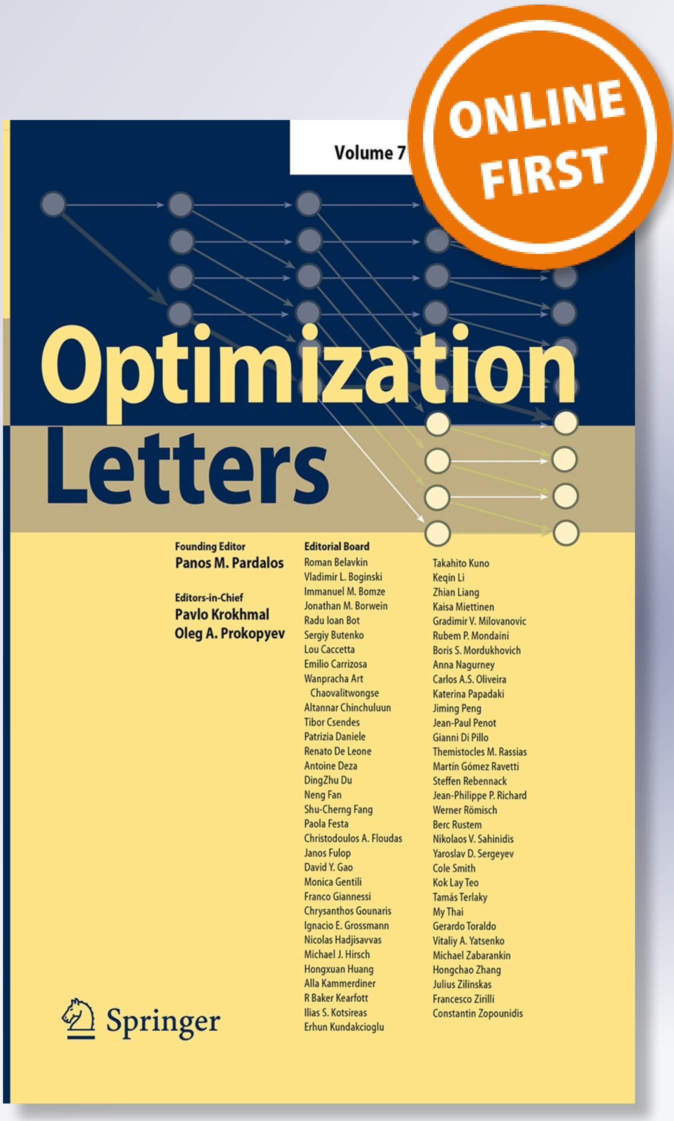
Two-dedicated-machine scheduling problem with precedence relations to minimize makespan

Evgeny Gafarov & Alexandre Dolgui

Optimization Letters

ISSN 1862-4472

Optim Lett
DOI 10.1007/s11590-013-0671-0



Springer

Optim Lett
DOI 10.1007/s11590-013-0671-0

ORIGINAL PAPER

Two-dedicated-machine scheduling problem with precedence relations to minimize makespan

Evgeny Gafarov · Alexandre Dolgui

Received: 18 April 2012 / Accepted: 24 June 2013

Abstract Two-dedicated-parallel-machine scheduling problem with precedence constraints to minimize makespan is considered. This problem originally appeared as a sub-problem in assembly line balancing but it has also its own applications. Complexity and approximation results for this scheduling problem and its special cases with chains of jobs or equal-processing-times are presented.

Keywords Parallel machine scheduling · Complexity · Assembly line balancing

1 Introduction

The following two-dedicated-parallel-machine scheduling problem is considered.

There is a set $N = \{1, 2, \ldots, n\} = N_1 \bigcup N_2 \bigcup N_{1or2} \bigcup N_{1and2}$ of n jobs that must be processed on two parallel machines. Jobs from N_1 have to be processed on the first machine, jobs from N_2 on the second machine, jobs from N_{1or2} can be processed on any of them, jobs from N_{1and2} use both machines simultaneously. Job preemption is not allowed. Each machine can handle only a single job at a time. All the jobs are assumed to be available for processing at time 0. For each job $j, j \in N$, a processing time $p_j \geq 0$ is known. Furthermore, arbitrary finish-start precedence relations $i \to j$ are introduced between jobs according to an acyclic directed graph G. Let S_j be a starting time of job j, $j = 1, 2, \ldots, n$. If $i \to j$,

E. Gafarov · A. Dolgui
Ecole Nationale Superieure des Mines, CNRS UMR6158, LIMOS, 42023 Saint-Etienne, France
e-mail: dolgui@emse.fr

E. Gafarov (✉)
Institute of Control Sciences of the Russian Academy of Sciences,
Profsoyuznaya St. 65, Moscow 117997, Russia
e-mail: axel73@mail.ru

Published online: 03 July 2013

then $S_i + p_i \leq S_j$. The objective is to determine the starting time S_j for each job j, $j = 1, 2, \ldots, n$, in such a way that the given precedence relations are fulfilled and makespan $C_{max} = \max_{j=1}^{n} C_j$, where $C_j = S_j + p_j$, is minimized. Denote this problem as $P2|prec, N_1, N_2, N_{1or2}, N_{1and2}|C_{max}$.

In this paper, we consider the special case of this problem, where $N_{1or2} = N_{1and2} = \emptyset$. We denote this case by $P2|prec, N_1, N_2|C_{max}$. A similar problem without precedence relations was considered in [1], where jobs are assigned to a machine in advance and incompatibility relations were defined over the tasks which forbids any two incompatible tasks to be processed at the same time.

Two-machine problems are a special case of scheduling problems with parallel machines (see, e.g., a survey [2]). Papers on different two parallel machines scheduling appear regularly. If $N_{1or2} = N$, i.e., $N_1 = N_2 = N_{1and2} = \emptyset$, then we have a well known problem $P2|prec|C_{max}$ for two identical parallel machines which is NP-hard [2]. An early-tardy scheduling problem for two parallel machines where some jobs need to be processed by one machine, while the others have to be processed by both machines simultaneously, was presented in [3].

Two-dedicated-parallel-machine scheduling problem originally appeared as a subproblem of the well-known two-sided assembly line balancing problem (TSALBP)[4, 5], where working stations are left- and right-sided. That means to solve TSALBP, some $P2|prec, N_1, N_2|C_{max}$ instances have to be solved. This scheduling problem also has other practical interpretations. For example, this is similar to a problem of a master and his apprentice who have to coordinate their interrelated operations. This latter model is widely used in practice.

In this paper we study complexity and approximability of the problem $P2|prec, N_1, N_2|C_{max}$. To the best of our knowledge there are no publications for the two-dedicated-parallel-machine scheduling with precedence relations. Although, publications on similar problems appear often. For example, in [6] a flow shop problem with dedicated machines is considered.

The rest of the paper is organized as follows. Section 2 presents some new complexity results. Approximations are discussed in Sect. 3. In Sect. 4 a connection to assembly line balancing problems is considered.

2 Complexity results

Denote by $P2|chain, N_1, N_2|C_{max}$ the special case of two-dedicated-parallel-machine scheduling problem, where G consists only chains of jobs and by $P2|prec, p_j = 1, N_1, N_2|C_{max}$ a special case with equal-processing-times of jobs.

3-Partition problem:
A set $N = \{b_1, b_2, \ldots, b_n\}$ of $n = 3m$ positive integers is given, where $\sum_{i=1}^{n} b_j = mB$ and $\frac{B}{4} < b_j < \frac{B}{2}$, $j = 1, 2, \ldots, n$. Does there exist a partition of N into m subsets $\overline{N}_1, \overline{N}_2, \ldots, \overline{N}_m$ such that each subset consists in exactly three numbers and the sum of the numbers in each subset is the same, i.e.,

$$\sum_{b_j \in \overline{N}_1} b_j = \sum_{b_j \in \overline{N}_2} b_j = \cdots = \sum_{b_j \in \overline{N}_m} b_j = B?$$

Lemma 1 *$P2|chain, N_1, N_2|C_{max}$ is NP-hard in the strong sense.*

Proof We demonstrate a reduction from the 3-partition problem. Given an instance of the 3-partition problem with $3m$ numbers, construct an instance of $P2|chain, N_1, N_2|C_{max}$ with $5m-1$ jobs. The first $3m$ jobs are independent, $p_j = b_j,\ j = 1, 2, \ldots, 3m$, and there is a chain of jobs $3m + 1 \rightarrow 3m + 2 \rightarrow 3m + 3 \rightarrow \cdots \rightarrow 5m - 1$, where $p_j = B,\ j = 3m + 1, 3m + 3, \ldots, 5m - 1$ and $p_j = 1,\ j = 3m + 2, 3m + 4, \ldots, 5m - 2$. Furthermore, $N_1 = \{3m + 1, 3m + 3, \ldots, 5m - 1\}$ and $N_2 = \{1, 2, \ldots, 3m, 3m + 2, 3m + 4, \ldots, 5m - 2\}$, see Fig.1a.

If and only if for this instance of the 3-partition problem the answer is "YES", then there is a schedule in which a subset of jobs which corresponds to set $\overline{N}_i$ is processed in parallel with job $3m + (2i - 1),\ i = 1, 2, \ldots, m$. Starting times $S_{3m+2i-1} = (B + 1)(i - 1),\ i = 1, 2, \ldots, m$, and $S_{3m+2i} = Bi + (i - 1),\ i = 1, 2, \ldots, m - 1$. For such a schedule $C_{max} = mB + m - 1$. □

Clique problem

Given a graph $G = (V, E)$ and an integer k, does G have a clique (i.e., a complete subgraph) on k vertices?

Lemma 2 *$P2|prec, p_j = 1, N_1, N_2|C_{max}$ is NP-hard in the strong sense.*

Proof We demonstrate a reduction from the Clique problem[1]. Introduce a job J_v for every vertex $v \in V$ and a job J_e for every edge $e \in E$, with $J_v \rightarrow J_e$ whenever v is endpoint of e. Denote $\overline{n} = |V|$ and $l = |E|$. The processing times of all the jobs are equal to 1. Jobs $J_v \in N_2, \forall v \in V$, and $J_e \in N_1, \forall e \in E$. We also add the chain of jobs $\overline{n} + 1 \rightarrow \overline{n} + 2 \rightarrow \overline{n} + 3 \rightarrow \overline{n} + 4$, where $p_{\overline{n}+1} = k,\ p_{\overline{n}+2} = k(k - 1)/2,\ p_{\overline{n}+3} = n - k,\ p_{\overline{n}+4} = l - k(k - 1)/2$ and $\overline{n} + 1, \overline{n} + 3 \in N_1,\ \overline{n} + 2, \overline{n} + 4 \in N_2$, see Fig.1b.

There is a schedule for which $C_{max} = \overline{n} + l = \sum_{i=\overline{n}+1}^{\overline{n}+4} p_i$, if and only if for this instance of clique problem the answer is "YES".

Denote a clique by $G'(V', E')$. Jobs $J_v,\ v \in V'$, are processed in parallel with job $\overline{n} + 1$. Jobs $J_e,\ e \in E'$, are scheduled to be executed in parallel with job $\overline{n} + 2$. Jobs $J_v,\ v \in V \backslash V'$ are processed in parallel with job $\overline{n} + 3$. Jobs $J_e,\ e \in E \backslash E'$ are executed in parallel with job $\overline{n} + 4$.

If there is no clique of size k, then after scheduling of k jobs $J_v,\ v \in V$, we will be able to schedule no more than $k(k - 1)/2 - 1$ jobs $J_e,\ e \in E$, in parallel with job $\overline{n} + 2$.

The jobs $\overline{n}+1, \overline{n}+2, \overline{n}+3, \overline{n}+4$ can be substituted for chains of $k,\ k(k-1)/2,\ \overline{n}-k$, and $l - k(k - 1)/2$ equal-processing-time jobs, respectively, i.e. this is the special case where $p_j = 1$.

So, the Lemma is proven. □

As a consequence from Lemma 2, for the special case $P2|prec, p_j = 1, N_1, N_2|C_{max}$ the approximation ratio of polynomial time algorithms is not less than $2/n$

[1] A similar idea was used in [2] for $P|prec, p_j = 1|C_{max}$ problem.

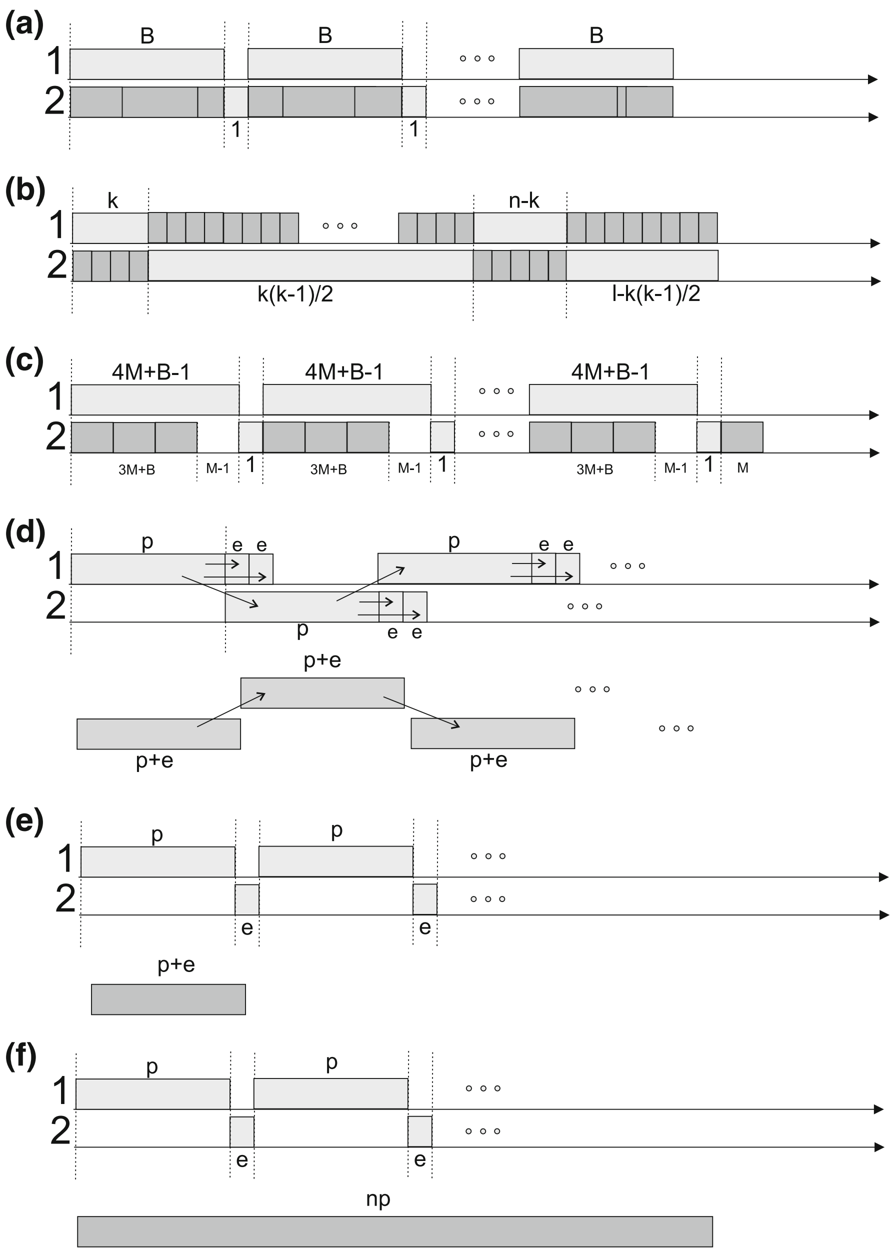


Fig. 1 Examples

and there is no fully-polynomial time approximation schema (FPTAS) for this special case.

Denote the problem where preemptions of jobs are allowed by $P2|prec, pmtn., N_1, N_2|C_{max}$.

Corollary 1 *$P2|prec, pmtn., N_1, N_2|C_{max}$ is NP-hard in the strong sense.*

Let $C^*_{max}(pmtn.)$ be the minimal makespan for the problem with preemptions.

Lemma 3 *For the problem $P2|prec, N_1, N_2|C_{max}$ an inequality $\frac{C^*_{max}}{C^*_{max}(pmtn.)} < 2$ holds and there is an instance for which $\frac{C^*_{max}}{C^*_{max}(pmtn.)} \approx 2$.*

Proof It's obvious that

$$\frac{1}{2}\sum_{j\in N} p_j \le C^*_{max}, C^*_{max}(pmtn.) < \sum_{j\in N} p_j.$$

So, the first part of Lemma is true.

To prove the second part let us consider an instance with a chain of $2n-1$ jobs, $p_j = p,\ j = 1, 3, \ldots, 2n-1$, and $p_j = e,\ j = 2, 4, \ldots, 2n-2$. In addition, an independent job $2n$ is given with $p_{2n} = np$. $N_1 = \{1, 3, \ldots, 2n-1\}$ and $N_2 = \{2, 4, \ldots, 2n-2, 2n\}$ (see Fig.1f). For such an instance $C^*_{max} = (n-1)(p+e)+np$ and $C^*_{max}(pmtn.) = np + (n-1)e$. Then, for $e \to 0$ the second part of the Lemma is true. □

3 Approximation by a list scheduling algorithm

To solve problems with precedence relations (e.g., SALBP-1[2], $P2|prec|C_{max}$) enumeration schemas based on the well-known List Scheduling (LS) Algorithm are usually used. The problem $P2|prec, N_1, N_2|C_{max}$ can be solved by an algorithm based on LS as well.

The main idea of LS is as follows: on each step $j = 1, 2, \ldots, n$, choose a job (operation) for which all predecessors are already scheduled and assign it from the earliest possible starting time according to precedence relations and resource restrictions. According to such an algorithm only *active* schedules will be constructed for which there is no job which can be shifted to an earlier starting time without violating precedence or resource constraints. Evidently among active schedules there are optimal ones, that's why, an optimal solution can be presented as a sequence (permutation) of n jobs, which denotes the order of jobs' choice in LS. Different domination rules are used in LS to choice a job e.g., a job with the maximal processing time among ready to be scheduled (LPT), or a job which belongs to a critical path (CP), etc.

List Scheduling is also widely used to compute Upper Bounds, i.e., to find feasible solutions. The question appears: "Which approximation ratios has LS algorithm with different domination rules?" Let us denote the optimal objective function value by C^*_{max} and the objective function value for the solution constructed by LS with a domination rule α by $C_{max}(LS_\alpha)$. It is known [2], that for the problem $P|prec|C_{max}$ we have $\frac{4}{3} \le \frac{C_{max}(LS_\alpha)}{C^*_{max}} < 2$ for any domination rule α which can be verified in a polynomial time.

[2] for the definition of SALBP-1, see Sect. 4.

Certainly, for the problem $P2|prec, N_1, N_2|C_{max}$, we have $\frac{C_{max}(LS_\alpha)}{C^*_{max}} < 2$, since

$$\frac{1}{2}\sum_{j\in N} p_j \le C^*_{max},\ C_{max}(LS_\alpha) < \sum_{j\in N} p_j.$$

For some problems it is useful to know the worst possible active schedule constructed by LS. Such problems with the opposite optimality criteria have both theoretical and practical significance [7]. Denote by $P2|prec, N_1, N_2|C_{max} \to \max$ a problem with opposite optimality criterion: maximizing the makespan, where only active schedules are considered. Unfortunately, this maximization problem is strongly NP-hard too.

Lemma 4 *$P2|chains, N_1, N_2|C_{max} \to \max$ is NP-hard in the strong sense.*

Proof We suggest a reduction from the 3-partition problem. Let an instance of the 3-partition problem with $3m$ numbers be given. Let $M = (mB)^2$. Construct an instance of $P2|chain, N_1, N_2|C_{max}$ with $5m+1$ jobs. The first $3m+1$ operations are independent, $p_j = M+b_j,\ j = 1, 2, \ldots, 3m$, and $p_{3m+1} = M$. In addition, there is a chain of jobs $3m+2 \to 3m+3 \to 3m+4 \to \cdots \to 3m+2m+1$, where $p_j = 4M+B-1,\ j = 3m+2, 3m+4, \ldots, 3m+2m$ and $p_j = 1,\ j = 3m+3, 3m+5, \ldots, 3m+2m+1$. Furthermore, $N_1 = \{3m+2, 3m+4, \ldots, 3m+2m\}$ and $N_2 = \{1, 2, \ldots, 3m+1, 3m+3, 3m+5, \ldots, 3m+2m+1\}$, see Fig.1c.

If and only if for the instance of the 3-partition problem the answer is "YES", then there is an active schedule in which the subset of jobs which corresponds to set $\overline{N}_i$ is processed in parallel with a job $3m+1+(2i-1),\ i = 1, 2, \ldots, m$. Starting times $S_{3m+1+(2i-1)} = (4M+B-1+1)(i-1),\ i = 1, 2, \ldots, m$, and $S_{3m+1+2i} = (4M+B-1)i+(i-1),\ i = 1, 2, \ldots, m$. Job $3m+1$ is processed independently from the time $(4M+B-1+1)m$. For this schedule $C_{max} = (4M+B)m+M$.

If the answer is "NO", then $C_{max} = (4M+B)m$, and there is a job $3m+1+(2i-1),\ i \in \{1, 2, \ldots, m\}$ which is processed in parallel with 4 jobs from set $\{1, 2, \ldots, 3m+1\}$ (including job $3m+1$). □

We can show that the approximation ratios of LS with the following domination rules is ≈ 2: CP is the critical path rule (choose a job which belongs to a critical path [2,9]), LPT (choose a job with the maximal processing time), MS (choose a job with the maximal number of immediate successors).

Lemma 5 *There are instances for which*

$$\frac{C_{max}(LS_\alpha)}{C^*_{max}} \approx 2,\ \alpha \in \{CP, LPT, MS\}.$$

Proof For the rule MS, consider an instance from Fig.1d. In this instance, there is a chain of k jobs with processing times p, where k is odd. Each of these jobs precedes two jobs with processing times e. Additionally, there is a chain of k jobs with processing times $p+e$. Then, $C_{max}(LS_{MS}) = (k-1)p+2e+k(p+e)$ and $C^*_{max} = k(p+e)+ke/2$. For $k \to \infty,\ e \to 0$, the lemma is true.

For the rule CP, consider an instance from Fig.1e. In this instance, we have a chain of $2k$ jobs with processing times p and e. Additionally, there is k independent jobs with processing times $p+e$ which have to be processed on the second machine. Then, $C_{max}(LS_{CP}) = 2k(p+e)$ and $C^*_{max} = k(p+2e)$. For $e \to 0$, the lemma is true.

If the instance for CP is modified by adding a job $2k+1$ with processing time $e/2$ which precedes all independent jobs with the processing time $p+e$, then for this modified instance $\frac{C_{max}(LS_{LPT})}{C^*_{max}} \approx 2$. □

We conjecture that the same relation is true for other rules α computed in a polynomial time.

4 Connection to assembly line balancing problems

Two-dedicated-parallel-machine scheduling problem originally appeared as a subproblem of the well-known two-sided assembly line balancing problem (TSALBP) [4,5].

To present TSALBP, we begin with a description of the basic simple assembly line balancing problem (SALBP). For SALBP, a single-model paced assembly line which continuously manufactures a homogeneous product in large quantities is considered (mass production). SALBP consists in finding an optimal line balance for a given cycle time c or a given number of machines, i.e., a feasible assignment of given operations to stations such that either the number of stations used m reaches its minimal value (SALBP-1) or the cycle time is minimized for a given number of stations (SALBP-2).

The SALBP-1 is formulated as follows. A set $N = \{1, 2, \dots, n\}$ of operations is given. For each operation $j \in N$, a processing time $t_j \geq 0$ is known. A cycle time $c \geq \max\{t_j,\ j \in N\}$ is also known and fixed. Furthermore, finish-start precedence relations $i \to j$ are defined between operations according to an acyclic directed graph G. The objective is to assign each operation $j,\ j = 1, 2, \dots, n$, to a station in such a way that:

- number m of stations used is minimized;
- for each station $k = 1, 2, \dots, m$, its total load time $\sum_{j \in N_k} t_j$ does not exceed c, where N_k is the set of operations assigned to station k;
- precedence relations are fulfilled, i.e. if $i \to j,\ i \in N_{k_1}$ and $j \in N_{k_2}$, then $k_1 \leq k_2$.

SALBP-1 is NP-hard in the strong sense. For surveys on results for SALBP-1, see [8–10]. There exists a special electronic library of benchmark data for this problem: http://www.assembly-line-balancing.de.

In contrast with SALBP-1, in the two-sided assembly line balancing problem of type 1 (TSALBP-1) instead of single stations, pairs of opposite stations are disposed in parallel. They work simultaneously at opposite sides of the same workpiece. Operations have to be performed on either a side of the line or can require both sides simultaneously.

While for SALBP-1 all jobs from a set N_k where $\sum_{j \in N_k} t_i \leq c$ can be processed on a single station, for TSALBP-1 another question appears: Is it possible to process all jobs from a set N_l where $c < \sum_{j \in N_l} t_i \leq 2c$ on a pair of opposite stations? So, the problem $P2|prec, N_1, N_2, N_{1or2}, N_{1and2}|C_{max}$ needs to be solved.

We can present a reduction similar to the presented in Lemma 1. This reduction can be made from a decision version of SALBP-1 to $P2|prec, N_1, N_2|C_{max}$. In the decision version of SALBP-1, we have to answer the question, if a feasible line balance exists with m stations?

Lemma 6 *Decision version of SALBP-1 can be reduced to $P2|prec, N_1, N_2|C_{max}$ in a polynomial time.*

Proof For this reduction there is a subset of jobs which corresponds to the set of operations from SALBP-1 with the processing times $p_j = t_j,\ j = 1, 2, \dots, n$, and the same precedence relations. Furthermore, a chain of long and short jobs $n+1 \to n+2 \to n+3 \to \dots \to n+2m-1$ is given, where $p_j = c,\ j = n+1, n+3, \dots, n+2m-1$ and $p_j = 1,\ j = n+2, n+4, \dots, n+2m-2$. If and only if for the instance of SALBP-1 the answer is "YES", then there is a schedule for which $C_{max} = mc + (m-1)$. □

Since to solve TSALBP-1, solution methods for SALBP-1 can be used, where $P2|prec, N_1, N_2, N_{1or2}, N_{1and2}|C_{max}$ has to be solved as a subproblem, it seems to be interesting that SALBP-1 can be reduced to a special case of $P2|prec, N_1, N_2|C_{max}$. In addition, in our previous work, we proved that there are instances of SALBP-1 for which no known Branch and Bound algorithm with a Lower Bound computed in a polynomial time can solve instances with $n = 60$ operations in an appropriate time [11]. So, it seems to be inadvisable to try to construct an effective Branch and Bound algorithm for the general case of the problem under consideration.

5 Conclusion

In this paper, we presented some complexity and approximation results for the two-dedicated-parallel-machine scheduling problem with precedence relations and minimization of makespan. This problem is a sub-problem of well-known two-sided assembly line balancing problem, but it has also its own practical interpretations and applications. Our results show that this two-machine problem is not easier than well-known SALBP-1, i.e. there is no Branch and Bound algorithm with Lower Bounds computed in a polynomial time that can solve instances of a special case [11] even for $n = 60$ jobs in an appropriate time. For the future research a question appears: is there is a constant a, $1 < a < 2$, for which the problem is approximable in polynomial time?

Acknowledgments The work was partially supported by Saint-Etienne Méetropole, France and Labex IMOBS3. The authors thanks also Chris Yukna for his help with English.

References

1. Lushchakova, I.N., Strusevich, V.A.: Scheduling incompatible tasks on two machines. Eur. J. Oper. Res. **200**(2), 334–346 (2010)
2. Lawler, E.L., Lenstra, J.K., Rinnoy Kan, A.H.G., Shmoys, D.B.: Sequencing and Scheduling: Algorithms and Complexity, Report BS-8909, Stichting Matematisch Centrum, Amsterdam (1989)
3. Lin, S.F., Chen, J.Y.: Two parallel machines scheduling. Syst. Anal. Model. Simul. **42**(10), 1429–1437 (2002)

4. Bartholdi, J.: Balancing two-sided assembly lines: a case study. Int. J. Prod. Res. **31**, 2447–2461 (1993)
5. Tapkan, P., Ozbakir, L., Baykasoglu, A.: Bees algorithm for constrained fuzzy multi-objective two-sided assembly line balancing problem. Optim. Lett. **39**(9–10), 1009–1015 (2007)
6. Hadda, H., Dridi, N., Hajri-Gabouj, S.: A note on the two-stage hybrid flow shop problem with dedicated machines. Optim. Lett. (2011). doi:10.1007/s11590-011-0365-4
7. Aloulou, M.A., Kovalyov, M.Y., Portmann, M.-C.: Maximization problems in single machine scheduling. Ann. Oper. Res. **129**, 21–32 (2004)
8. Rekiek, B., Dolgui, A., Delchambre, A., Bratcu, A.: State of art of assembly lines design optimization. Annu. Rev. Control **26**(2), 163–174 (2002)
9. Scholl, A., Becker, C.: State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. Eur. J. Oper. Res. **168**(3), 666–693 (2006)
10. Baybars, I.: A survey of exact algorithms for the simple assembly line balancing. Manag. Sci. **32**, 909–932 (1986)
11. Gafarov, E.R., Dolgui, A.: Hard Special Case and Other Complexity Results for SALBP-1. Research Report LIMOS UMR CNRS 6158 (2012)