "preemptive/nonpreemptive" problems for identical parallel machines reduce to "wait/no-wait" problems for a unit-time job shop. The reductions remain invariable the number of jobs, the number of machines, the precedence relation and the minimization criterion. Employing known NP-hard identical-parallel-machines problems they provide a general NP-hardness proof for many unit-time job-shop problems inculing those which had unknown complexity status before. In a particular case, they prove a polynomial-time equivalence between the problems in the above two reductions of Kubiak, Sethi and Sriskandarajah.

Basic techniques applied here are: stretching job processing times such that they become divisible by the number of machines; schedule and instance transformations by relatively small shifting jobs and their due dates in time; decomposing one-operation jobs into unit-time operations; and distributing them among the machines by a periodic way.

The structure of the paper is as follows. Section 1 represents a consolidated picture of complexity results in unit-time job-shop scheduling attained earlier and in this paper. Section 2 considers stretching and shifting scheduling techniques. Section 3 considers a periodic unit-time job-shop and represents the main result. Possible generalizations and related open problems are discussed in conclusion.

1. Complexity results in unit-time job-shop scheduling

Tables 1, 2 and 3 contain only maximal problems known to be solvable in polynomial time, minimal problems known to be NP-hard, minimal problems known to be strongly NP-hard and related references. The star $\star$ denotes the reference to this paper. The "$\bar{n}$" in the job characteristics field denotes a fixed upper bound for the number n of jobs.

The complexity status of $J2|r_j, p_{ij} = 1|L_{\max}$ is unclear. Pseudopolynomial-time algorithms are known only for $J2|p_{ij} = 1|\Sigma w_j U_j$ [K96A] and $J2|no\,wait, p_{ij} = 1|C_{\max}$ [K89]. Hence, only these two problems are proved to be ordinarily NP-hard. The other NP-hard problems in Table 2 as well as the NP-hard $J2|no\,wait, r_j, p_{ij} = 1|C_{\max}$, which is equivalent to $J2|no\,wait, p_{ij} = 1|L_{\max}$ by symmetry, and $J2|no\,wait, p_{ij} = 1|\Sigma T_j$ are open for the ordinary or strong NP-hardness.

The letter C in the machine environment field denotes a *cycle shop*, a special case of a job shop, where all the jobs have the same route passing through the machines like in a *flow shop* but repetitions of machines in the route are allowed. If the machines in a cycle shop have different speeds, then s_{ν_i} denotes the speed of the ν_ith machine which has to process the ith operation in each job.

The complexity status of unit-time cycle-shop problems except well-solvable $C|p_{ij} = 1|C_{\max}$ [T85] and $C2|p_{ij} = 1/s_{\nu_i}|C_{\max}$ [T85] is unknown. A related result is a polynomial-time algorithm for $C|p_{ij} = 1/s_{\nu_i}|C_{\max}$ finding a schedule with the performance ratio $1 + (3m + h)/n$ [T86], where h is the number of operations in each job. Besides, $C2|p_{ij} \in \{1, 2\}|C_{\max}$, a restricted version of the strongly NP-hard $J2|p_{ij} \in \{1, 2\}|C_{\max}$ [LRK79] is also so [T81].

Table 1. Problems solvable in polynomial time

$C\|p_{ij} = 1\|C_{\max}$	[T85]
$C2\|p_{ij} = 1/s_{\nu_i}\|C_{\max}$	[T85]
$J2\|r_j, p_{ij} = 1\|C_{\max}$	[T93],[T97]
$J2\|p_{ij} = 1\|\Sigma C_j$	[KT96]
$J2\|p_{ij} = 1\|\Sigma U_j$	[K96A]
$J2\|no\,wait, p_{ij} = 1\|\Sigma C_j$	[K96B]
$J\|prec, r_j, n \le \bar{n}, p_{ij} = 1\|\Sigma w_j T_j$	[BK96]
$J\|prec, r_j, n \le \bar{n}, p_{ij} = 1\|\Sigma w_j U_j$	[BK96]

Table 2. NP-hard problems

$J2\|r_j, p_{ij} = 1\|\Sigma C_j$	[⋆]
$J2\|r_j, p_{ij} = 1\|\Sigma U_j$	[⋆]
$J2\|p_{ij} = 1\|\Sigma w_j C_j$	[⋆]
$J2\|p_{ij} = 1\|\Sigma T_j$	[⋆]
$J2\|p_{ij} = 1\|\Sigma w_j U_j$	[K96A],[⋆]
$J2\|no\,wait, p_{ij} = 1\|C_{\max}$	[T85],[SL86],[K89],[⋆]
$J2\|no\,wait, p_{ij} = 1\|\Sigma w_j C_j$	[⋆]

Table 3. Strongly NP-hard problems

$J2\|chains, m_j = 1, p_{ij} = 1\|C_{\max}$	[T85],[TSS94]
$J2\|chains, p_{ij} = 1\|\Sigma C_j$	[⋆]
$J2\|r_j, p_{ij} = 1\|\Sigma w_j C_j$	[⋆]
$J2\|p_{ij} = 1\|\Sigma w_j T_j$	[⋆]
$J2\|no\,wait, chains, p_{ij} = 1\|C_{\max}$	[⋆]
$J2\|no\,wait, chains, p_{ij} = 1\|\Sigma C_j$	[⋆]
$J2\|no\,wait, r_j, p_{ij} = 1\|\Sigma C_j$	[⋆]
$J2\|no\,wait, r_j, p_{ij} = 1\|L_{\max}$	[⋆]
$J2\|no\,wait, p_{ij} = 1\|\Sigma w_j T_j$	[⋆]
$J3\|p_{ij} = 1\|C_{\max}$	[LRK79]
$J3\|p_{ij} = 1\|\Sigma C_j$	[L]
$J3\|no\,wait, p_{ij} = 1\|C_{\max}$	[SL86]
$J3\|no\,wait, p_{ij} = 1\|\Sigma C_j$	[SL86]

Note that Kravchenko's algorithm [K96A] for $J2|p_{ij} = 1|\Sigma U_j$ uses as a subroutine an algorithm for $J2|p_{ij} = 1|L_{\max}$ which is equivalent to $J2|r_j, p_{ij} = 1|C_{\max}$ by symmetry.

Among job-shop problems with bounded number of jobs $J2|n \le \bar{n}|\Sigma w_j T_j$ [BKS95], $J2|n \le \bar{n}|\Sigma w_j U_j$ [BKS95], $J|prec, r_j, n \le 2|\Sigma w_j T_j$ [S91] and $J|prec, r_j, n \le 2|\Sigma w_j U_j$ [S91] are solvable in polynomial time. Therefore, their unit-time restricted versions are also so.

Other complexity results on job-shop scheduling with arbitrary processing time operations that are not related to unit-time job-shop scheduling can be found in [LLRKS93] and in the web page *http://www.mathematik.uni-osnabrueck.de/research/OR/class/*.

2. Preliminaries

2.1. Denotations and assumptions. All denotations related to scheduling will follow the notation from [LLRKS93]: n, the *number of jobs*; m, the *number of machines*; O_{ij}, the ith *operation* in the job J_j which has to be processed by the *machine* $M_{\mu_{ij}}$; p_{ij}, the *processing time* of O_{ij}; C_j, m_j, r_j, d_j and w_j, the *completion time*, *number of operations*, *release date*, *due date* and the *weight* of J_j, respectively; $\beta_1, ..., \beta_7$ describe the *preemption, no-wait, precedence-relation, release-date, jobs-number, operations-number, processing-time* constraints, respectively; $\beta_u = \circ$, where $\circ$ is the empty symbol, means that the constraint β_u is removed; β_{u-v} denotes the constraint sequence $\beta_u, \beta_{u+1}, ..., \beta_v$, where $u, v \in \{1, ..., 7\}$. Parameters p_{ij}, r_j, d_j and w_j are considered to be integer. The index i will be deleted in donotations of one-operation job processing times.

The pair (I, k) will denote the instance of the *decision problem* related to an instance I of the scheduling problem $\alpha|\beta|\gamma$ with upper bound k for γ values [GJ79]. We consider only decision problems belonging to NP [GJ79], so we will assume that for any instance (I, k) there exists a feasible schedule having a code of length l polynomial in size of (I, k). Thus, all number parameters specifying the schedule can be represented by fractions with denominators at most l and, therefore:

(a) any positive difference between the parameters is at least $1/l^2$; and
(b) all the parameters become integer after increasing by the common multiplier $l!$.

To prove a polynomial-time reduction $\alpha|\beta|\gamma \propto \alpha'|\beta'|\gamma'$ we will describe an instance I' of $\alpha'|\beta'|\gamma'$ and upper bound k' that can be constructed in time polynomial in size of (I, k) and show that the answers for (I, k) and (I', k') are the same [GJ79].

2.2. Schedule and instance transformations. Let I be an instance of $\alpha|\beta|\gamma$, and σ be a feasible schedule for I. Without loss of generality assume that $d_j = 0$ or $w_j = 1$ for $j = 1, 2, ..., n$ if γ does not include due dates or γ is not a weighted criterion, respectively. Set $w = w_1 + w_2 + ... + w_n$. For integers $a > 0$ and $b \geq 0$ construct the instance $aI + b$ from I by replacing p_{ij}, r_j, d_j by $ap_{ij}, ar_j, ad_j + b$ for all $j = 1, 2, ..., n$ and the schedule $a\sigma$ by stretching σ by a times. Obviously, $a\sigma$ is a feasible schedule for aI and all completion times C_j in σ become aC_j in $a\sigma$.

For integers $b_j \geq 0$ define $a\sigma + b$ to be a feasible schedule for aI with completion times $aC_j + b_j$, where $b_j \leq b$ for all $j = 1, 2, ..., n$. Obviously, $a\sigma + b$ is a feasible schedule for $aI + b$. Let $[\sigma, I]$ denotes the γ value of σ for I. Then it is easy to make sure that for $\gamma \in \{C_{\max}, L_{\max}, \Sigma C_j, \Sigma w_j C_j, \Sigma T_j, \Sigma w_j T_j\}$

$$a[\sigma, I] = [a\sigma, aI] \leq [a\sigma + b, aI] \leq [a\sigma, aI] + bw. \tag{2.1}$$

Now let $b > 0$ and $a/b > l^2$. Then from the property (a) we have: $C_j - d_j > 0 \Leftrightarrow C_j - d_j \geq 1/l^2 \Leftrightarrow a(C_j - d_j) \geq a/l^2 \Leftrightarrow aC_j - (ad_j + b) > 0 \Leftrightarrow (aC_j + b_j) - (ad_j + b) > 0$. It means that the set of late jobs in σ for I, in $a\sigma$ for aI and in $a\sigma + b$ for $aI + b$ is the same. Therefore, for $\gamma \in \{\Sigma U_j, \Sigma w_j U_j\}$

$$[\sigma, I] = [a\sigma, aI] = [a\sigma + b, aI + b]. \tag{2.2}$$

2.3. Schedules divisible by m. We call a schedule *divisible by m* if the length of every nonpreemptive part of every operation in it is divisible by m. A scheduling problem of finding a schedule divisible by m will be denoted by job characteristics $\beta_1 \in \{m\, pmtn, \circ\}$ and $\beta_6 = p_{ij} = mg_{ij}$, where g_{ij} are integers. Obviously, the number of all nonpreemptive parts in any schedule divisible by m does not exceed $g = \sum_{j=1}^{n} \sum_{i=1}^{m_j} g_{ij}$.

The following lemma shows that any scheduling problem in the three-field classification reduces to a problem of finding a schedule divisible by m.

Lemma 2.1.

$$\begin{aligned} \alpha|pmtn, \beta_{2-7}|\gamma &\quad\propto\quad \alpha|m\, pmtn, \beta_{2-6}, p_{ij} = mg_{ij}|\gamma\,, \\ \alpha|\beta_{2-7}|\gamma &\quad\propto\quad \alpha|\beta_{2-6}, p_{ij} = mg_{ij}|\gamma\,. \end{aligned}$$

Proof. To prove the first reduction take $a = ml!$ and set $I' = aI$,

$$k' = \begin{cases} ak & \text{if } \gamma \in \{C_{\max}, L_{\max}, \Sigma C_j, \Sigma w_j C_j, \Sigma T_j, \Sigma w_j T_j\} \\ k & \text{if } \gamma \in \{\Sigma U_j, \Sigma w_j U_j\}. \end{cases}$$

By the property (b) all the lengths of nonpreemptive parts of operations in σ become integer and divisible by m in $a\sigma$. Since the left equalities in (2.1) and (2.2) hold, the answer for (I, k) is the same as for (I', k'). Besides, $\log a$ is bounded by a polynomial in l, so (I', k') can be constructed in polynomial time. For the second reduction it is sufficient to take $a = m$. □

2.4. Schedules modulo m. We call a schedule *modulo m* if start times of nonpreemptive parts of operations scheduled on M_i are $(i-1)(\mathrm{mod}\, m)$ for $i = 1, 2, ..., m$. Note that adding this requirement to $\alpha|\beta|\gamma$ can only increase the minimum γ value. A scheduling problem of finding a schedule modulo m will be denoted by $\alpha(\mathrm{mod}\, m)|\beta|\gamma$. Let I be an instance of $\alpha|\beta|\gamma$ and σ be a schedule for I. The following *shift procedure* transforms σ into a feasible schedule modulo m for I that is denoted by $\sigma(\mathrm{mod}\, m)$.

Define $\mathcal{P}$ to be the set of all nonpreemptive parts of operations of jobs in σ. Introduce a precedence relation $\prec$ on $\mathcal{P}$ as follows. Set $P \prec Q$ if and only if: P and Q belong to J_j and J_k, respectively, such that $J_j \to J_k$ if J_j and J_k are in a precedence relation $\to$; or P and Q are parts of O_{ij} and O_{lj}, respectively, where $i < l$; or P and Q are parts of one operation and Q must be processed after P; or P and Q are scheduled in σ on one machine, which processes Q after P.

Let $\mathit{Min}\, \mathcal{P}$ be the set of minimal parts of $\mathcal{P}$ in accordence with $\prec$, $\mathit{Rest}\, \mathcal{P} = \mathcal{P} \setminus \mathit{Min}\, \mathcal{P}$, $\mathit{Near}_i\, t = \min\{\, u : u \ge t \;\;\&\;\; u = (i-1)(\mathrm{mod}\, m)\,\}$, and $\mathit{Machine}\, P$ be the index of the machine on which P is scheduled in σ. Denote by s_P and t_P the start times of P in σ and $\sigma(\mathrm{mod}\, m)$, respectively. Informally, the shift procedure moves the parts to the right so that they attain earliest possible start times $(i-1)(\mathrm{mod}\, m)$ without violation of $\prec$.

SHIFT PROCEDURE

```
while 𝒫 ≠ ∅ do
    for P ∈ Min 𝒫 do
        i ← Machine P
        t_P ← Near_i s_P
    d ← max { t_P − s_P : P ∈ Min 𝒫 }
    for P ∈ Rest 𝒫 do
        s_P ← s_P + d
    𝒫 ← Rest 𝒫
```

Since $d < m$, the procedure increases the completion time of every job by at most $m|\mathcal{P}| \leq mg$, therefore

$$(a\sigma)(\mathrm{mod}\, m) \text{ is an example of } a\sigma + mg \tag{2.3}$$

for any positive integer a. The following lemma shows that any problem of finding schedules divisible by m reduces to a problem of finding schedules modulo m and divisible by m as well.

Lemma 2.2.

$$\alpha|m\,pmtn, \beta_{3-6}, p_{ij} = mg_{ij}|\gamma \quad \propto \quad \alpha(\mathrm{mod}\, m)|m\,pmtn, \beta_{3-6}, p_{ij} = mg'_{ij}|\gamma\,,$$
$$\alpha|\beta_{3-6}, p_{ij} = mg_{ij}|\gamma \quad \propto \quad \alpha(\mathrm{mod}\, m)|\beta_{3-6}, p_{ij} = mg'_{ij}|\gamma\,.$$

Proof A: the case $\gamma \in \{C_{\max}, L_{\max}, \Sigma C_j, \Sigma w_j C_j, \Sigma T_j, \Sigma w_j T_j\}$. Take $a = l^2(gmw + 1)$, $b = mg$ and set $I' = aI$, $k' = ak + bw$. If σ is a feasible schedule for I, then $(a\sigma)(\mathrm{mod}\, m)$ is a feasible schedule for I'. Applying (2.1) and (2.3) we have:

$$[\sigma, I] \leq k \;\Rightarrow\; [(a\sigma)(\mathrm{mod}\, m), aI] = [a\sigma + b, aI] \leq a[\sigma, I] + bw \leq ak + bw = k'$$

On the other hand, let $[\sigma, I] > k$ for any feasible schedule σ for I. From the equality in (2.1) and the property (a) we have $[a\sigma, aI] = a[\sigma, I] \geq ak + a/l^2 = ak + bw + 1$ for any feasible schedule $a\sigma$ for aI. Applying (2.1) and (2.3) again we have for any $(a\sigma)(\mathrm{mod}\, m)$ for I':

$$[\sigma, I] > k \;\Rightarrow\; [(a\sigma)(\mathrm{mod}\, m), aI] = [a\sigma + b, aI] \geq a[\sigma, I] \geq ak + bw + 1 > k'. \square$$

Proof B: the case $\gamma \in \{\Sigma U_j, \Sigma w_j U_j\}$. Take the same a and b as in the proof A. Note that $a/b > l^2$ and set $I' = aI + bw$, $k' = k$. The equalities (2.2) and (2.3) show that the answers for (I, k) and (I', k') are the same. $\square$

Informally, Lemmas 2.1 and 2.2 say that all the criteria listed in the introduction remain insensitive to relatively small time shifts of operations when big stretching schedules out in time. From the other hand, the stretching can be small enough to save a polynomial length of schedule codes.

3. Main result

3.1. A periodic unit-time job shop. A unit-time job-shop problem we call *periodic* if $m_j = mh_j$, $m > 1$ and

(i) $\mu_{ij} = (i-1)(\operatorname{mod} m) + 1$ for all $i = 1, ..., m$, $j = 1, ..., n$, and

(ii) the lengths of uninterrupted parts of jobs in the schedule sought for are divisible by m.

This problem will be denoted by $J\alpha_2(\text{per})|\beta_{2-5}, m_j = mh_j, p_{ij} = 1|\gamma$. Thus, every uninterrupted part has the first operation on M_1 and the last operation on M_m. To determine a schedule for this problem it is sufficient to indicate just start times and lengths of uninterrupted parts of jobs because the unit-time operations inside the parts are periodically distributed among the machines.

Lemma 3.1. *Let I be an instance of $J\alpha_2(\text{per})|\beta_{2-5}, m_j = mh_j, p_{ij} = 1|\gamma$ and a schedule σ for I meets requirements β_{2-5} and (ii). Then σ is feasible for I if and only if the overlap of time intervals of two uninterrupted parts with start times s and t implies $s \neq t(\operatorname{mod} m)$.*

Proof. Let $O_{pr}, ..., O_{p+q,r}, ...$ and $O_{xz}, ..., O_{x+y,z}, ...$ be uninterrupted parts overlapping each other, s and t be start times of O_{pr} and O_{xz}, respectively, $O_{p+q,r}$ and $O_{x+y,z}$ both be on the same machine. Then from (i) and (ii) we have $\mu_{pr} = \mu_{xz}$, i.e., $p = x(\operatorname{mod} m)$ and $\mu_{p+q,r} = \mu_{x+y,z}$, i.e., $p + q = (x + y)(\operatorname{mod} m)$. Therefore, $O_{p+q,r}$ and $O_{x+y,z}$ have the same start time $u \Leftrightarrow q = u - s \;\&\; y = u - t \Leftrightarrow p + u - s = (x + u - t)(\operatorname{mod} m) \Leftrightarrow p - s = (x - t)(\operatorname{mod} m) \Leftrightarrow s = t\,(\operatorname{mod} m)$. □

Lemma 3.2.

$$J\alpha_2(\text{per})|\beta_{2-5}, m_j = mh_j, p_{ij} = 1|\gamma \quad \propto \quad J\alpha_2|\beta_{2-5}, p_{ij} = 1|\gamma\,.$$

Proof. Set $I' = I$, $k' = k$. When $\beta_2 = \textit{no wait}$, the reduction is obvious. When $\beta_2 = \circ$, it is sufficient to show that any feasible schedule σ' for I' can be transformed into a feasible schedule σ for I, i.e. observing (ii), without increasing the completion times. The transformation is as follows.

Looking through σ' from left to right find the first uninterrupted part of length not divisible by m. Since it is first, it should be the chain of operations $O_{i+1,j}, ..., O_{i+k,j}$, where $\mu_{i+1,j} = 1, ..., \mu_{i+k,j} = k < m$. If $O_{i+k,j}$ occupies time unit $t-1$, then time unit t on M_{k+1} is idle, and $O_{i+k+1,j}$ occupies on M_{k+1} time unit greater than t. Let us move $O_{i+k+1,j}$ to the left in time unit t. Since this operation is inside a job, the move does not violate the precedence relation or release dates, and the completion times are not being increased. Obviously, repeating this procedure gives the necessary transformation. □

3.2. For any criteria in the three-field classification a unit-time job shop is not easier than identical parallel machines. The following lemma reveals a key connection between unit-time job-shop scheduling and identical-parallel-machines scheduling.

Lemma 3.3. *If* $\alpha_2 \neq 1$ *, then*

$$P\alpha_2(\mathrm{mod}\, m)|m\, pmtn, \beta_{3-5}, p_j = mg_j|\gamma \quad \sim \quad J\alpha_2(\mathrm{per})|\beta_{3-5}, m_j = mh_j, p_{ij} = 1|\gamma\,,$$
$$P\alpha_2(\mathrm{mod}\, m)|\beta_{3-5}, p_j = mg_j|\gamma \quad \sim \quad J\alpha_2(\mathrm{per})|no\, wait,\, \beta_{3-5}, m_j = mh_j, p_{ij} = 1|\gamma\,.$$

Proof. Set $k' = k$, $h_j = g_j$, for all $j = 1, ..., n$, and construct I' replacing every one-operation job J_j in I by multi-operation job J'_j with unit-time operations $O_{1j}, ..., O_{mh_j,j}$. Due to the definition of schedules divisible by m and the requirement (ii) the replacement provides a natural one-to-one correspondence between nonpreemptive parts of J_j and uninterrupted parts of J'_j with the same length. So, the unit-time operations of J'_j can be considered as blocks of a decomposition of J_j.

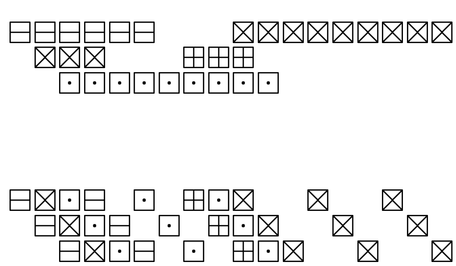
If σ is a feasible schedule for I, then it can be transformed into a feasible schedule σ' for I' by translating each nonpreemptive part of each J_j in σ into the corresponding uninterrupted part of J'_j and assigning to it the same start time. Since start times on different machines in σ are not comparable modulo m, start times of any two overlapping uninterrupted parts are also not comparable modulo m. Therefore, due to Lemma 3.1 σ' is feasible for I'.

Analogously, any feasible schedule σ' for I' can be transformed into a feasible schedule σ for I by translating uninterrupted parts of σ' into the corresponding nonpreemptive parts of σ with assigning for them the same start times t on $M_{t\,(\mathrm{mod}\, m)+1}$. So, start times on M_i are $(i-1)(\mathrm{mod}\, m)$. Since Lemma 3.1 holds, any two overlapping nonpreemptive parts should appear on different machines. Thus, σ is feasible for I.

The transformations of σ into σ' and vice versa remain the completion times of J_j and J'_j the same, therefore, γ values for σ and σ' are the same. □

It is interesting to note that the proof provides not only a polynomial-time equivalence, but even a polynomial-time isomorphism [GJ79] between the problems.

Example 3.1. The following picture demonstrates the transformations of a schedule modulo 3 and divisible by 3 for 3 identical parallel machines into the corresponding periodic 3-machine unit-time job shop schedule and vice versa.



Consecutively applying Lemmas 2.1 and 2.2 for $\alpha = P\alpha_2$, Lemmas 3.3 and 3.2, we obtain a chain of reductions, which proves the main result.

Theorem 3.1. *If* $\alpha_2 \neq 1$ *, then*

$$P\alpha_2|pmtn, \beta_{3-5}|\gamma \quad \propto \quad J\alpha_2|\beta_{3-5}, p_{ij} = 1|\gamma\,,$$
$$P\alpha_2|\beta_{3-5}|\gamma \quad \propto \quad J\alpha_2|no\,wait,\, \beta_{3-5}, p_{ij} = 1|\gamma\,.$$

In the following evident corollary from Theorem 3.1 we take into account the equivalences $P\alpha_2|pmtn|\Sigma w_j C_j \sim P\alpha_2||\Sigma w_j C_j$ [M59], $1|pmtn|\gamma \sim 1||\gamma$ [CMM67] and the trivial reduction $1|\beta|\gamma \propto P2|\beta|\gamma$.

Corollary 3.1. *The reductions listed below prove the NP-hardness or strong NP-hardness of unit-time job-shop scheduling problems listed in the third column. This follows from the fact that the problems listed in the second column are NP-hard or strongly NP-hard as it has been established in the papers, references to which are listed in the first column.*

Table 4. Reductions proving the NP-hardness

[DLY90]	$P2\|pmtn, r_j\|\Sigma C_j$	$\propto$	$J2\|r_j, p_{ij} = 1\|\Sigma C_j$
[DLW92]	$P2\|pmtn, r_j\|\Sigma U_j$	$\propto$	$J2\|r_j, p_{ij} = 1\|\Sigma U_j$
[BCS74],[LRKB77]	$P2\|pmtn\|\Sigma w_j C_j$	$\propto$	$J2\|p_{ij} = 1\|\Sigma w_j C_j$
[K72]	$1\|pmtn\|\Sigma w_j U_j$	$\propto$	$J2\|p_{ij} = 1\|\Sigma w_j U_j$
[DL90]	$1\|pmtn\|\Sigma T_j$	$\propto$	$J2\|p_{ij} = 1\|\Sigma T_j$
[LRKB77]	$P2\|\|C_{\max}$	$\propto$	$J2\|no\,wait, p_{ij} = 1\|C_{\max}$
[BCS74],[LRKB77]	$P2\|\|\Sigma w_j C_j$	$\propto$	$J2\|no\,wait, p_{ij} = 1\|\Sigma w_j C_j$

Table 5. Reductions proving the strong NP-hardness

[DLY91]	$P2\|pmtn, chains\|\Sigma C_j$	$\propto$	$J2\|chains, p_{ij} = 1\|\Sigma C_j$
[LLLRK84]	$1\|pmtn, r_j\|\Sigma w_j C_j$	$\propto$	$J2\|r_j, p_{ij} = 1\|\Sigma w_j C_j$
[L77],[LRKB77]	$1\|pmtn\|\Sigma w_j T_j$	$\propto$	$J2\|p_{ij} = 1\|\Sigma w_j T_j$
[DLY91]	$P2\|chains\|C_{\max}$	$\propto$	$J2\|no\,wait, chains, p_{ij} = 1\|C_{\max}$
[DLY91]	$P2\|chains\|\Sigma C_j$	$\propto$	$J2\|no\,wait, chains, p_{ij} = 1\|\Sigma C_j$
[LRKB77]	$1\|r_j\|\Sigma C_j$	$\propto$	$J2\|no\,wait, r_j, p_{ij} = 1\|\Sigma C_j$
[LRKB77]	$1\|r_j\|L_{\max}$	$\propto$	$J2\|no\,wait, r_j, p_{ij} = 1\|L_{\max}$
[L77],[LRKB77]	$1\|\|\Sigma w_j T_j$	$\propto$	$J2\|no\,wait, p_{ij} = 1\|\Sigma w_j T_j$

Earlier NP-hardness proofs for $J2|no\,wait, p_{ij} = 1|C_{\max}$ [T85],[SL86] and $J2|p_{ij} = 1|\Sigma w_j U_j$ [K96A] employ reductions from PARTITION and $1||\Sigma w_j U_j$ [K72], respectively. As Lenstra [L] points out, the NP-hardness proofs in the strong sense for $J2|chains, p_{ij} = 1|\Sigma C_j$, $J2|r_j, p_{ij} = 1|\Sigma w_j C_j$ and $J2|p_{ij} = 1|\Sigma w_j T_j$ can be obtained by reductions from 3-PARTITION [GJ78].

Concluding remarks

Summing up we can infer that the complexity status of unit-time job-shop problems with criteria $C_{\max}$, ΣC_j, $\Sigma w_j C_j$, ΣT_j, $\Sigma w_j T_j$, ΣU_j and $\Sigma w_j U_j$ is clear. The only gap is $J2|r_j, p_{ij} = 1|L_{\max}$ which remains as minimal as maximal open problem. Theorem 3.1 proves the reduction $P2|pmtn, r_j|L_{\max} \propto J2|r_j, p_{ij} = 1|L_{\max}$, however this says nothing about the complexity of the latter problem because even $Q|pmtn, r_j|L_{\max}$ [M82],[FG86] is solvable in polynomial time.

Theorem 3.1 can be useful in identical-parallel-machines scheduling. For example, in the case of bounded number of jobs the reduction $P|pmtn, prec, r_j, n \leq \bar{n}|\Sigma w_j T_j \propto J|prec, r_j, n \leq \bar{n}, p_{ij} = 1|\Sigma w_j T_j$, where the latter problem is solvable in polynomial time [BK96], shows that the former problem (which certainly is not trivial) is also so.

There are cases in which Theorem 3.1 reduces problems solvable in polynomial time to NP-hard problems or ordinarily NP-hard problems to strongly NP-hard problems. It proves that some unit-time job-shop problems are harder than their identical-parallel-machines counterparts, unless P=NP. For example,

[MC69],[MC70]	$P2\|pmtn, prec\|C_{\max}$	$\propto$	$J2\|prec, p_{ij} = 1\|C_{\max}$	[T85],[TSS94]
[M59]	$P3\|pmtn\|C_{\max}$	$\propto$	$J3\|p_{ij} = 1\|C_{\max}$	[LRK79]
[M59]	$P3\|pmtn\|\Sigma C_j$	$\propto$	$J3\|p_{ij} = 1\|\Sigma C_j$	[L]
[CMM67]	$P3\|\|\Sigma C_j$	$\propto$	$J3\|no\,wait, p_{ij} = 1\|\Sigma C_j$	[SL86]
[LRKB77]	$P3\|\|C_{\max}$	$\propto$	$J3\|no\,wait, p_{ij} = 1\|C_{\max}$	[SL86]

The problems on the right side are strongly NP-hard. Meanwhile, the problems on the left side (and even their extensions [LLRKS93]) are solvable in polynomial time except the NP-hard $P3||C_{\max}$ which is ordinarily NP-hard because even $Qm|r_j|C_{\max}$ can be solved in pseudopolynomial-time [LLRKS93]. Informally, the NP-hardness of the first four and the strong NP-hardness of the fifth unit-time job-shop problem in the above reductions are independent of identical-parallel-machines scheduling.

Although Lemma 2.2 is proved only for the criteria $C_{\max}$, $L_{\max}$, ΣC_j, $\Sigma w_j C_j$, ΣT_j, $\Sigma w_j T_j$, ΣU_j, $\Sigma w_j U_j$ the proof A can be easily adjusted for the criteria $f_{\max}$ and Σf_j, where f_j are *nondecreasing real cost functions* with the derivatives bounded by an exponential e in $q(l)$ for some polynomial q. For this purpose, the *mean value theorem* has to be applied to get the inequality

$$[a\sigma + b, aI] = [a\sigma, aI] + b \sum_{j=1}^{n} f'_j(C_j + b_j c) \leq [a\sigma, aI] + bne,$$

where $0 \leq c \leq 1$, and w has to be replaced by ne. Thus, Theorem 3.1 is true for this case as well because all the other lemmas are true for any minimization criterion. A criterion which makes a unit-time job-shop problem easier than its identical-parallel-machines counterpart has not been found.

Acknowledgements

The author thanks Peter Brucker, Han Hoogeveen, Jan Karel Lenstra, Joseph Leung, Nicholas Solntseff and anonymous referees for their comments.

References

[BK96] P. Brucker and A. Kraemer, *Polynomial algorithms for resource-constrained and multiprocessor task scheduling problems*, EJOR **90** (1996), 214–226.

[BKS95] P. Brucker, S. A. Kravchenko and Y. N. Sotskov, *On the complexity of two machine job-shop scheduling with regular objective functions*, OSM Reihe P, Heft 172, 1995.

[BCS74] J. L. Bruno, E. G. Coffman, Jr., and R. Sethi, *Scheduling independent tasks to reduce mean finishing time*, Comm. ACM **17** (1974), 382–387.

[CMM67] R. W. Conway, W. L Maxwell, and L. W. Miller, *Theory of scheduling*, Addison-Wesley, Reading, MA, 1967.

[DL90] J. Du and J. Y.-T. Leung, *Minimizing total tardiness on one machine is NP-hard*, Math. Oper. Res. **15** (1990), 483–495.

[DLW92] J. Du, J. Y.-T. Leung, and C. S. Wong, *Minimizing the number of late jobs with release time constraints*, J. Combin. Math. Combin. Comput. **11** (1992), 97–107.

[DLY90] J. Du, J. Y.-T. Leung, and G. H. Young, *Minimizing mean flow time with release time constraints*, Theor. Comput. Sci. **75** (1990), 347–355.

[DLY91] J. Du, J. Y.-T. Leung, and G. H. Young, *Scheduling chain-structured tasks to minimize makespan and mean flow time*, Inform. and Comput. **92** (1991), 219–236.

[FG86] A. Federgruen and Groenevelt, *Preemptive scheduling of uniform machines by ordinary network flow techniques*, Management Sci. **32** (1986), 431–349.

[GJ78] M. R. Garey and D. S. Johnson, *Strong NP-completeness results: motivation, examples and implications*, J. ACM **25** (1978), 499–508.

[GJ79] M. R. Garey and D. S. Johnson, *Computers and intractability: a guide to the theory of NP-completeness*, Freeman, San Francisco, 1979.

[GLLRK79] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan, *Optimization and approximation in deterministic sequencing and scheduling: a survey*, Ann. Discrete Math. **5** (1979), 287–326.

[K72] R. M. Karp, *Reducibility among combinatorial problems*, in Complexity of Computer Computations (R. E. Miller and J. W. Thatcher, ed.), Plenum Press, New York, 1972, pp. 85–103.

[K96A] S. A. Kravchenko, *Minimizing the number of late jobs for the two-machine unit-time job-shop scheduling problem*, submitted to Discrete Appl. Math.

[K96B] S. A. Kravchenko, *A polynomial algorithm for a two-machine no-wait job-shop scheduling problem*, to appear in EJOR.

[K89] W. Kubiak, *A pseudopolynomial algorithm for a two-machine no-wait job shop problem*, EJOR **43** (1989), 267–270.

[KSS95] W. Kubiak, S. Sethi and C. Sriskandarajah, *An efficient algorithm for a job shop problem*, Mathematics of Industrial Systems **1** (1995), 203–216.

[KT96] W. Kubiak and V. G. Timkovsky, *A polynomial-time algorithm for total completion time minimization in two-machine job-shop with unit-time operations*, EJOR **94** (1996), 310–320.

[LLLRK84] J. Labetoulle, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan, *Preemptive scheduling of uniform machines subject to release dates*, in Progress in Combinatorial Optimization (W. R. Pulleyblank, ed.), Academic Press, 1984, pp. 245–261.

[L77] E. L. Lawler, *A 'pseudopolynomial' algorithm for sequencing jobs to minimize total tardiness*, Ann. Discrete Math. **1** (1977), 331–342.

[L82] E. L. Lawler, *Preemptive scheduling of precedence-constrainted jobs on parallel machines*, in Deterministic and Stochastic Scheduling (M. A. H. Dempster, J. K. Lenstra and A. H. G. Rinnooy Kan, eds.), Reidel, Dordrecht, 1982, pp. 101–123.

[LLRK82] E. L. Lawler, J. K. Lenstra and A. H. G. Rinnooy Kan, *Recent developments in deterministic sequencing and scheduling: a survey*, in Deterministic and Stochastic Scheduling (M. A. H. Dempster, J. K. Lenstra and A. H. G. Rinnooy Kan, eds.), Reidel, Dordrecht, 1982, pp. 35–73.

[LLRKS93] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, *Sequencing and scheduling: algorithms and complexity*, in Handbook on Operations Research and Management Science, Volume 4: Logistics of Production and Inventory (S. C. Graves, A. H. G. Rinnooy Kan, and P. Zipkin, eds.), Elsevier Science Publishers B. V., Amsterdam, 1993, pp. 445–552.

[L] J. K. Lenstra, *Private communication.*

[LRK79] J. K. Lenstra and A. H. G. Rinnooy Kan, *Computational complexity of discrete optimization problems*, Ann. Discrete Math. **4** (1979), 121–140.

[LRKB77] J. K. Lenstra, A. H. G. Rinnooy Kan, and P. Brucker, *Complexity of machine scheduling problems*, Ann. Discrete Math. **1** (1977), 343–362.

[M82] C. U. Martel, *Preemptive scheduling with release times, deadlines and due dates*, J. ACM **29** (1982), 812–829.

[M59] R. McNaughton, *Scheduling with deadlines and loss functions*, Management Sci. **6** (1959), 1–12.

[MC69] R. R. Muntz and E. G. Coffman Jr, *Optimal preemptive scheduling on two-processor systems*, IEEE Trans. Comput. **C-18**, 1014–1020.

[MC70] R. R. Muntz and E. G. Coffman Jr, *Preemptive scheduling of real time tasks on multiprocessor systems*, J. ACM **17**, 324–338.

[S91] Yu. N. Sotskov, *On the complexity of shop scheduling problems with two or three jobs*, EJOR **53** (1991), 326–336.

[SL86] C. Sriskandarajah and P. Ladet, *Some no-wait jobs scheduling problems: complexity results*, EJOR **24** (1986), 424–438.

[TSS94] V. S. Tanaev, Y. N. Sotskov and V. A. Strusewich, *Scheduling theory: multi-stage systems*, Kluwer, 1994.

[T81] V. G. Timkovsky, *Computational complexity and approximation of the cycle shop scheduling problem*, in The 1st All-Union Consultation on Statistical and Discrete Analysis of Non-Numerical Information, on Expert Estimates and Discrete Optimization (Yu. N. Turin, ed.), Kazakh State University, Moscow–Alma-Ata, 1981, pp. 220–221. (in Russian)

[T85] V. G. Timkovsky, *On the complexity of scheduling an arbitrary system*, Soviet Journal of Computer and System Sciences (1885), no. 5, 46–52.

[T86] V. G. Timkovsky, *An approximation for the cycle shop scheduling problem*, Ekonomika i Matematicheskije Metodi **22** (1986), no. 1, 171–174. (in Russian)

[T93] V. G. Timkovsky, *The complexity of unit-time job-shop scheduling*, Technical Report, Department of Computer Science and Systems, McMaster University, Hamilton, 1993.

[T97] V. G. Timkovsky, *A polynomial-time algorithm for the two-machine unit-time release-date job-shop schedule-length problem*, Discrete Appl. Math. **77** (1997), 185–200.