

Online integrated production–distribution scheduling problems without preemption

Xin Feng · Yongxi Cheng · Feifeng Zheng ·
Yinfeng Xu

Published online: 24 February 2015
© Springer Science+Business Media New York 2015

Abstract We study an integrated production–distribution scheduling problem where jobs are released by customers to a manufacturer over time. The jobs are released online, that is, at any time the information of the number, release and processing times of future jobs is unknown, and the processing time of a job becomes known when the job is released. The manufacturer processes the jobs on a single machine. During the processing of jobs preemption is not allowed. Completed jobs are delivered in batches to customers via sufficient capacitated vehicles. For the objective of minimizing the sum of the total delivery time and the total distribution cost, we present a 3-competitive algorithm for the single-customer case and then extend the result to the multi-customer case. A lower bound of two on the competitive ratio of the problem is also given.

Keywords Supply chain scheduling · Integrated production–distribution problems · Online algorithm · Competitive analysis

1 Introduction

In manufacturing supply chain management, production and distribution are two important stages in operation. There has been a lot of literature on integrated

X. Feng · Y. Cheng (✉) · Y. Xu
School of Management, Xi'an Jiaotong University, Xi'an 710049, China
e-mail: chengyx@mail.xjtu.edu.cn

X. Feng
e-mail: fengxin.xjtu@stu.xjtu.edu.cn

Y. Xu
e-mail: yfxu@mail.xjtu.edu.cn

F. Zheng
Glorious Sun School of Business and Management, Donghua University, Shanghai 200051, China
e-mail: ffzheng@dhu.edu.cn

production–distribution scheduling on supply chain management in recent decades. [Hall and Potts \(2003\)](#) pointed out that the essential issue is the coordination of batching and delivery decisions. [Chen and Vairaktarakis \(2005\)](#) considered the objective of minimizing $\alpha S(D_j) + (1 - \alpha)T$, where $S(D_j)$ is a function of delivery times of completed jobs and measures the customer service level, and T denotes the total distribution cost. The constant α ($0 \leq \alpha \leq 1$) represents the relative preference between the customer service level and the total distribution cost. The above literature assumes that all the information of jobs is known at the very beginning, that is considers the offline version of the problem. The reader is referred to [Chen \(2010\)](#) for a comprehensive review on the integrated production–distribution scheduling problem.

In practice, a manufacturer usually makes production and delivery decisions with little information of future jobs. Thus the online version of the integrated production–distribution scheduling problem arises to deal with such scenario. The performance of an online algorithm is generally evaluated by its *competitive ratio* (see e.g., [Borodin and El-Yaniv 1998](#)). In the terminology of our problem, for any job instance I , let $A(I)$ be the objective value of the schedule produced by an online algorithm A , and let $OPT(I)$ be the objective value of an optimal offline schedule for I . Then A is said to be ρ -competitive if $A(I) \leq \rho OPT(I) + \varepsilon$ holds for any job instance I , where $\rho \geq 1$ and ε is a fixed constant. The competitive ratio of algorithm A is the infimum of the set of all values ρ such that A is ρ -competitive.

[Averbakh and Xue \(2007\)](#) studied an online integrated production–distribution scheduling problem where jobs are released over time and the manufacturer processes at most one job at a time. The problem is under a preempt-resume model, that is the processing of a job can be interrupted and then be resumed later from where it was preempted. There are sufficient vehicles of infinite capacity, and the distribution is in *direct* mode such that completed jobs from the same customer can be delivered in one vehicle. For the objective of minimizing the sum of the total flow time and the total delivery cost, they proposed an optimal 2-competitive algorithm for the case with a single customer and a $2m$ -competitive algorithm for the case with m customers. [Averbakh \(2010\)](#) further studied the above problem with capacitated vehicles. Recently, [Averbakh and Baysan \(2013a\)](#) presented a $(3 + \alpha)$ -competitive algorithm for the problem under the preempt-resume model (with capacitated or uncapacitated deliveries), where α is the ratio of the processing time of the longest job to the processing time of the shortest job.

[Averbakh and Baysan \(2012\)](#) investigated a semi-online problem such that a lower bound P for all job processing times is known beforehand, aiming at minimizing the total flow time plus the total delivery cost. They presented a $2D/(D + P)$ -competitive algorithm where D is the cost of a delivery. [Averbakh and Baysan \(2013b\)](#) also studied a semi-online problem for a distribution center where the processing time of an order is assumed to be zero, with the objective to minimize the total delay time plus the total delivery cost. The distribution center is able to foresee the orders to be released in the next S time units. They presented an online algorithm whose competitive ratio is $(4D + S)/(2D + S)$ when $0 < S < 2D$, and is $(S + D)/S$ when $S \geq 2D$.

[Hoogeveen and Vestjens \(2000\)](#) studied a non-preemptive problem to minimize the time by which all jobs have been delivered. They presented an optimal $(\sqrt{5} + 1)/2$ -competitive algorithm. With the same objective function, [van den Akker et al. \(2000\)](#)

studied a preempt-restart problem where an interrupted job needs to be processed later from the scratch, and they presented an optimal $3/2$ -competitive algorithm.

There is also plenty of research on the offline version of the integrated production–distribution scheduling problem. Recently, [Fan et al. \(2015\)](#) studied the problem on a single machine with availability constraint, with the same objective to minimize the sum of the total delivery time and the total distribution cost as in our paper. [Pei et al. \(2014\)](#) and [Lu et al. \(2015\)](#) studied the problem on a serial batch machine, with the objective to minimize the makespan, that is the maximum delivery completion time of the jobs. [Lee \(2015\)](#) studied the problem in which the delivery cost depends on time period for each delivery, subject to the no-wait condition for the finished products, that is the finished products are immediately delivered to the retailer. [Gao et al. \(2015\)](#) studied the problem in which orders are processed and delivered in batches, subject to the no-wait condition between the production and distribution of each batch.

In this paper, we study an online non-preemptive integrated production–distribution scheduling problem, with the objective to minimize the sum of the total delivery time and the total distribution cost. For both single-customer and multi-customer cases, we present competitive online algorithms and lower bounds on the competitive ratio of the problem.

The rest of our paper is organized as follows. In Sect. 2 we formally describe the problem studied. In Sect. 3, we present an online algorithm for the single-customer case and prove that it is 3-competitive, by a novel construction of a schedule for the preemptive version of the problem studied. In Sect. 4, we extend our result to the multi-customer case. The paper is concluded in Sect. 5.

2 Problem description

We study a supply chain scheduling problem including production and distribution stages. Jobs $J = \{J_1, J_2, \dots, J_n\}$ are released by m customers to a manufacturer over time, where the value of n is unknown beforehand. Each job J_j has a release time r_j and a processing time p_j . At any instant, there is no information about the release and processing times of future jobs, parameters of a job become known when the job is released. The manufacturer can only start processing a job after it is released. Assume that in the production stage there is a single machine to process the jobs, and at most one job can be processed at a time. Preemption is not allowed. A job is called a *completed job* after its completion on processing. A schedule regarding the production stage is called a *production schedule*. A production schedule specifies for each job its start time for processing.

In the distribution stage, there are sufficient vehicles with uniform capacity $c > 1$ to deliver completed jobs to customers. That is, at most c jobs can be included in each delivery. As in [Averbakh and Xue \(2007\)](#), transportation time of a delivery is assumed to be zero (nonzero transportation time would increase the objective values of all solutions by the same constant, and so would not change the validity of our results). We assume that each delivery from the manufacturer to a customer k incurs a delivery cost T_k , which depends only on k (i.e., independent of the number of jobs included in the delivery). A schedule regarding the distribution stage is called a *distribution*

schedule. A distribution schedule specifies for each delivery its start time and the jobs included in it. A *full schedule* specifies both the production schedule and the distribution schedule. Hereafter, unless stated otherwise when we say a schedule we mean a full schedule.

For each job J_j , we use C_j to denote its processing completion time, and use d_j to denote its delay time which is the time between its completion of processing and the start of its delivery. The delivery completion time D_j of job J_j , under the assumption that the transportation time of a delivery is zero, is then $D_j = C_j + d_j$. The objective is to minimize $\sum_{j=1}^n D_j + TC$, where TC is the total cost of all deliveries.

We use the five-field notation $\alpha|\beta|\pi|\delta|\gamma$ introduced by [Chen \(2010\)](#) to represent the problem studied, where α , β , π , δ and γ denote machine configuration, job restrictions, delivery characteristics, number of customers, and the objective function respectively. In this paper, we study the following two problems with $m = 1$ customer and with $m > 1$ customers respectively.

$$P_1 : 1|r_j|V(\infty, c), direct|1| \sum D_j + TC.$$

$$P_2 : 1|r_j|V(\infty, c), direct|m| \sum D_j + TC.$$

Here $V(\infty, c)$ means that there are sufficient vehicles and each has a capacity of c , and *direct* means that only jobs released from the same customer are allowed to be included in one delivery.

3 The single-customer problem P_1

In this section, we present a 3-competitive algorithm for Problem P_1 where all jobs are released from a single customer. Since there is only one customer, each delivery has the same transportation cost T . Hence, if there are u deliveries in total, then the total delivery cost $TC = uT$. Before presenting our online algorithm, we first give a lower bound on the competitive ratio for Problem P_1 for any deterministic online algorithm.

Consider the online scheduling problem $1|r_j| \sum C_j$, which is a special case of Problem P_1 with $T = 0$. Hence, the lower bound on the competitive ratio for $1|r_j| \sum C_j$ is also a lower bound for P_1 . [Hoogeveen and Vestjens \(1996\)](#) proved a lower bound of 2 on the competitive ratio for $1|r_j| \sum C_j$, which indicates the following theorem.

Theorem 3.1 *For Problem P_1 , no deterministic online algorithm can be better than 2-competitive.*

3.1 An online algorithm for Problem P_1

[Lu et al. \(2003\)](#) proposed a 2-competitive online algorithm SSPT (Shifted *SPT*, where *SPT* stands for Shortest Processing Time first rule) for the scheduling problem $1|r_j| \sum C_j$, which is a special case of Problem P_1 with $T = 0$. [Averbakh \(2010\)](#) studied a preemptive model

$$1|r_j, pmtn|V(\infty, c), direct|1| \sum F_j + TC,$$

where F_j denotes the flow time of job J_j , that is the time between the release and the delivery of J_j . Averbakh presented an optimal 2-competitive algorithm for this problem. By combining the ideas used in the above two algorithms, we present the following online algorithm for Problem P_1 where preemption is not allowed.

Algorithm A1 In the production stage, schedule jobs by the SSPT algorithm as follows: Reset each job J_j available for processing at time \tilde{r}_j , where \tilde{r}_j is an arbitrary real number within interval $[\max\{r_j, p_j\}, r_j + p_j]$. At any moment the machine becomes available, schedule from the available jobs the one with shortest processing time.

In the distribution stage, one delivery is made as soon as at least one of the following two situations happens, and each delivery takes as many available jobs as possible.

- (a) the number of completed but undelivered jobs is at least c ;
- (b) the total delay time of completed but undelivered jobs is equal to $2T$.

3.2 Analysis of Algorithm A1

For any job instance I of Problem P_1 , we use σ to denote the schedule produced by Algorithm A1 on I . Let σ_p and σ_d be the production schedule and the distribution schedule of σ respectively, and let D_σ and TC_σ be the total delivery time (of all jobs) and the total transportation cost (of all deliveries) of σ respectively. Let S_j be the starting time of job J_j in schedule σ .

Given I and σ , we construct another job instance $I(\sigma)$ as follows. For each job J_j of instance I we define a job, also denoted by J_j , for instance $I(\sigma)$ with the same processing time p_j but with shifted release time $\bar{r}_j = \min\{S_j, 2r_j + p_j\}$. Preemption is allowed for $I(\sigma)$, that is, we may interrupt the processing of any job in $I(\sigma)$ and continue processing it from where it is interrupted at a later moment. We use P'_1 to denote the corresponding preemptive version of Problem P_1 as follows (then $I(\sigma)$ is an instance of Problem P'_1).

$$P'_1 : 1|\bar{r}_j, pmtn|V(\infty, c), direct|1| \sum D_j + TC.$$

The *shortest remaining processing time* (SRPT) rule prescribes to process at each instant the job with the smallest remaining processing time among all released unfinished jobs, which can be implemented online. For the online preemptive scheduling problem $1|\bar{r}_j, pmtn| \sum C_j$, which is a special case of Problem P'_1 with $T = 0$, the SRPT rule gives an optimal solution (Pruhs et al. 2004). The following lemma will be used for later analysis.

Lemma 3.2 (Pruhs et al. 2004) *For Problem P'_1 in the production stage (which is preemptive), at any instant the number of completed jobs in an SRPT production schedule is not less than that in any other production schedule.*

By the same arguments as in Averbakh (2010), for both problems P_1 and P'_1 , when there are c or more undelivered finished jobs, there is no benefit to further delay a delivery (since otherwise the delay times of these jobs are increased without any benefit gained from saving the transportation cost). Thus, we only consider full schedules where there is a delivery as soon as there are c or more finished undelivered jobs, and each delivery takes as many available jobs as possible. However, there may exist deliveries at other times when there are less than c undelivered finished jobs.

Let σ^* be an optimal offline schedule for job instance I of Problem P_1 , and let $OPT(I)$ be the objective value of σ^* . Clearly, for the distribution schedule σ_d^* of σ^* , each delivery is made at the completion time (in the production schedule σ_p^*) of some job in I , and each delivery takes all the currently undelivered finished jobs.

The main ingredient of our analysis is a novel construction of a schedule $\bar{\sigma}$ for each instance $I(\sigma)$ of Problem P'_1 as follows: the production schedule $\bar{\sigma}_p$ of $\bar{\sigma}$ is in SRPT rule, and the distribution schedule $\bar{\sigma}_d$ of $\bar{\sigma}$ has the same configuration as that of σ_d^* for I . That is, assume that there are s deliveries in total in σ_d^* , and they are made at the completion times of the i_1 th, i_2 th, \dots , i_s th ($i_1 < i_2 < \dots < i_s = n$) completed jobs of I in σ_p^* respectively, then, there are also s deliveries in total in $\bar{\sigma}_d$, and they are made respectively at the completion times of the i_1 th, i_2 th, \dots , $i_s (= n)$ th completed jobs of $I(\sigma)$ in $\bar{\sigma}_p$, and each delivery takes all the currently undelivered finished jobs (from the optimality of schedule σ^* , the number of such jobs cannot be larger than c). Thus, for each $h = 1, 2, \dots, s$, the number of jobs in the h th delivery of $\bar{\sigma}_d$ is equal to that of the h th delivery of σ_d^* . Let \bar{F} be the objective value of schedule $\bar{\sigma}$. We have the following lemma.

Lemma 3.3 $\bar{F} + TC_{\bar{\sigma}} \leq 2OPT(I)$.

Proof We construct another schedule σ' for job instance I from the optimal offline schedule σ^* as follows. Let C_j^* be the completion time of job J_j in σ^* . In the production stage of σ' (which is non-preemptive), the completion time of each job J_j is set to be $2C_j^*$. For the distribution stage of σ' , there is a one to one correspondence between deliveries in σ_d' and σ_d^* as follows. In the optimal offline schedule σ^* , each delivery must be made at the completion time of some job included in the delivery (and the job is the last completed job in the delivery), and this also holds for the deliveries in σ' . Each delivery in σ' contains the same set of completed jobs as that in σ^* .

From the construction of σ' , it can be verified that the processing of all jobs in I do not overlap with each other in σ' , and for each job J_j its delivery time in σ' is $2D_j^*$, i.e. twice as its delivery time D_j^* in σ^* . Thus, $D_{\sigma'} = 2D_{\sigma^*}$.

The start time of each job J_j in schedule σ' is $S'_j = 2C_j^* - p_j \geq 2r_j + p_j$, where the inequality is due to $C_j^* \geq r_j + p_j$. Hence, in the corresponding instance $I(\sigma)$ of Problem P'_1 , the release time of job J_j is $\bar{r}_j = \min\{S_j, 2r_j + p_j\} \leq 2r_j + p_j \leq S'_j$. It follows that σ' is a feasible schedule for $I(\sigma)$.

From the construction of schedule $\bar{\sigma}$ for $I(\sigma)$, the two distribution schedules $\bar{\sigma}_d$ and σ_d' have the same configuration. Since the production schedule $\bar{\sigma}_p$ is in SRPT rule, by Lemma 3.2, the i th delivery in $\bar{\sigma}_d$ is not later than the i th delivery in σ_d' for each $i = 1, 2, \dots, s$. Therefore, $D_{\bar{\sigma}} \leq D_{\sigma'} = 2D_{\sigma^*}$. Also, from the construction of $\bar{\sigma}_d$, we have $TC_{\bar{\sigma}} = TC_{\sigma^*}$. Therefore,

$$\bar{F} + TC_{\bar{\sigma}} = D_{\bar{\sigma}} + 2TC_{\bar{\sigma}} \leq 2D_{\sigma^*} + 2TC_{\bar{\sigma}} = 2D_{\sigma^*} + 2TC_{\sigma^*} = 2OPT(I).$$

The lemma is proved. □

Given a full schedule for Problem P_1 , a delivery in the schedule is said to be *unsaturated* if it contains less than c completed jobs, otherwise the delivery is said to be *saturated*. For schedule σ produced by Algorithm A1 on I , we use L_i and l_i to denote the start time of the i th unsaturated delivery and the number of completed jobs in it, respectively. We virtually set $L_0 = 0$. The last delivery in σ is considered to be unsaturated even if it contains c jobs. Assume that there are w unsaturated deliveries in σ in total. Then, L_w is the start time of the last delivery in σ , and from Algorithm A1, we have $L_0 < L_1 < \dots < L_w$, and $l_i \leq c$ for $i = 1, 2, \dots, w$.

If a delivery is started within time interval $(L_{i-1}, L_i]$, we say it is a delivery within $(L_{i-1}, L_i]$ for simplicity, for $i = 1, 2, \dots, w$. For schedule σ , let q_i denote the number of saturated deliveries within $(L_{i-1}, L_i]$, and let $d(i)$ denote the total delay time of the $q_i + 1$ deliveries within $(L_{i-1}, L_i]$ (q_i saturated deliveries and one unsaturated delivery). Here the delay time of a delivery is the sum of the delay times of all jobs in the delivery.

Let $F(I)$ be the objective value of schedule σ produced by Algorithm A1 on I . Next, we prove the following lemma.

Lemma 3.4 $F(I) \leq \frac{3}{2}(\bar{F} + TC_{\bar{\sigma}})$.

Proof In schedule σ , for each $i = 1, 2, \dots, w$, there are q_i saturated deliveries and one unsaturated delivery within $(L_{i-1}, L_i]$, and the $q_i + 1$ deliveries contain $q_i c + l_i$ completed jobs in total. The total transportation cost of schedule σ is

$$TC_{\sigma} = \sum_{i=1}^w (q_i + 1)T.$$

By Algorithm A1, the delay time of each delivery in σ is at most $2T$, and so $d(i) \leq 2(q_i + 1)T$. Hence,

$$\begin{aligned} F(I) &= \sum_{j=1}^n D_j + TC_{\sigma} \\ &= \sum_{i=1}^w \left(\sum_{j \in (L_{i-1}, L_i]} C_j + d(i) \right) + \sum_{i=1}^w (q_i + 1)T \\ &= \sum_{i=1}^w \sum_{j \in (L_{i-1}, L_i]} C_j + \sum_{i=1}^w d(i) + \sum_{i=1}^w (q_i + 1)T \\ &\leq \sum_{i=1}^w \sum_{j \in (L_{i-1}, L_i]} C_j + 3 \sum_{i=1}^w (q_i + 1)T. \end{aligned}$$

The production stage of Algorithm A1 is the same as Algorithm SSPT proposed by Lu et al. (2003). By Theorem 2 in Lu et al. (2003), for any instance I of Problem P_1 , the production schedule σ_p produced by Algorithm A1 on I is an SRPT production schedule for the corresponding instance $I(\sigma)$ of the preemptive problem P'_1 . Thus, by the construction of schedule $\bar{\sigma}$, the production schedules of $\bar{\sigma}$ (for problem instance $I(\sigma)$) and σ (for problem instance I) are the same, that is $\bar{\sigma}_p = \sigma_p$. It follows that for each $i = 1, 2, \dots, w$, the number of jobs completed within $(L_{i-1}, L_i]$ in schedule $\bar{\sigma}$ is the same as that in schedule σ , which is $q_i c + l_i$.

For each $j = 1, 2, \dots, n$, let \bar{C}_j and \bar{D}_j be the completion time and the delivery time of job J_j in $\bar{\sigma}$, respectively. Because $\bar{\sigma}_p = \sigma_p$, we have $\bar{C}_j = C_j$ for each $j = 1, 2, \dots, n$, where C_j is the completion time of job J_j in σ . For each $i = 1, 2, \dots, w$, let \bar{F}_i be the portion of \bar{F} contributed from all the jobs completed and deliveries made within $(L_{i-1}, L_i]$ in schedule $\bar{\sigma}$, and let $TC_{\bar{\sigma}}^i$ be the total cost of all deliveries made within $(L_{i-1}, L_i]$ in schedule $\bar{\sigma}$.

Since the number of jobs completed within $(L_{i-1}, L_i]$ in $\bar{\sigma}$ is $q_i c + l_i$, and $\bar{\sigma}_d$ has the same configuration as that of σ_d^* (the distribution schedule of an optimal schedule σ^* for I), it follows that the number of deliveries made within $(L_{i-1}, L_i]$ in $\bar{\sigma}$ is at least q_i . For each $i = 1, 2, \dots, w$, based on the number of deliveries made within $(L_{i-1}, L_i]$ in $\bar{\sigma}$, there are the following two cases.

Case 1. The number of deliveries made within $(L_{i-1}, L_i]$ in schedule $\bar{\sigma}$ is at least $q_i + 1$. By the above analysis, in this case we have

$$\begin{aligned} \bar{F}_i + TC_{\bar{\sigma}}^i &= \sum_{j \in (L_{i-1}, L_i]} \bar{D}_j + 2TC_{\bar{\sigma}}^i \\ &\geq \sum_{j \in (L_{i-1}, L_i]} \bar{D}_j + 2(q_i + 1)T \\ &\geq \sum_{j \in (L_{i-1}, L_i]} \bar{C}_j + 2(q_i + 1)T \\ &= \sum_{j \in (L_{i-1}, L_i]} C_j + 2(q_i + 1)T. \end{aligned}$$

Case 2. The number of deliveries made within $(L_{i-1}, L_i]$ in schedule $\bar{\sigma}$ is exactly q_i . In this case, from the construction of the distribution schedule of $\bar{\sigma}$, the number of jobs completed within $(L_{i-1}, L_i]$ in $\bar{\sigma}$ must be strictly less than $(q_i + 1)c$. Since $\sigma_p = \bar{\sigma}_p$, it follows that $l_i < c$ (that is, the unsaturated delivery made at time point L_i in σ contains $l_i < c$ jobs). In schedule $\bar{\sigma}$, for the $q_i c + l_i$ jobs completed within $(L_{i-1}, L_i]$, clearly the q_i deliveries within $(L_{i-1}, L_i]$ contain at most $q_i c$ jobs, and so they do not deliver any of the last l_i jobs completed within $(L_{i-1}, L_i]$. Hence, in $\bar{\sigma}$ all the last l_i jobs completed within $(L_{i-1}, L_i]$ are delivered after time L_i .

In schedule σ , since the unsaturated delivery with start time L_i delivers $l_i < c$ jobs (which are the last l_i jobs completed within $(L_{i-1}, L_i]$ in σ), by Algorithm A1 the total delay time of these l_i jobs at time L_i is equal to $2T$. Because $\bar{\sigma}_p = \sigma_p$, it follows that the total delay time of the last l_i jobs completed within $(L_{i-1}, L_i]$ in $\bar{\sigma}$ at time L_i is also equal to $2T$. Hence, the total delay time of these l_i jobs in $\bar{\sigma}$ is larger than

$2T$ (since they are delivered after time L_i in $\bar{\sigma}$). Let \bar{d}_j be the delay time of job J_j in schedule $\bar{\sigma}$. By the above analysis, in this case we have

$$\begin{aligned} \bar{F}_i + TC_{\bar{\sigma}}^i &= \sum_{j \in (L_{i-1}, L_i]} \bar{D}_j + 2TC_{\bar{\sigma}}^i \\ &= \sum_{j \in (L_{i-1}, L_i]} \bar{C}_j + \sum_{j \in (L_{i-1}, L_i]} \bar{d}_j + 2q_i T \\ &> \sum_{j \in (L_{i-1}, L_i]} \bar{C}_j + 2T + 2q_i T \\ &= \sum_{j \in (L_{i-1}, L_i]} C_j + 2(q_i + 1)T. \end{aligned}$$

From the above, in both cases we have

$$\bar{F}_i + TC_{\bar{\sigma}}^i \geq \sum_{j \in (L_{i-1}, L_i]} C_j + 2(q_i + 1)T,$$

for $i = 1, 2, \dots, w$. It follows that

$$\begin{aligned} F(I) &\leq \sum_{i=1}^w \sum_{j \in (L_{i-1}, L_i]} C_j + 3 \sum_{i=1}^w (q_i + 1)T \\ &< \frac{3}{2} \sum_{i=1}^w \left(\sum_{j \in (L_{i-1}, L_i]} C_j + 2(q_i + 1)T \right) \\ &\leq \frac{3}{2} \sum_{i=1}^w \left(\bar{F}_i + TC_{\bar{\sigma}}^i \right) \\ &= \frac{3}{2} \left(\bar{F} + TC_{\bar{\sigma}} \right). \end{aligned}$$

The lemma is proved. □

By combining Lemmas 3.3 and 3.4, we have the following theorem.

Theorem 3.5 *For Problem P_1 , Algorithm A1 is 3-competitive.*

4 The multi-customer problem P_2

In this section, we study Problem P_2 where jobs are released from m different customers. For each $k = 1, 2, \dots, m$, let T_k be the transportation cost for customer k , and let u_k be the total number of deliveries made to customer k , then the total delivery cost $TC = \sum_{k=1}^m u_k T_k$.

For the preempt-resume production model with the objective of minimizing the sum of the total flow time and the total delivery cost, [Averbakh and Xue \(2007\)](#) presented

a $2m$ -competitive algorithm if the vehicles are of unbounded capacity; If the vehicles are of bounded capacity c , Averbakh (2010) presented a 2γ -competitive algorithm with

$$\gamma = \min \left\{ c, 1 + \left(1 - \frac{1}{c} \right) \frac{\sum_{k=1}^m T_k}{T_{\min}} \right\}, \quad (1)$$

where $T_{\min} = \min\{T_k | k = 1, 2, \dots, m\}$. On the contrast, Problem P_2 studied in this section is non-preemptive and has the objective to minimize $\sum D_j + TC$.

4.1 An online algorithm for Problem P_2

We present an online algorithm, denoted by A2, for Problem P_2 . The algorithm is an extension of Algorithm A1 for the single-customer case.

Algorithm A2 In the production stage, schedule all jobs by the SSPT algorithm.

In the distribution stage, completed jobs are delivered to each customer k as soon as either of the following two situations happens, and each delivery takes as many available jobs (to the corresponding customer k) as possible.

- (a) the number of completed but undelivered jobs for customer k is at least c ;
- (b) the total delay time of completed but undelivered jobs for customer k is equal to T_k .

4.2 Analysis of Algorithm A2

Given an instance I of Problem P_2 , we use σ to denote the schedule produced by Algorithm A2 on I , and use σ^* to denote an optimal offline schedule for I . In the same way as in Sect. 3, from I and σ we construct an instance $I(\sigma)$, which is an instance of the preemptive version P'_2 of Problem P_2 as follows.

$$P'_2: \quad 1|\bar{r}_j, pmtn|V(\infty, c), direct|m| \sum D_j + TC.$$

Assume that in the optimal offline schedule σ^* of I , all the deliveries (to the m customers) are made at l distinct time points $\tau_1 < \tau_2 < \dots < \tau_l$. We construct a schedule $\tilde{\sigma}$ for $I(\sigma)$ as follows: the production schedule $\tilde{\sigma}_p$ of $\tilde{\sigma}$ is in SRPT rule for all the jobs from m customers, and in the distribution schedule $\tilde{\sigma}_d$ of $\tilde{\sigma}$, at time points $2\tau_1, 2\tau_2, \dots, 2\tau_l$, all the completed jobs are delivered to their respective customers in the minimum number of shipments, that is, at each time point $2\tau_i$, for each customer there is at most one unsaturated delivery (we will explain later in Lemma 4.1 that at time point $2\tau_l$, all jobs of $I(\sigma)$ are completed in $\tilde{\sigma}_p$, and so it is valid for $2\tau_l$ to be the last delivery time point of $\tilde{\sigma}_d$). Let $OPT(I)$ and \tilde{F} be the objective value of σ^* and $\tilde{\sigma}$ respectively. We have the following lemma.

Lemma 4.1 $\tilde{F} \leq \gamma OPT(I)$, where γ is defined in (1).

Proof We use TC_{σ^*} and $TC_{\tilde{\sigma}}$ to denote the total distribution cost of σ^* and $\tilde{\sigma}$, respectively. We first show that $TC_{\tilde{\sigma}} \leq \gamma TC_{\sigma^*}$. The idea of this part is similar to the proof of Lemma 5 in [Averbakh \(2010\)](#).

Since each delivery in σ^* contains at most c completed jobs, while each delivery in $\tilde{\sigma}$ contains at least one completed job, it follows that

$$TC_{\tilde{\sigma}} \leq cTC_{\sigma^*}. \tag{2}$$

Assume that in schedule σ^* there are in total z_k deliveries for customer k , and in schedule $\tilde{\sigma}$ there are in total m_k^s saturated deliveries and m_k^u unsaturated deliveries for customer k . Then, for $k = 1, 2, \dots, m$,

$$m_k^s c + m_k^u \leq z_k c.$$

From the construction of $\tilde{\sigma}$, at each time point $2\tau_i$ there is at most one unsaturated delivery for each customer k , implying that $m_k^u \leq l$ for $k = 1, 2, \dots, m$. On the other hand, the total number of deliveries in σ^* satisfies that $\sum_{s=1}^m z_s \geq l$. Hence, for $k = 1, 2, \dots, m$,

$$m_k^u \leq \sum_{s=1}^m z_s.$$

By combining the above two inequalities, we have that for $k = 1, 2, \dots, m$,

$$\begin{aligned} m_k^s + m_k^u &= \left(m_k^s + \frac{1}{c}m_k^u\right) + \left(1 - \frac{1}{c}\right)m_k^u \\ &\leq z_k + \left(1 - \frac{1}{c}\right)\sum_{s=1}^m z_s. \end{aligned} \tag{3}$$

Now, the ratio between $TC_{\tilde{\sigma}}$ and TC_{σ^*} can also be bounded from the above as follows

$$\begin{aligned} \frac{TC_{\tilde{\sigma}}}{TC_{\sigma^*}} &= \frac{\sum_{k=1}^m (m_k^s + m_k^u) T_k}{\sum_{k=1}^m z_k T_k} \\ &\leq \frac{\sum_{k=1}^m \left(z_k + \left(1 - \frac{1}{c}\right)\sum_{s=1}^m z_s\right) T_k}{\sum_{k=1}^m z_k T_k} \\ &= 1 + \frac{\left(1 - \frac{1}{c}\right)\sum_{s=1}^m z_s \sum_{k=1}^m T_k}{\sum_{k=1}^m z_k T_k} \\ &\leq 1 + \frac{\left(1 - \frac{1}{c}\right)\sum_{s=1}^m z_s \sum_{k=1}^m T_k}{T_{\min} \sum_{k=1}^m z_k} \\ &= 1 + \left(1 - \frac{1}{c}\right) \frac{\sum_{k=1}^m T_k}{T_{\min}}, \end{aligned} \tag{4}$$

where in the above the first inequality holds by Inequality (3). By the definition of γ and Inequalities (2) and (4), we have

$$TC_{\tilde{\sigma}} \leq \gamma TC_{\sigma^*}. \tag{5}$$

Let D_{σ^*} and $D_{\tilde{\sigma}}$ denote the total delivery time (of all jobs) of σ^* and $\tilde{\sigma}$, respectively. Similar to the proof of Lemma 3.3, we can construct a schedule σ' for I from the optimal offline schedule σ^* , in which each job J_j is completed at time $2C_j^*$, and each delivery in σ' contains the same set of completed jobs as that in σ^* . Similarly as the argument in the proof of Lemma 3.3, we have that the deliveries in σ' are made at time points $2\tau_1, 2\tau_2, \dots, 2\tau_l$, $D_{\sigma'} = 2D_{\sigma^*}$, and σ' is a feasible schedule for $I(\sigma)$ (This also indicates that in schedule σ' at time point $2\tau_l$, all jobs of $I(\sigma)$ are completed. Since $\tilde{\sigma}_p$, the production schedule of $\tilde{\sigma}$, is in SRPT rule, and by the property of the SRPT rule stated in Lemma 3.2, we have that in $\tilde{\sigma}_p$ at time point $2\tau_l$, all jobs of $I(\sigma)$ are also completed, which validates the construction of $\tilde{\sigma}$ in which $2\tau_l$ is the last delivery time point).

Next we show that $D_{\tilde{\sigma}} \leq D_{\sigma'}$. The idea is similar to the proof of Corollary 1 in Averbakh and Xue (2007). For $i = 1, 2, \dots, l$, let f_i' and \tilde{f}_i be the number of jobs delivered by time $2\tau_i$ in σ' and $\tilde{\sigma}$, respectively. Then, $f_l' = \tilde{f}_l = n$. Let $f_0' = \tilde{f}_0 = 0$. We have

$$D_{\sigma'} = \sum_{i=1}^l 2\tau_i (f_i' - f_{i-1}') = 2n\tau_l - \sum_{i=1}^{l-1} 2(\tau_{i+1} - \tau_i) f_i',$$

and

$$D_{\tilde{\sigma}} = \sum_{i=1}^l 2\tau_i (\tilde{f}_i - \tilde{f}_{i-1}) = 2n\tau_l - \sum_{i=1}^{l-1} 2(\tau_{i+1} - \tau_i) \tilde{f}_i.$$

From the constructions of σ' and $\tilde{\sigma}$, and by the property of the SRPT rule stated in Lemma 3.2, we have that $\tilde{f}_i \geq f_i'$ for $i = 1, 2, \dots, l$. Thus, $D_{\tilde{\sigma}} \leq D_{\sigma'}$.

Since $D_{\sigma'} = 2D_{\sigma^*}$, it follows that $D_{\tilde{\sigma}} \leq 2D_{\sigma^*}$. From the definition of γ , it is easy to verify that $c \geq 2$ and $m \geq 2$ imply $\gamma \geq 2$, and so

$$D_{\tilde{\sigma}} \leq \gamma D_{\sigma^*}. \tag{6}$$

By Inequalities (5) and (6), together with $\tilde{F} = D_{\tilde{\sigma}} + TC_{\tilde{\sigma}}$ and $OPT(I) = D_{\sigma^*} + TC_{\sigma^*}$, it follows that $\tilde{F} \leq \gamma OPT(I)$. The lemma is proved. \square

For schedule σ , similarly as the notations defined before Lemma 3.4 in Sect. 3, we use L_i^k to denote the start time of the i th unsaturated delivery for customer k , and use l_i^k to denote the number of completed jobs in it. We virtually set $L_0^k = 0$ for each k . Also, the last delivery for each customer k , is considered to be unsaturated even if it contains c jobs. Let w_k be the total number of unsaturated deliveries for customer k in σ . Let q_i^k denote the number of saturated deliveries made in time interval $(L_{i-1}^k, L_i^k]$

for customer k , and let $d_k(i)$ denote the total delay time of the $q_i^k + 1$ deliveries made within $(L_{i-1}^k, L_i^k]$ for customer k , in schedule σ .

By Algorithm A2, the delay time of each delivery for customer k in σ is at most T_k , it follows that $d_k(i) \leq (q_i^k + 1)T_k$ for $i = 1, 2, \dots, w_k$. Let $F(I)$ be the objective value of schedule σ .

Lemma 4.2 $F(I) \leq 2\tilde{F}$.

Proof For each customer k , let \mathcal{J}_k be the set of jobs in I which are released from customer k . We use $F_k(I)$ (respectively, \tilde{F}_k) to denote the portion of $F(I)$ (respectively, \tilde{F}) contributed from jobs in \mathcal{J}_k and deliveries to customer k , and use TC_σ^k (respectively, TC_σ^k) to denote the total delivery cost for customer k in σ (respectively, $\tilde{\sigma}$).

Then, $F(I) = \sum_{k=1}^m F_k(I)$, and $\tilde{F} = \sum_{k=1}^m \tilde{F}_k$. For each $k = 1, 2, \dots, m$,

$$\begin{aligned}
 F_k(I) &= \sum_{j \in \mathcal{J}_k} D_j + TC_\sigma^k \\
 &= \sum_{i=1}^{w_k} \left(\sum_{j \in (L_{i-1}^k, L_i^k] \cap \mathcal{J}_k} C_j + d_k(i) \right) + \sum_{i=1}^{w_k} (q_i^k + 1) T_k \\
 &\leq \sum_{i=1}^{w_k} \sum_{j \in (L_{i-1}^k, L_i^k] \cap \mathcal{J}_k} C_j + 2 \sum_{i=1}^{w_k} (q_i^k + 1) T_k, \tag{7}
 \end{aligned}$$

where in the above the inequality holds because $d_k(i) \leq (q_i^k + 1)T_k$, for $i = 1, 2, \dots, w_k$.

By the same argument as in the proof of Lemma 3.4, in the production stage Algorithm A2 processes all jobs of I in SSPT schedule, which is the same as the SRPT schedule for the corresponding instance $I(\sigma)$ of the preemptive problem P'_2 . That is, $\tilde{\sigma}_p = \sigma_p$.

Let $TC_\sigma^k(i)$ be the portion of TC_σ^k contributed from all deliveries to customer k made within $(L_{i-1}^k, L_i^k]$. Let \tilde{F}_k^i be the portion of \tilde{F}_k contributed from all jobs in \mathcal{J}_k completed within $(L_{i-1}^k, L_i^k]$, together with all deliveries to customer k made within $(L_{i-1}^k, L_i^k]$. Let \tilde{C}_j (respectively, \tilde{D}_j) be the completion time (respectively, delivery time) of job J_j in $\tilde{\sigma}$, and let C_j be the completion time of job J_j in σ . Since $\tilde{\sigma}_p = \sigma_p$, we have $\tilde{C}_j = C_j$ for $j = 1, 2, \dots, n$.

Next we bound \tilde{F}_k^i from below. For the following analysis, we only consider jobs for customer k unless stated otherwise. For each $i = 1, 2, \dots, w_k$, based on the number of deliveries for customer k made within $(L_{i-1}^k, L_i^k]$ in $\tilde{\sigma}$, there are the following two cases.

Case 1. The number of deliveries for customer k made within $(L_{i-1}^k, L_i^k]$ in schedule $\tilde{\sigma}$ is no less than $q_i^k + 1$. By the above analysis, in this case we have

$$\begin{aligned} \tilde{F}_k^i &= \sum_{j \in (L_{i-1}^k, L_i^k] \cap \mathcal{J}_k} \tilde{D}_j + TC_{\tilde{\sigma}}^k(i) \\ &\geq \sum_{j \in (L_{i-1}^k, L_i^k] \cap \mathcal{J}_k} \tilde{C}_j + (q_i^k + 1)T_k \\ &= \sum_{j \in (L_{i-1}^k, L_i^k] \cap \mathcal{J}_k} C_j + (q_i^k + 1)T_k. \end{aligned}$$

Case 2. The number of deliveries for customer k made within $(L_{i-1}^k, L_i^k]$ in schedule $\tilde{\sigma}$ is no more than q_i^k . We use $q_i^k - s_i^k$ to denote this number, where $0 \leq s_i^k \leq q_i^k$. Since the number of jobs for customer k completed within $(L_{i-1}^k, L_i^k]$ in σ (also in $\tilde{\sigma}$ because $\tilde{\sigma}_p = \sigma_p$) is $q_i^k c + l_i^k$, it follows that in $\tilde{\sigma}$ the number of jobs for customer k completed within $(L_{i-1}^k, L_i^k]$ but delivered after L_i^k is at least $l_i^k + s_i^k c$, and from the construction of $\tilde{\sigma}_d$ these jobs are the last completed ones within $(L_{i-1}^k, L_i^k]$. Let \tilde{d}_j be the delay time of job J_j in $\tilde{\sigma}$. We further divide Case 2 into two subcases:

Case 2.1. The delivery for customer k made at time point L_i^k is not the last delivery for customer k in σ (that is, $i < w_k$). In this case since the delivery is unsaturated, it must have $l_i^k < c$. For the last l_i^k completed jobs for customer k within $(L_{i-1}^k, L_i^k]$ in schedule σ (also in schedule $\tilde{\sigma}$), by Algorithm A2 their total delay time at time point L_i^k is T_k .

For the $s_i^k c$ jobs for customer k completed right before the last l_i^k jobs within $(L_{i-1}^k, L_i^k]$ (which are delivered after L_i^k in $\tilde{\sigma}$ in this case), since $l_i^k < c$, it follows that the total delay time of each c of them at L_i^k is more than T_k , which is the total delay time at L_i^k of the last l_i^k jobs completed within $(L_{i-1}^k, L_i^k]$ for customer k . Hence, at time point L_i^k the total delay time of the jobs for customer k completed within $(L_{i-1}^k, L_i^k]$ but delivered after L_i^k in $\tilde{\sigma}$ is more than $(1 + s_i^k)T_k$. Thus, in this case we have

$$\begin{aligned} \tilde{F}_k^i &= \sum_{j \in (L_{i-1}^k, L_i^k] \cap \mathcal{J}_k} \tilde{D}_j + TC_{\tilde{\sigma}}^k(i) \\ &= \sum_{j \in (L_{i-1}^k, L_i^k] \cap \mathcal{J}_k} (\tilde{C}_j + \tilde{d}_j) + (q_i^k - s_i^k)T_k \\ &= \sum_{j \in (L_{i-1}^k, L_i^k] \cap \mathcal{J}_k} C_j + \sum_{j \in (L_{i-1}^k, L_i^k] \cap \mathcal{J}_k} \tilde{d}_j + (q_i^k - s_i^k)T_k \\ &> \sum_{j \in (L_{i-1}^k, L_i^k] \cap \mathcal{J}_k} C_j + (q_i^k + 1)T_k. \end{aligned}$$

Case 2.2. The delivery for customer k made at time point L_i^k is the last delivery for customer k in σ (that is, $i = w_k$). In this case, we count into \tilde{F}_k^i the deliveries in $\tilde{\sigma}$ for customer k after time L_i^k , and the number of such deliveries is at least $s_i^k + 1$ since no less than $l_i^k + s_i^k c$ jobs are delivered after time L_i^k in $\tilde{\sigma}$. Thus, in this case we have

$$\begin{aligned} \tilde{F}_k^i &\geq \sum_{j \in (L_{i-1}^k, L_i^k] \cap \mathcal{J}_k} \tilde{D}_j + TC_{\tilde{\sigma}}^k(i) + (s_i^k + 1)T_k \\ &= \sum_{j \in (L_{i-1}^k, L_i^k] \cap \mathcal{J}_k} \tilde{D}_j + (q_i^k + 1)T_k \\ &\geq \sum_{j \in (L_{i-1}^k, L_i^k] \cap \mathcal{J}_k} C_j + (q_i^k + 1)T_k. \end{aligned}$$

Hence, in both Case 1 and 2, we have

$$\tilde{F}_k^i \geq \sum_{j \in (L_{i-1}^k, L_i^k] \cap \mathcal{J}_k} C_j + (q_i^k + 1)T_k.$$

Therefore,

$$\tilde{F}_k \geq \sum_{i=1}^{w_k} \tilde{F}_k^i \geq \sum_{i=1}^{w_k} \sum_{j \in (L_{i-1}^k, L_i^k] \cap \mathcal{J}_k} C_j + \sum_{i=1}^{w_k} (q_i^k + 1)T_k. \tag{8}$$

By Inequalities (7) and (8), it follows that $F_k(I) \leq 2\tilde{F}_k$ for each $k = 1, 2, \dots, m$, and so $F(I) \leq 2\tilde{F}$. The lemma is proved. □

By combining Lemmas 4.1 and 4.2, we have the following theorem.

Theorem 4.3 *For Problem P_2 , Algorithm A2 is 2γ -competitive where γ is defined in (1).*

Corollary 4.4 *If delivery costs to different customers are equal, then the competitive ratio of Algorithm A2 is not greater than $2 \min \{c, 1 + (1 - \frac{1}{c})m\}$.*

5 Concluding remarks

In this paper, we investigate an online integrated supply chain scheduling problem without preemption. We consider two cases with a single customer and with multiple customers respectively. The objective is to minimize the sum of the total delivery time and the total distribution cost. For the single-customer case, we present a 3-competitive algorithm and give a lower bound of 2 on the competitive ratio for any deterministic online algorithm; for the multi-customer case, we present a 2γ -competitive algorithm where γ is defined in (1). Clearly, the lower bound of 2 on the competitive ratio for

the single-customer case also applies to the multi-customer case. It is interesting to further tighten the above bounds on the competitive ratio for both cases.

This study was focused on the development of competitive online algorithms. For further research, it also would be interesting to conduct a computational study of the effectiveness of the algorithms proposed. Since the offline versions of Problems P_1 and P_2 both contain the strongly NP-hard problem $1|r_j|\sum C_j$ (Lenstra et al. 1977) as special case, the key for such study is clearly deriving good lower bounds for the offline versions of P_1 and P_2 which can be efficiently computed. The lower bounds used to prove the results in this paper are derived from schedules $\bar{\sigma}$ (in Lemma 3.3) and $\tilde{\sigma}$ (in Lemma 4.1), whose constructions are partially based on the optimal offline schedules for P_1 and P_2 respectively, and so it seems not likely that there is an efficient way to compute them.

Acknowledgments The authors would like to thank the two anonymous reviewers for their helpful comments and suggestions. This work was partially supported by the National Natural Science Foundation of China under Grant Nos. 71071123, 11101326 and 71172189, and the Program for Changjiang Scholars and Innovative Research Team in University (IRT1173).

References

- Averbakh I (2010) Online integrated production-distribution scheduling problems with capacitated deliveries. *Eur J Oper Res* 200:377–384
- Averbakh I, Baysan M (2012) Semi-online two-level supply chain scheduling problems. *J Sched* 15(3):381–390
- Averbakh I, Baysan M (2013a) Approximation algorithm for the on-line multi-customer two-level supply chain scheduling problem. *Oper Res Lett* 41(6):710–714
- Averbakh I, Baysan M (2013b) Batching and delivery in semi-online distribution systems. *Discret Appl Math* 161:28–42
- Averbakh I, Xue Z (2007) Online supply chain scheduling problems with preemption. *Eur J Oper Res* 181:500–504
- Borodin A, El-Yaniv R (1998) Online computation and competitive analysis. Cambridge University Press, Cambridge
- Chen ZL (2010) Integrated production and outbound distribution scheduling: review and extensions. *Oper Res* 58(1):130–148
- Chen ZL, Vairaktarakis GL (2005) Integrated scheduling of production-distribution operations. *Manag Sci* 51(4):614–628
- Fan J, Lu XW, Liu PH (2015) Integrated scheduling of production and delivery on a single machine with availability constraint. *Theor Comput Sci* 562:581–589
- Gao S, Qi L, Lei L (2015) Integrated batch production and distribution scheduling with limited vehicle capacity. *Int J Prod Econ* 160:13–25
- Hall NG, Potts CN (2003) Supply chain scheduling: batching and delivery. *Oper Res* 51(4):566–584
- Hoogeveen JA, Vestjens APA (1996) Optimal on-line algorithms for single-machine scheduling. In: Proceedings 5th international conference on integer programming and combinatorial optimization, Vancouver, British Columbia, Canada, June 3–5 (1996) Lecture Notes in Computer Science 1084. Springer, Berlin, pp 404–414
- Hoogeveen JA, Vestjens APA (2000) A best possible online algorithm for minimizing maximum delivery time on a single machine. *SIAM J Discret Math* 13(1):56–63
- Lee IS (2015) A coordinated scheduling of production-and-delivery under dynamic delivery cost environments. *Comput Ind Eng* 81:22–35
- Lenstra JK, Rinnooy Kan AHG, Brucker P (1977) Complexity of machine scheduling problems. *Ann Discret Math* 1:343–362
- Lu X, Sitters R, Stougie L (2003) A class of online scheduling algorithms to minimize total completion time. *Oper Res Lett* 31(3):232–236

- Lu LF, Zhang LQ, Wan L (2015) Integrated production and delivery scheduling on a serial batch machine to minimize the makespan. *Theor Comput Sci*. doi:[10.1016/j.tcs.2015.01.014](https://doi.org/10.1016/j.tcs.2015.01.014)
- Pei J, Pardalos PM, Liu XB, Fan WJ, Yang SL (2014) Serial batching scheduling of deteriorating jobs in a two-stage supply chain to minimize the makespan. *Eur J Oper Res*. doi:[10.1016/j.ejor.2014.11.034](https://doi.org/10.1016/j.ejor.2014.11.034)
- Pruhs K, Sgall J, Torng E (2004) On-line scheduling. In: Leung Joseph Y-T (ed) *Handbook of scheduling: algorithms, models, and performance analysis*. CRC Press, Florida, pp 15.1–15.41 (Chapter 15)
- van den Akker M, Hoogeveen H, Vakhania N (2000) Restarts can help in the online minimization of the maximum delivery time on a single machine. *J Sched* 3(6):333–341