

Complexity of Scheduling Problem in Single-Machine Flexible Manufacturing System with Cyclic Transportation and Unlimited Buffers

Marie-Laure Espinouse¹ · Grzegorz Pawlak² ·
Malgorzata Sterna² 

Received: 25 January 2016 / Accepted: 29 December 2016 / Published online: 12 January 2017
© The Author(s) 2017. This article is published with open access at Springerlink.com

Abstract The paper concerns complexity studies on the scheduling problem arising in a simple flexible manufacturing system. The system consists of a single machine, one depot (both with unlimited buffers), and one vehicle (automated guided vehicle). The vehicle operates according to the regular metro strategy. This means that it travels in cycles of the constant length, without stops, transporting at most one job at a time between the depot and the machine. The machine executes available jobs in the non-preemptive way. The goal is to minimize the schedule length, i.e., to minimize the number of vehicle cycles necessary to transport and execute all jobs in the system. We prove the strong NP-hardness of this problem and show that any list algorithm has the worst-case performance ratio equal to 2. Moreover, we mention that special cases of the considered problem, with zero transportation times from the depot to the machine and from the machine to the depot, are polynomially solvable.

Keywords Flexible manufacturing system · Single machine · One depot · One vehicle · NP-hardness proof · Approximation algorithm

Communicated by Anita Schöbel.

✉ Malgorzata Sterna
malgorzata.sterna@cs.put.poznan.pl

Marie-Laure Espinouse
marie-laure.espinouse@g-scop.grenoble-inp.fr

Grzegorz Pawlak
grzegorz.pawlak@cs.put.poznan.pl

¹ Université Grenoble Alpes, CNRS, G-SCOP, Grenoble, France

² Institute of Computing Science, Poznan University of Technology, Poznan, Poland

Mathematics Subject Classification 90B35 · 11Y16**1 Introduction**

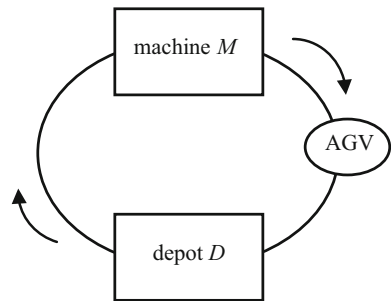
Flexible manufacturing systems (FMS) play an extremely important role in the contemporary industry. Most factories are equipped with machine stations and storage areas connected with a transportation system of various types such as robots, automated guided vehicles (AGV), and conveyors [1]. The production process in such systems is described by many parameters as well as constraints, and its quality is estimated with regard to many objective functions, from the classical ones, like the schedule length, to the more specialized ones, like the late work [2]. On the one hand, practical applications inspired intensive studies on flexible manufacturing systems within operational research, which resulted in numerous papers on specific problems [3–9], including robotic cells problems [10, 11] and surveys of results [12–16]. On the other hand, closely related problems, concerning automated guided vehicles traffic control systems, could be found in logistics, e.g., in automated container terminals [17–21].

Due to the strong connection to practice, studies on flexible manufacturing system focus mainly on complex models, since their main goal is to propose efficient methods for solving real optimization problems. For this reason, no systematic research on particular FMS models has been done, as one can observe for “classical” scheduling models. In this latter case, researchers have intensively explored the borderline between easy and intractable problems. From the theoretical point of view, it is interesting to determine the complexity status of the simplest FMS models, which are in general NP-hard.

In this paper, we considered a basic flexible manufacturing system which consists of one machine, one depot (both with unlimited buffers), and one vehicle (AGV) moving in cycles of the constant length. Such a model is the simplest system with the AGV-based transportation, and its complexity status has not been studied yet. Most literature has been devoted to FMS problems with many vehicles [22–24] and with the unidirectional single-loop transportation system [25]. A single vehicle has been analyzed in the context of more complex systems: with buffers [26], with many machines (station) [27], or with dedicated machines [23]. Particularly, Suri and Desiraju [28] considered the production environment with only one vehicle, but in a more complex FMS, namely in the FMS with a single discrete material handling device (MHD). Lacomme et al. [23] deal with the problem of simultaneously scheduling jobs on machines and identical automated guided vehicles, analyzing many machines and many AGVs.

In this work, we show that the basic problem of minimizing the schedule length on a single machine, one depot, unlimited buffers, and one AGV with the constant cycle length is intractable. In Sect. 2, we define the problem formally, and we provide practical motivations for it. In Sect. 3, we present the strong NP-hardness proof for the considered model. In Sect. 4, we show that the worst-case performance ratio for any list heuristic is equal to 2, and we comment on some polynomially solvable special

Fig. 1 FMS system with single machine, depot, and AGV



cases. The paper is concluded and the further research directions are mentioned in Sect. 5.

2 Problem Definition

We consider the problem of simultaneously scheduling and transporting jobs in the flexible manufacturing system (FMS), consisting of a single machine, a depot (storage), and one automated guided vehicle (AGV) (cf. Fig 1).

All jobs are available at time zero at depot D . They have to be delivered by the AGV from depot D to machine M for non-preemptive processing. A job, after being completed, has to be transported by the AGV from the machine back to the depot. The AGV travels between the machine and the depot, in cycles of the constant length, picking up and dropping off jobs, if they are available. The AGV keeps moving even if there is no job to be transported, unless the last job is delivered to the depot. The goal is to minimize the length of the whole schedule, i.e., to minimize the delivery time of the last job to depot D .

More formally speaking, we study the problem of executing a set of n jobs on machine M , $J = \{J_1, J_2, \dots, J_j, \dots, J_n\}$. Each job J_j is described by processing time p_j necessary for its completion by the machine. The transportation times from/to the depot are job independent. The AGV moves without any break from the depot to the machine in time t_1 , and from the machine to the depot in time t_2 , so its constant cycle time is equal to $T = t_1 + t_2$. The loading and unloading times are included in the transportation times t_1 and t_2 . Moreover, we assume that both the depot and the machine are equipped with buffers of unlimited capacity, the AGV can take only one job at a time, and it cannot stop either at the machine or at the depot to wait for a job. To construct a schedule for the machine and the AGV simultaneously, one has to determine for each job J_j three time moments:

- the depart time from the depot to the machine - t_j^D ,
- the completion time on the machine - ct_j^M ,
- the delivery time from the machine to the depot (i.e., the completion time at the depot) - ct_j^D .

Obviously, the processing of job J_j on machine M can start only after its delivery from depot D , i.e., $ct_j^M - p_j \geq t_j^D + t_1$. Similarly, the completion time of the job on the

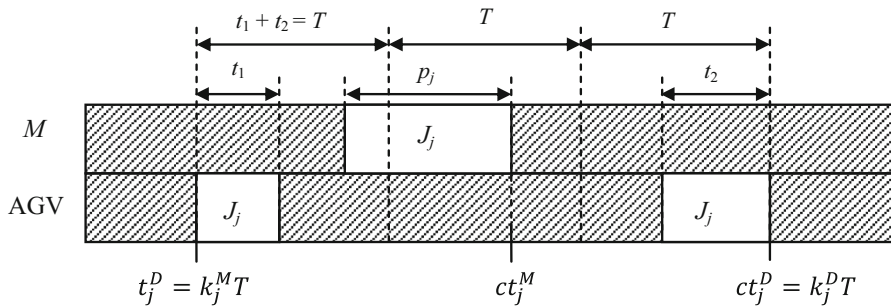


Fig. 2 Problem parameters for job J_j

machine determines its earliest feasible depart time to the depot, i.e., $ct_j^D - t_2 \geq ct_j^M$. Taking into account the fact that the AGV moves within the constant cycle time, and the depart time from the depot (t_j^D) and the delivery time to the depot (ct_j^D) can take only integer values, being multiples of the cycle length T . This means that for each job J_j one has to determine two cycle numbers (slots), in which this job is taken by the AGV to the machine (k_j^M), and delivered back to the depot (k_j^D). Hence, depart and delivery times are equal to $t_j^D = k_j^M T$ and $ct_j^D = k_j^D T$, respectively, where k_j^M is a nonnegative integer and k_j^D is a positive integer. Problem parameters for one job J_j are depicted in Fig 2.

The quality of a solution is estimated with regard to the schedule length (makespan). Actually, it is possible to determine two makespan values: at the machine and at the depot. The machine makespan is defined as the maximum job completion time at the machine, i.e., $C_{\max}^M = \max_{j=1 \dots n} \{ct_j^M\}$. This makespan is important, since it allows evaluating the machine utilization, but it does not take into account the necessity of delivering all jobs to the depot after their processing. The length of the whole schedule in the considered system is determined by the depot makespan, defined as the delivery time of the last job, $C_{\max}^D = \max_{j=1 \dots n} \{ct_j^D\}$, or as the total duration of vehicle cycles necessary to deliver and process all jobs, $C_{\max}^D = \max_{j=1 \dots n} \{k_j^D\} T$.

The problem of minimizing makespan on a single machine without any constraint is obviously easy. However, the existence of the transportation system makes the case intractable, as shown in the next section. The transportation system considered in the paper is a basic one: It works according to the metro transportation strategy, where a vehicle travels the system cyclically and continuously within the constant cycle time. A more flexible and popular strategy—transport on demand—allows AGVs to stop and wait at the machine and/or at the depot. Despite its inflexibility, the metro strategy is also an interesting subject of research. It can be found in production systems, namely subsystems, equipped with robot arms making cyclic movements or specialized types of conveyors. Furthermore, a very natural application field for this model comes from micro logistic systems. The AGV can represent a shuttle delivering and picking up groups of employees to/from a certain location (machine). The shuttle station repre-

sents a depot. Groups have to perform jobs of given lengths. The machine makespan models the time at which the whole project (the set of jobs) is finished, while the storage makespan represents the time in which all groups of employees return to the starting point. Similar problems can be observed in mines, where workers and machines are transported with lifts. We can meet them also in factories or warehouses in the process of loading/unloading goods/parts/material, where tracks cyclically deliver empty containers for loading, and pick up filled ones.

3 Problem Intractability

The problem of minimizing the schedule length in the simple FMS system, consisting of one machine, one depot, and one AGV, operating according to the metro strategy, is strongly NP-hard. Within this section, we will show the transformation from the strongly NP-complete three partition problem [29] to the decision counterpart of the considered optimization problem. We will show, in Theorem 3.1, that even a special case, with equal transportation times from/to the depot (i.e., $t_1 = t_2 = t$), is intractable, resulting in the intractability of the general problem with distinct times (i.e., $t_1 \neq t_2$).

Depot Makespan Minimization Problem (DMMP)

Instance: A set of n jobs $J = \{J_1, J_2, \dots, J_j, \dots, J_n\}$ characterized by processing time p_j , the AGV cycle length $T = t + t$, constant C_{\max} .

Question: Is it possible to find a schedule on the machine and the AGV simultaneously, with the depot makespan C_{\max}^D not exceeding C_{\max} ?

3-Partition Problem (3PP)

Instance: Constant B , a finite set of elements (integers) $A = \{a_1, a_2, \dots, a_j, \dots, a_{3m}\}$, such that $\frac{B}{4} < a_j < \frac{B}{2}$ for $j = 1 \dots 3m$ and $\sum_{j=1}^{3m} a_j = mB$.

Question: Is it possible to find m disjoint subsets $A_1, A_2, \dots, A_i, \dots, A_m$ of set A , such that $\sum_{a_j \in A_i} a_j = B$ for each $i = 1 \dots m$?

Theorem 3.1 *The depot makespan minimization problem is NP-complete in the strong sense.*

Proof Obviously, DMMP belongs to the class NP, since for a given schedule we can verify in polynomial time whether $C_{\max}^D \leq C_{\max}$. Furthermore, we can transform each instance of the strongly NP-complete 3PP to an instance of DMMP in the following way. We construct a set of $n = 4m + 1$ jobs J , which includes $3m$ jobs $\{J_1, \dots, J_{3m}\}$ with the processing times determined by the size of elements from 3PP ($p_j = a_j$), and additional $m + 1$ jobs $\{J_{3m+1}, \dots, J_{4m+1}\}$, each with processing time equal to $3B$. The AGV cycle length T is set to B , and threshold C_{\max} takes value $4B(m + 1)$. We ask whether it is possible to construct a schedule for DMMP with the depot makespan $C_{\max}^D \leq 4B(m + 1)$. Summing up, the instance transformation is as follows:

$$\begin{aligned} n &= 4m + 1, \\ p_j &= a_j \text{ for } j = 1, \dots, 3m, \\ p_j &= 3B \text{ for } j = 3m + 1, \dots, 4m + 1, \end{aligned}$$

$$T = t + t = B,$$

$$C_{\max} = 4B(m + 1),$$

where

$$\frac{B}{4} < p_j < \frac{B}{2} \text{ for } j = 1, \dots, 3m,$$

$$\sum_{j=1}^{3m} p_j = mB.$$

First, we will show that if the answer for DMMP is *YES*, i.e., there exists a schedule with $C_{\max}^D \leq 4B(m + 1)$, then 3PP has a solution too.

Obviously, the depot makespan (C_{\max}^D) cannot be shorter than the time necessary to deliver the first job to the machine (t), increased by the total job processing time ($\sum_{j=1}^n p_j$), and the time necessary to deliver the last job to the depot (t):

$$C_{\max}^D \geq t + \sum_{j=1}^n p_j + t$$

$$= T + \sum_{j=1}^{3m} p_j + \sum_{j=3m+1}^{4m+1} p_j = B + mB + (m + 1)(3B) = 4B(m + 1).$$

Thus, $C_{\max}^D = 4B(m + 1)$, and there is no idle time between two consecutive jobs on the machine. To construct a schedule on the machine without idle time, we cannot start it with any job $J_j \in \{J_1, \dots, J_{3m}\}$. Such a job would finish at the machine M at time $t + p_j < t + \frac{B}{2}$, while the second job can be delivered by the AGV to the machine at earliest at time $t + T = t + B$. The machine would be idle at least between $t + \frac{B}{2}$ and $t + B$. The first job $J_{[1]}$ in the sequence has to be taken from the subset $\{J_{3m+1}, \dots, J_{4m+1}\}$, since it finishes at the machine at $ct_{[1]}^M = t + p_{[1]} = t + 3B$, giving the AGV enough time to deliver the next jobs for continuous processing by the machine.

The first job $J_{[1]}$, after processing by the machine, is delivered to the depot at time $ct_{[1]}^D = t + 3B + t = 4B$. There are still $4m$ jobs to be delivered to the depot in the following time slots (i.e., the following AGV cycles). Since the AGV takes only one job at a time, it needs at least $4mT = 4mB$ time units to transport the remaining jobs from the machine to the depot. The last delivery to the depot finishes at earliest at time $ct_{[1]}^D + 4mT = 4B + 4mB = 4B(m + 1)$. Taking into account that $C_{\max}^D \leq 4B(m + 1)$, the AGV has to deliver jobs from the machine to the depot at each time slot between $4B$ and $4B(m + 1)$.

To achieve this goal, the second job cannot be taken from $\{J_{3m+1}, \dots, J_{4m+1}\}$. Such a job would be completed on the machine at $ct_{[1]}^M + 3B = t + 3B + 3B = t + 6B$, delivered to the depot at $t + 6B + t = 7B$, and there would be no job to be delivered to the depot from the machine at time moments $5B$ and $6B$.

Actually, the next three jobs have to be chosen from subset $\{J_1, \dots, J_{3m}\}$, followed by another job from subset $\{J_{3m+1}, \dots, J_{4m+1}\}$. The second job $J_{[2]} \in \{J_1, \dots, J_{3m}\}$ finishes at the machine at time $ct_{[2]}^M = ct_{[1]}^M + p_{[2]} = t + 3B + p_{[2]} < t + 3B + \frac{B}{2}$, and it is delivered to the depot at $ct_{[2]}^D = 5B$ (i.e., at the first available AGV cycle after its completion; $ct_{[2]}^D \geq ct_{[2]}^M + t = t + 3B + p_{[2]} + t = 4B + p_{[2]} \geq 4B + \frac{B}{4}$). If the third job came from the set $\{J_{3m+1}, \dots, J_{4m+1}\}$, then it would be completed only at time $t + 3B + p_{[2]} + 3B = t + 6B + p_{[2]}$, and it would be delivered to the depot at time $8B$, causing two idle AGV cycles ($6B$ and $7B$). Similarly, two jobs from $\{J_1, \dots, J_{3m}\}$ followed by a job from $\{J_{3m+1}, \dots, J_{4m+1}\}$ would cause one idle AGV cycle ($7B$). On the other hand, executing four jobs from $\{J_1, \dots, J_{3m}\}$ would cause idle time on the machine. Such jobs $J_i, J_j, J_k, J_l \in \{J_1, \dots, J_{3m}\}$ would finish on the machine at $t + 3B + p_i + p_j + p_k + p_l < t + 3B + 4\frac{B}{2} = t + 5B$, while the sixth job can be delivered to the machine at earliest at $t + 5B$. Thus, the fifth job $J_{[5]}$ has to be chosen from subset $\{J_{3m+1}, \dots, J_{4m+1}\}$ containing jobs of the longer processing time ($3B$).

Hence, in the considered schedule, two jobs $J_{[1]}, J_{[5]} \in \{J_{3m+1}, \dots, J_{4m+1}\}$ are separated by exactly three jobs $J_{[2]}, J_{[3]}, J_{[4]} \in \{J_1, \dots, J_{3m}\}$. Moreover, these three jobs have to be processed in exactly $p_{[2]} + p_{[3]} + p_{[4]} = B$ time units to obtain the depot makespan $C_{\max}^D \leq 4B(m+1)$. If $p_{[2]} + p_{[3]} + p_{[4]} < B$, then there would be idle time on the machine, between the completion time of job $J_{[4]}$ and the delivery of the fifth job $J_{[5]}$ to the machine (i.e., between $ct_{[4]}^M = t + 3B + p_{[2]} + p_{[3]} + p_{[4]} < t + 3B + B = t + 4B$ and $t + 4B$). As we have mentioned, jobs $J_{[1]}, J_{[2]}, J_{[3]}, J_{[4]}$ are delivered to the depot at times $4B, 5B, 6B$ and $7B$, respectively. If $p_{[2]} + p_{[3]} + p_{[4]} > B$, then $J_{[5]}$ would finish at $ct_{[5]}^M = t + 3B + p_{[2]} + p_{[3]} + p_{[4]} + 3B > t + 3B + B + 3B = t + 7B$, and it would be delivered to the depot at time $ct_{[5]}^D = 9B$, causing one idle AGV cycle ($8B$). The first part of the schedule, for the first five jobs mentioned above, is depicted in Fig. 3

Summing up, the considered schedule is a sequence of $(m+1)$ jobs taken from subset $\{J_{3m+1}, \dots, J_{4m+1}\}$, separated with m groups of exactly three jobs taken from $\{J_1, \dots, J_{3m}\}$, lasting exactly B time units. The whole solution is presented in Fig. 4. For the sake of clarity, jobs within both subsets are renumbered in increasing order in Fig. 4. The specific groups of three jobs from subset $\{J_1, \dots, J_{3m}\}$ determine the solution of the 3-partition problem.

If the answer for 3PP is YES, then obviously we can construct a schedule for DMMP with the depot makespan not exceeding $4B(m+1)$ (cf. Fig. 4). The schedule consists of m groups of three jobs J_k, J_l, J_r ($1 \leq k, l, r \leq 3m$), corresponding to particular subsets A_i ($i = 1 \dots m$) from 3PP solution, preceded and followed (separated) with

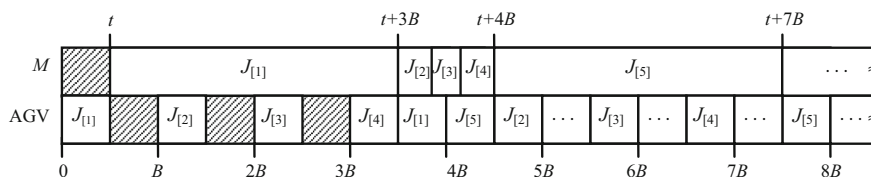


Fig. 3 First five jobs of DMMP schedule

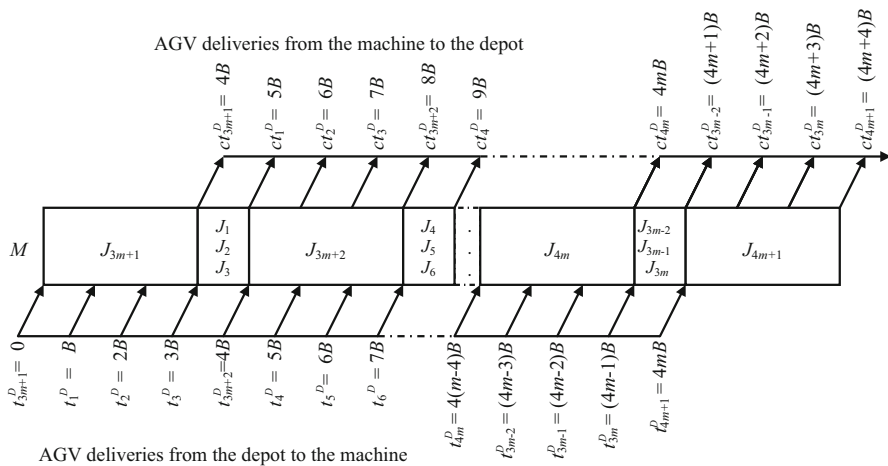


Fig. 4 DMMP schedule

single jobs J_r ($3m+1 \leq r \leq 4m+1$). The depot makespan C_{\max}^D is equal to $4B(m+1)$, and it does not exceed the given threshold C_{\max} . \square

The decision counterpart of the depot makespan minimization problem is strongly NP-complete, so the considered optimization problem is strongly NP-hard. The result holds even for a special case of the problem with equal AGV transportation times from the depot (t_1) and to the depot ($t_2 = t_1 = t$). The case with $t_1 \neq t_2$ is obviously intractable too.

4 Worst-Case Performance Ratio

The analyzed problem of scheduling a set of jobs on a single machine with a cyclic metro transportation system is strongly NP-hard, so there exists no exact algorithm solving it, in either polynomial or pseudo-polynomial time (unless $P \neq NP$). Obviously, we can apply heuristic methods to construct a feasible schedule. List heuristics are basic and commonly used approaches to tackle with hard scheduling problems (e.g., [30]). The analysis of their behavior often delivers useful hints for constructing efficient exact and heuristic methods (e.g., [31, 32]). For the considered problem, any list method is a 2-approximation algorithm. In the following theorems, we will show that the worst-case performance ratio for any list algorithm is equal to 2 and that this bound is asymptotically tight (obviously, excluding exponential time algorithms browsing all possible solutions of the problem).

Theorem 4.1 *For any list algorithm solving the depot makespan minimization problem, the worst-case performance ratio is equal to 2, i.e., $\frac{C_{\max}^D}{C_{\max}^{D*}} \leq 2$, where C_{\max}^D denotes the depot makespan of the list schedule, and C_{\max}^{D*} denotes the optimal depot makespan.*

Proof First, we will specify the lower bound for the depot makespan in the considered system. In any feasible schedule, all n jobs have to be delivered from the depot to the machine, and—after processing—delivered from the machine to the depot. Since the vehicle transports only one job at a time, it needs at least n cycles of length T , i.e., it needs at least nT time units, to transport all jobs. Moreover, since the AGV cannot wait at the machine, it needs one more cycle to deliver the last job to the depot. Thus, from the AGV's point of view, the optimal depot makespan C_{\max}^{D*} cannot be smaller than $nT + T$. On the other hand, executing all jobs requires delivering the first job to the machine, processing all jobs (at least in time $\sum_{j=1}^n p_j$), and delivering the last job from the machine to the depot, where both mentioned deliveries consume T time units. Thus, from the machine's point of view, the optimal depot makespan C_{\max}^{D*} cannot be smaller than $\sum_{j=1}^n p_j + T$. Summing up, the following bound holds:

$$C_{\max}^{D*} \geq \max \left\{ nT + T, \sum_{j=1}^n p_j + T \right\}. \quad (1)$$

Now, we will determine the upper bound of the depot makespan in any list schedule. As we have mentioned, in any feasible schedule, all jobs have to be delivered from the depot to the machine, processed and delivered from the machine back to the depot. During executing job J_j by the machine, the AGV makes at most $\lceil \frac{p_j}{T} \rceil$ cycles of length T . Thus, executing all jobs by the machine requires at most $\sum_{j=1}^n \lceil \frac{p_j}{T} \rceil$ AGV cycles. Taking into account the necessity of delivering the last job from the machine to the depot within one extra AGV cycle, the maximum number of AGV cycles equals $\sum_{j=1}^n \lceil \frac{p_j}{T} \rceil + 1$. Thus, the depot makespan of any list schedule is bounded from above by $(\sum_{j=1}^n \lceil \frac{p_j}{T} \rceil + 1)T$. Since $\lceil \frac{p_j}{T} \rceil \leq \frac{p_j}{T} + 1$, we have $C_{\max}^D \leq (\sum_{j=1}^n \lceil \frac{p_j}{T} \rceil + 1)T \leq (\sum_{j=1}^n (\frac{p_j}{T} + 1) + 1)T = (\sum_{j=1}^n \frac{p_j}{T} + n + 1)T$ and finally

$$C_{\max}^D \leq nT + T + \sum_{j=1}^n p_j. \quad (2)$$

Thus, we can state that

$$\frac{C_{\max}^D}{C_{\max}^{D*}} \leq \frac{nT + T + \sum_{j=1}^n p_j}{\max \left\{ nT + T, \sum_{j=1}^n p_j + T \right\}}. \quad (3)$$

Therefore, if $nT \geq \sum_{j=1}^n p_j$, then

$$\frac{C_{\max}^D}{C_{\max}^{D*}} \leq \frac{nT + T + \sum_{j=1}^n p_j}{nT + T} \leq \frac{nT + T + nT}{nT + T} \leq \frac{2(nT + T)}{nT + T} = 2. \quad (4)$$

Otherwise, if $nT \leq \sum_{j=1}^n p_j$, then

$$\begin{aligned} \frac{C_{\max}^D}{C_{\max}^{D*}} &\leq \frac{nT + T + \sum_{j=1}^n p_j}{\sum_{j=1}^n p_j + T} \\ &\leq \frac{\sum_{j=1}^n p_j + T + \sum_{j=1}^n p_j}{\sum_{j=1}^n p_j + T} \leq \frac{2(T + \sum_{j=1}^n p_j)}{T + \sum_{j=1}^n p_j} = 2. \end{aligned} \quad (5)$$

Summing up, the bound $\frac{C_{\max}^D}{C_{\max}^{D*}} \leq 2$ holds. \square

Theorem 4.2 *The worst-case performance ratio for any list algorithm, $\frac{C_{\max}^D}{C_{\max}^{D*}} \leq 2$, is asymptotically tight.*

Proof We will consider the instance of the depot makespan minimization problem with $n = k^2 + k + 1$ jobs. The set of jobs contains $(k + 1)$ long jobs with $p_j = kT$ for $j = 1 \dots (k + 1)$ and k^2 short jobs with $p_j = \frac{T}{k}$ for $j = k + 2, \dots (k^2 + k + 1)$.

The optimal schedule for this instance is shown in Fig. 5. Since there is no idle time, the optimal depot makespan is equal to

$$C_{\max}^{D*} = t_1 + (k + 1)kT + k^2 \frac{T}{k} + t_2 = k^2T + 2kT + T. \quad (6)$$

Figure 6 shows the list schedule generated according to SPT (shortest processing time) rule. This schedule contains idle times, since the machine has to wait for delivering short jobs with processing times $\frac{T}{k}$. The depot makespan of SPT solution is equal to

$$C_{\max}^D = t_1 + k^2T + (k + 1)kT + t_2 = 2k^2T + kT + T. \quad (7)$$

Comparing both schedules, we have:

$$\lim_{n \rightarrow \infty} \frac{C_{\max}^D}{C_{\max}^{D*}} = \lim_{k \rightarrow \infty} \frac{2k^2T + kT + T}{k^2T + 2kT + T} = 2. \quad (8)$$

\square

Obviously both Theorems 4.1 and 4.2 hold also for a simpler transportation system with $t_1 = t_2 = t$. The general depot makespan minimization problem, with $t_1 \neq 0$ and $t_2 \neq 0$, and its special case with $t_1 = t_2$ are strongly NP-hard, and a list algorithm is 2-approximation heuristic. On the other hand, the special cases, with $t_1 = 0$ or $t_2 = 0$, are polynomially solvable. If $t_1 = 0$, i.e., all jobs are available at the machine at time zero, then the optimal schedule can be constructed by the shortest processing time list algorithm. In the SPT schedule, jobs are completed as soon as possible (the sum of job completion times on the machine is minimum), allowing the AGV for delivering jobs from the machine to the depot. If $t_2 = 0$, i.e., all jobs remain after their processing at the machine, then the optimal schedule can be constructed by the longest processing time list algorithm (LPT). In the LPT schedule, the machine processes longer jobs

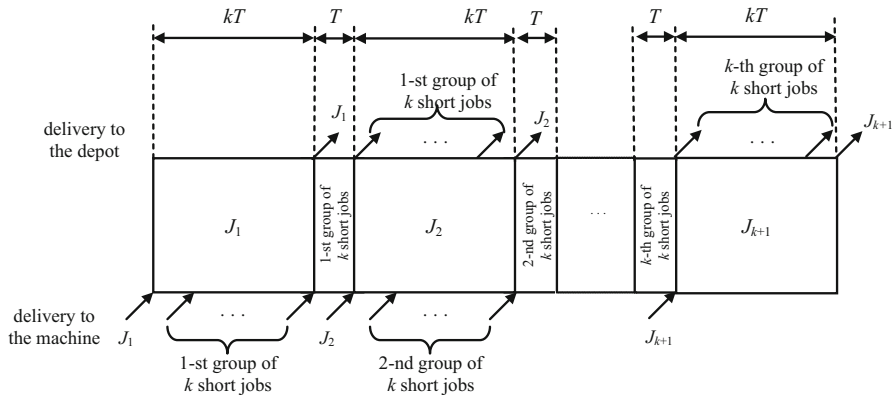


Fig. 5 Optimal machine schedule

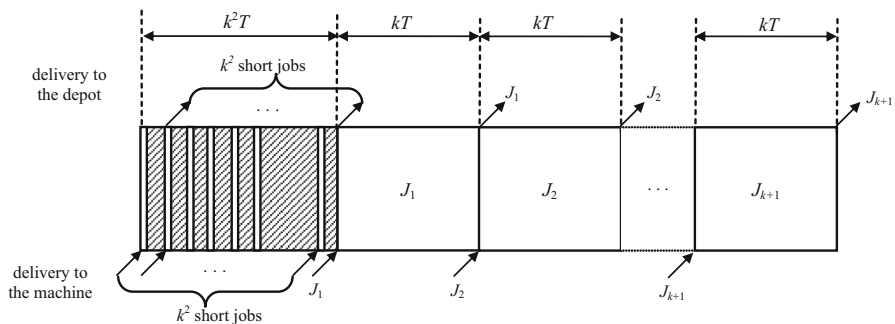


Fig. 6 SPT machine schedule

first, giving the AGV time to deliver as many jobs from the depot to the machine as possible. Obviously, for $t_1 = t_2 = 0$ any schedule is optimal.

5 Conclusions

The paper concerns the simple FMS system consisting of one machine, one depot and one AGV transporting jobs within cycles of the constant length. We proved strong NP-hardness of the problem, and we showed its 2-approximability. Although the literature on flexible manufacturing systems is very rich, such a basic FMS system, as the considered one, has not been studied yet. Proving the complexity status of the problem of minimizing the storage makespan in this FMS is important, since any more complex system must be intractable too.

Within the future research, we would like to take into account other performance measures, or more complicated machine environments as shop systems (cf., e.g., [33]), and use the list heuristics as components of more complex methods such as meta-heuristic (e.g., [31, 34]) or hyperheuristic algorithms (e.g., [35]).

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Muller, T.: Automated guided vehicles. IFS Publications Ltd, Kempston (1983)
2. Blazewicz, J., Pesch, E., Sterna, M., Werner, F.: Total late work criteria for shop scheduling problems. In: Inderfurth, K., Schwodiauer, G., Domschke, W., Juhnke, F., Kleinschmidt, P., Wascher, G. (eds.) Operations research proceedings 1999, pp. 354–359. Springer, Heidelberg (2000)
3. Blazewicz, J., Eiselt, H.A., Finke, G., Laporte, G., Weglarz, J.: Scheduling tasks and vehicles in a flexible manufacturing systems. *Int. J. Flex. Manuf. Syst.* **4**, 5–16 (1991)
4. Blazewicz, J., Pawlak, G., Walter, B.: Scheduling production tasks in a two stage FMS. *Int. J. Prod. Res.* **40**, 4341–4352 (2002)
5. Kats, V., Levner, E.: A fast algorithm for 2-cyclic robotic scheduling with a fixed robot route and interval processing times. *Eur. J. Oper. Res.* **209**, 51–56 (2011)
6. Prakash, A., Chan, F.T.S., Deshmukh, S.G.: FMS scheduling with knowledge based genetic algorithm. *Expert Syst. Appl.* **38**, 3161–3171 (2011)
7. Raman, N., Talbot, F.B., Rachamadugu, R.V.: Simultaneous scheduling of machines and material handling devices in automated manufacturing. In: Stecke, C., Suri, R. (eds.) Proceedings of the second ORSA/TIMS conference on flexible manufacturing system: operations research models and applications, pp. 55–465. Elsevier, Amsterdam (1986)
8. Shin, K.S., Park, J.-O., Kim, Y.K.: Multi-objective FMS process planning with various flexibilities using a symbiotic evolutionary algorithm. *Comput. Oper. Res.* **38**, 702–712 (2011)
9. Sterna, M.: Late work minimization in a small flexible manufacturing system. *Comput. Ind. Eng.* **52**(2), 210–228 (2007)
10. Brauner, N.: Identical part production in cyclic robotic cells - Concepts, overview and open questions. *Discret. Appl. Math.* **156**, 2480–2492 (2008)
11. Steiner, G., Xue, Z.: Scheduling in re-entrant robotic cells: Algorithm and complexity. *J. Sched.* **8**, 25–48 (2005)
12. Blazewicz, J., Pawlak, G., Tanas, M., Wojciechowicz, W.: New algorithms for coupled tasks scheduling—a survey. *RAIRO Oper. Res.* **46**, 335–353 (2012)
13. Co, C.G., Tanchoco, J.M.A.: A review of research on AGVs vehicle management. *Eng. Cost. Prod. Econ.* **21**, 35–42 (1991)
14. Le-Anh, T., De Koster, M.B.M.: A review of design and control of automated guided vehicles systems. *Eur. J. Oper. Res.* **171**, 1–23 (2006)
15. Qiu, L., Hsu, W.-J., Huang, S.-Y., Wang, H.: Scheduling and routing algorithms for AGVs: a survey. *Int. J. Prod. Res.* **40**, 745–760 (2002)
16. Vis, I.F.A.: Survey of research in the design and control of automated guided vehicle systems. *Eur. J. Oper. Res.* **170**, 677–709 (2006)
17. Angeloudis, P., Bell, M.G.H.: An uncertainty-aware AGV assignment algorithm for automated container terminals. *Transp. Res. Part E Logist. Transp. Rev.* **46**, 354–366 (2010)
18. Bak, S., Blazewicz, J., Pawlak, G., Plaza, M., Burke, E.K., Kendall, G.: A parallel branch and bound approach to the rectangular guillotine strip cutting problem. *INFORMS J. Comput.* **23**(1), 15–25 (2011)
19. Drozdowski, M., Kowalski, D., Mizgajski, J., Mokwa, D., Pawlak, G.: Mind the gap: a study of tube tour. *Comput. Oper. Res.* **39**, 2705–2714 (2012)
20. Evers, J.J.M., Koppers, S.A.J.: Automatic guided vehicle traffic control at a container termina. *Transp. Res. A Pol. Pract.* **30**, 21–34 (1996)
21. Rashidi, H., Tsang, E.P.K.: A complete and an incomplete algorithm for automated guided vehicle scheduling in container terminals. *Comput. Math. Appl.* **61**, 630–641 (2011)
22. Blazewicz, J., Finke, G., Espinouse, M.-L., Pawlak, G.: Scheduling vehicles in a cyclic flexible flow-shop. *Eur. J. Autom J Eur des Syst Automa (JESA)* **32**(4), 441–452 (1998)
23. Lacomme, P., Larabi, M., Tchernev, N.: Job-shop based framework for simultaneous scheduling of machines and automated guided vehicles. *Int. J. Prod. Econ.* **143**(1), 24–34 (2013)

24. Nishi, T., Hiranaka, Y., Grossmann, I.E.: A bilevel decomposition algorithm for simultaneous production scheduling and conflict-free routing for automated guided vehicles. *Comput. Oper. Res.* **38**, 876–88 (2011)
25. Ventura, J.A., Rieksts, B.Q.: Optimal location of dwell points in a single loop AGV system with time restrictions on vehicle availability. *Eur. J. Oper. Res.* **192**, 93–104 (2009)
26. Caumont, A., Lacomme, P., Moukrim, A., Tchernev, N.: An MILP for scheduling problems in an FMS with one vehicle. *Eur. J. Oper. Res.* **199**, 706–722 (2009)
27. Lacomme, P., Moukrim, A., Tchernev, N.: Simultaneous job input sequencing and vehicle dispatching in a single-vehicle automated guided vehicle system: a heuristic branch-and-bound approach coupled with a discrete events simulation model. *Int. J. Prod. Res.* **43**, 1911–1942 (2005)
28. Suri, R., Desiraju, R.: Performance analysis of flexible manufacturing systems with a single discrete material-handling device. *Int. J. Flex. Manuf. Sys.* **9**, 223–249 (1997)
29. Garey, M.R., Johnson, D.S.: *Computers and intractability: a guide to the theory of NP-completeness*. W.H. Freeman and Co, San Francisco (1979)
30. Blazewicz, J., Pesch, E., Sterna, M., Werner, F.: Flow shop scheduling with late work criterion—choosing the best solution strategy. *Lect. Notes Comput. Sci.* **3285**, 68–75 (2004)
31. Blazewicz, J., Pesch, E., Sterna, M., Werner, F.: Metaheuristics for late work minimization in two-machine flow shop with common due date. *Lect. Notes Artif. Int.* **3698**, 222–234 (2005)
32. Sterna, M.: Dominance relations for two-machine flow-shop problem with late work criterion. *Bull. Pol. Acad. Sci. Tech.* **55**(1), 59–69 (2007)
33. Blazewicz, J., Pesch, E., Sterna, M., Werner, F.: A note on two-machine job shop with weighted late work criterion. *J. Sched.* **10**(2), 87–95 (2007)
34. Pesch, E., Sterna, M.: Late work minimization in flow shop by a genetic algorithm. *Comput. Ind. Eng.* **57**(4), 120–1209 (2009)
35. Cichowicz, T., Drozdowski, M., Frankiewicz, M., Pawlak, G., Rytwinski, F., Wasilewski, J.: Hyperheuristics for cross-domain search. *Bull. Pol. Acad. Sci. Tech.* **60**(4), 801–808 (2012)