Fixed-sequence Single Machine Scheduling and Outbound Delivery Problems

Azeddine Cheref, Alessandro Agnetis, Christian Artigues, Jean-Charles Billaut

Fixed-sequence single machine scheduling and outbound delivery problems

Azeddine Cheref [*] Alessandro Agnetis [†] Christian Artigues [‡]
Jean-Charles Billaut [§]

Abstract

In this paper, we consider an integrated production and outbound delivery scheduling problem. In particular, we address the situation in which the scheduling sequence and the delivery sequence are the same and predefined. A set of jobs are processed on a single machine and finished jobs are delivered to the customers by a single capacitated vehicle. Each job has a processing time and transportation times between customers are taken into account. Since the sequence is given, the problem consists to form batches of jobs and our objective is to minimize the sum of the delivery times or general functions of the delivery times. The NP-hardness of the general problem is established and a pseudopolynomial time dynamic programming algorithm is given. Some particular cases are treated, for which complexity proofs or polynomial time algorithms are given.

1 Introduction

This paper deals with a model for coordinating production and delivery schedules. In many production systems, finished products are delivered from the factory to multiple customer locations, warehouses, or distribution centers by delivery vehicles. An increasing amount of research has been devoted, during the last twenty years, to devise integrated models for production and distribution. These models have been largely analyzed and reviewed by [Chen, 2010], who proposed a detailed classification scheme. The models reflect the variety of issues, including systems structure, vehicle/transportation system characteristics, delivery modes, various types of time constraints. In the large majority of the models in the literature, the coordination of production and distribution is achieved through the creation of *batches*, i.e., several parts are shipped together and delivered to their respective destinations during a single trip. When forming batches, one must therefore take into account both production information (such as processing time, release dates etc) and delivery information (such as customer location, time windows etc). Most of the models presented in the literature explicitly take into account transportation times to reach the customers' location, but there are no proper routing decisions, since the number of distinct customers is typically very small. Hence, the focus of the analysis is often on scheduling and batching.

[*] cheref@laas.fr
Université François-Rabelais Tours / CNRS, 64 av. J. Portalis, 37200 Tours, France
LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France
[†] agnetis@dii.unisi.it
Dipartimento di Ingegneria dell'Informazione, Universitá degli Studi di Siena, via Roma 56, 53100 Siena, Italy
[‡] artigues@laas.fr
LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France
[§] jean-charles.billaut@univ-tours.fr
Université François-Rabelais Tours / CNRS, 64 av. J. Portalis, 37200 Tours, France

Many studies consider delivery as a separate step after production, but do not model it in details, e.g. assuming that a sufficiently large number of vehicles is available to deliver the products at any time or assuming that it has only one customer. We briefly discuss some works related to integrated scheduling and delivery decisions. [Li et al., 2005] analyze the joint problem of production sequencing and batch formation, in order to minimize total delivery time, given that delivery is performed by a single vehicle. Total delivery time is a meaningful indicator of the overall efficiency of the delivery process. They show that in general the problem is NP-hard, and then give polynomial time algorithms for the problem with a fixed number of distinct destinations. In [Chen and Lee, 2008], there are various destinations but a batch can only contain jobs of the same destination. [Fan et al., 2015], consider a non availability time interval on the machine, several vehicles without capacity constraint and only transportation costs (no vehicle routing problem associated to the delivery). A single vehicle with a storage area and one or two customers is considered in [Chang and Lee, 2004]. The authors show that the problem is NP-hard for the makespan objective. Other complexity results are provided as well as polynomial time algorithms for special cases. For the same objective function, [Li and Ou, 2005] study the pickup and delivery problem in which a single vehicle travels between the machine and the warehouse, whereas [Wang and Cheng, 2009] study a similar problem in which three different locations and two vehicles are considered. The first vehicle transports unprocessed jobs between the warehouse and the factory and the second one transports finished jobs between the factory and the customer. [Hall et al., 2001] consider the problem in which the delivery dates are fixed in advance. In order to minimize the makespan, [Gao et al., 2015] consider the problem on a single machine scheduling and a unique capacitated vehicle with a no wait constraint, i.e., the batch must be delivered as soon as it is completed. They show that the problem is NP-hard, and then give polynomial time algorithms for the special case with constant travel times.

The cases where the delivery problem is between machines (inbound delivery) are reported by [Lee and Chen, 2001] and [Hurink and Knust, 2001], where the authors consider multiple machines and transportation time between them. [Lee and Chen, 2001] consider also the case where the finished jobs are delivered to one customer location. Some complexity results are given. In [Agnetis et al., 2014], [Agnetis et al., 2015] there are two machines and the transportation is made between these machines, various cases are treated and the objective is to minimize the total transportation cost.

Using the terminology of [Chen, 2010], the models presented in this paper concern *batch delivery with routing*, i.e., orders going to different customers can be delivered together in the same shipment (batch).

However, a distinctive feature of the problems is studied. We consider a fixed sequence of production and delivery, i.e. the jobs must be delivered in the same order in which they are produced. Examples of situations in which the customer sequence is *fixed* are reported by [Armstrong et al., 2008] and [Viergutz and Knust, 2014]. For maximizing the total satisfied demand in a single round trip, the authors consider that the products expire in a constant time after their completion time and that a delivery time window exists for each product. In [Lenté and Kergosien, 2014], the authors search for a batching of jobs where a single capacitated vehicle is used for the delivery. Polynomial time algorithms are proposed for minimizing the makespan, the maximum lateness and the number of tardy jobs but the sum of delivery times is not treated. [Tsirimpas et al., 2008] minimize the overall distance traveled for the single vehicle routing with a predefined customer sequence. This problem can be seen as one of the problems treated in [Lenté and Kergosien, 2014] for the makespan minimization with zero processing times (no scheduling problem). In [Agnetis et al., 2014] and [Agnetis et al., 2015] and [Li and Ou, 2005], special cases with a predefined sequence are addressed and for each of them, polynomial time algorithms are given.

Here we will mainly focus on the problem of deciding how to form batches with a given production sequence (problem P1). The more general problem in which the sequence is not fixed and has to be decided (proved NP-hard in [Li et al., 2005])) is denoted by P2.

Our contributions are as follows. We completely characterize the complexity of Problem P1, showing that when the objective function Z is to minimize the total delivery time, it is NP-hard

in the ordinary sense, and that it can be solved in pseudopolynomial time for any sum-type function of the delivery times. Then, we focus on the special case in which all transportation times are identical (constant travel times), and we show that when Z is the total delivery time minimization, both problems P1 and P2 can be solved in polynomial time. The paper is organized as follows. In Section 2 we present the problem formally and show that the problem is NP-hard when Z is the total delivery time, and that it can be solved in pseudopolynomial time when Z is any sum-type objective function. In Section 3, we study two particular cases that remain NP-hard: the case where preemption is allowed and the case where the number of locations is smaller than the number of jobs. In Section 4 we study two polynomially solvable cases: the case where the number of locations is fixed and the case where travel times are constant. Finally, some conclusions and future research directions are presented in Section 5.

2 Problem definition and complexity

2.1 Problem definition and notation

The problem considered in this paper can be described as follows. A set of n jobs is given and their production sequence is known. Each job J_j, $j = 1, \ldots, n$, requires a certain *processing time* p_j on a single machine, and must be delivered to a certain location site. For the sake of simplicity, when it does not create confusion, we use j to refer to the destination of job J_j. We denote by $t_{i,j}$ the transportation time from destination i to destination j. For analogy with vehicle routing problems, we refer to the manufacturer's location as the *depot*. We use M to denote the depot (manufacturer), hence $t_{M,j} = t_{j,M}$ is the transportation time between the depot and destination j. Unless otherwise specified, we assume that transportation times are symmetric and satisfy the triangle inequality.

Deliveries are carried out by a single *vehicle*. The vehicle loads a certain number of jobs that have been produced and departs towards the corresponding destinations. Once all the jobs have been delivered, it returns to the depot, hence completing a *round trip*. The set of jobs delivered during a single round trip constitutes a *batch*. The capacity c of the vehicle is the maximum number of jobs it can load and hence deliver in a round trip. The jobs must be delivered in the order in which they are produced, hence the production sequence also specifies the sequence in which the customers have to be reached.

The problem consists in deciding a partition of all jobs into *batches*, i.e., a *batching scheme*. Each batch will be routed according to the manufacturing sequence.

In general, the performance of the system depends on all the concurrent decisions: production scheduling, batching and vehicle routing. This requires therefore an *integrated model*, allowing one to coordinate all these aspects. A *solution* to our problem with fixed sequence, is the specification of a batching scheme.

Given a solution, we denote by C_j the *completion time* of job J_j on the single machine, which is also the time at which the job is released for delivery, i.e., the batch including job J_j cannot start before C_j. We denote by D_j the delivery time of J_j, i.e., the time at which the job J_j is delivered at its destination.

The performance measures we consider in this paper depend on such delivery times. In particular, denoting with Z the performance measure, in this paper we consider:

- the *total delivery time*, i.e., $Z = \sum_{j=1}^{n} D_j$
- a *general sum-type performance index*, i.e., $Z = \sum_{j=1}^{n} f_j(D_j)$, where $f_j(D_j)$ is a general, nondecreasing function of D_j, $j = 1, \ldots, n$.

Note that the latter case includes total (weighted) delivery time, total (weighted) tardiness, etc.

We consider the following problem:

Problem $P1(Z)$**.** *Given n jobs of length p_j, $j = 1, \ldots, n$, transportation times $t_{i,j}$ for all i, j, and a sequence σ, find a batching scheme $\mathcal{B}$ such that Z is minimized.*

2.2 Complexity

Since the production sequence is given, and since jobs are delivered to the respective customers in the same given order, we assume that the job sequence is $\sigma = (J_1, J_2, ..., J_n)$. Only travel times $t_{j,j+1}$ are relevant, as well as times $t_{j,M} = t_{M,j}$, representing the travel time between customer j and the manufacturer and vice-versa.

Let us first consider the problem where the objective function is the total delivery time, i.e., problem $P1(\sum_{j=1}^{n} D_j)$. For our purposes, we introduce the following problem.

EVEN-ODD PARTITION (EOP). A set of n pairs of positive integers $(a_1, b_1), (a_2, b_2), \ldots, (a_n, b_n)$ is given, in which, for each i, $a_i > b_i$. Letting $K = \sum_{i=1}^{n}(a_i + b_i)$, is there a partition $(S, \bar{S})$ of the index set $I = \{1, 2, \ldots, n\}$ such that

$$\sum_{i \in S} a_i + \sum_{i \in \bar{S}} b_i = K/2? \tag{1}$$

EOP is NP-hard in the ordinary sense [Garey et al., 1988]. In the following, we will actually use the following slightly modified version of the problem.

MODIFIED EVEN-ODD PARTITION (MEOP). A set of n pairs of positive integers $(a_1, b_1), (a_2, b_2), \ldots, (a_n, b_n)$ is given, in which, for each i, $a_i > b_i$. Letting $Q = \sum_{i=1}^{n}(a_i - b_i)$, is there a partition $(S, \bar{S})$ of the index set $I = \{1, 2, \ldots, n\}$ such that

$$\sum_{i \in S}(a_i - b_i) = Q/2? \tag{2}$$

Note that the two problems are indeed equivalent. In fact, suppose that EOP has a partition $(S, \bar{S})$. The corresponding instance of MEOP also admits the same partition. In fact, subtracting $\sum_{i=1}^{n} b_i = \sum_{i \in S} b_i + \sum_{i \in \bar{S}} b_i$ from both sides of (1), one obtains

$$\sum_{i \in S}(a_i - b_i) = K/2 - \sum_{i=1}^{n} b_i \tag{3}$$

Now, from the definitions of K and Q it turns out that

$$Q = K - 2\sum_{i=1}^{n} b_i$$

and hence (3) is indeed (2). We next show the following result.

Theorem 2.1 *$P1(\sum_{j=1}^{n} D_j)$ is NP-hard.*

Proof. The problem is obviously in NP. Given an instance of MEOP, we build an instance of P1 as follows. There are $3n + 3$ jobs. The processing times of the jobs are defined as follows:

$$\begin{aligned} p_1 = 0, \quad & p_2 = 0, \quad & p_3 = 0 \\ p_{3i+1} = 1, \quad & p_{3i+2} = 1, \quad & p_{3i+3} = 4x_i + b_i - 2 \text{ for all } i = 1, \ldots, n-1, \\ p_{3n+1} = 4x_n + b_n + Q/2, \quad & p_{3n+2} = 0, \quad & p_{3n+3} = 0 \end{aligned}$$

where the x_i are defined as.

$$x_i = (3a_i - 2b_i + 3(n-i)(a_i - b_i))/2 \text{ for all } i = 1 \ldots n \tag{4}$$

and $x_{n+1} = 0$.

In the following, we refer to the set of jobs $(J_{3i+1}, J_{3i+2}, J_{3i+3})$, $i = 0, \ldots, n$, as the *triple* T_{i+1}.

For what concerns the travel times, we let:

- for each $i = 0, 1, \ldots, n-1$, one has

  - $t_{M,3i+1} = t_{3i+1,M} = t_{M,3i+2} = t_{3i+2,M} = t_{M,3i+3} = t_{3i+3,M} = x_{i+1}$,
  - $t_{3i+1,3i+2} = a_{i+1}, t_{3i+2,3i+3} = b_{i+1}, t_{3i+3,3i+4} = x_{i+1} + x_{i+2}$.
- $t_{M,3n+1} = 0, t_{3n+1,M} = 0, t_{M,3n+2} = 0, t_{3n+2,M} = 0, t_{M,3n+3} = 0$.
- $t_{3n+1,3n+2} = 0, t_{3n+2,3n+3} = 0$.

Finally, vehicle capacity is $c = 2$. The problem consists in determining whether a solution exists such that the total delivery time does not exceed

$$f^* = \sum_{i=1}^{n}(3C_{3i} + 7x_i + b_i) + C_{3n+1} + C_{3n+2} + C_{3n+3} - Q/2. \tag{5}$$

For shortness, we call *feasible* a schedule satisfying (5). The proof has the following scheme.

1. We first establish via Lemma 2.2 that if a feasible schedule exists, then there is one having a certain structure, called *triple-oriented*,
2. We analyze some properties of this structure,
3. We show that a triple-oriented schedule of value f^* exists if and only if MEOP is a yes-instance.

Lemma 2.2 *If a feasible schedule exists, then there exists one satisfying the following property: for all $i = 1, \ldots, n$, jobs J_{3i} and J_{3i+1} are NOT in the same batch.*

Proof. Suppose that a feasible schedule exists in which, for a certain i ($1 \leq i \leq n$), jobs J_{3i} and J_{3i+1} are in the same batch. Since $c = 2$, the batch contains no other job. As a consequence, after delivering J_{3i+1}, the vehicle must go back to M in order to load the next jobs and start a new trip. If we denote by τ the start time of the round trip of jobs J_{3i} and J_{3i+1}, job J_{3i} is delivered at time $D_{3i} = \tau + t_{M,3i}$ and job J_{3i+1} is delivered at time $D_{3i+1} = \tau + t_{M,3i} + t_{3i,3i+1}$. Therefore we have $D_{3i} = \tau + x_i$ and $D_{3i+1} = \tau + x_i + (x_i + x_{i+1})$. The vehicle is back at M at time $\tau + 2x_i + 2x_{i+1}$. Now, if we replace this batch with two batches of one job each, the delivery times of both jobs as well as the time at which the vehicle is back at M are unchanged. Therefore, there is an equivalent solution where J_{3i} and J_{3i+1} are not in the same batch. □

We call *triple-oriented* a schedule satisfying Lemma 2.2. The reason of this name is that the schedule is decomposed according to triples. More precisely, since $c = 2$, for each triple $T_{i+1} = (J_{3i+1}, J_{3i+2}, J_{3i+3})$, $i = 0, \ldots, n-1$, A consequence of Lemma 2.2 is that there are *exactly* two batches, and only two possibilities, namely:

- either the first batch is $\{J_{3i+1}, J_{3i+2}\}$ and the second is $\{J_{3i+3}\}$,
- or the first batch is $\{J_{3i+1}\}$ and the second is $\{J_{3i+2}, J_{3i+3}\}$.

We call these two possibilities *option A* and *option B* respectively (see Fig. 1). Namely, let us view option B as the *Base option*, and A as a variant to it.

Round trip length. Letting M_i^A and M_i^B denote the round trip length of the jobs of T_i in the two cases. One has:

$$M_i^A = 4x_i + a_i \tag{6}$$
$$M_i^B = 4x_i + b_i \tag{7}$$

Since $a_i > b_i$, option A implies a longer round trip length than the Base option. The difference between the two (i.e., the additional time with option A with respect to B) is precisely $a_i - b_i$.

Lemma 2.3 *In any triple-oriented schedule, the vehicle is never idle, except possibly before loading J_{3n+1}.*

Proof. Let consider the first triple T_1. The vehicle starts at time 0 (to deliver batch $\{J_1\}$ or $\{J_1, J_2\}$), and is back at time $4x_1 + b_1$ or at time $4x_1 + a_1$. The completion time of T_2 is precisely equal to $C_6 = \sum_{j=1}^{6} p_j = 1 + 1 + 4x_1 + b_1 - 2 = 4x_1 + b_1$. Therefore, as $a_1 > b_1$, the vehicle can immediately start the delivery of the jobs of T_2 as soon as it is back to the depot. For the

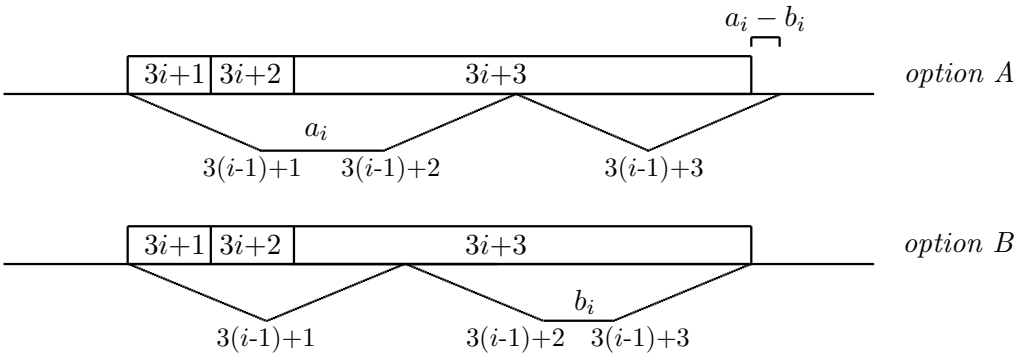

Figure 1: Round trips with options A and B.

same reasons, the delivery of the jobs of T_i cannot be smaller than, the duration of the jobs of T_{i+1}, and the vehicle will be able to start immediately the delivery of the jobs of T_{i+1}. This reasoning does not hold for the last triple T_{n+1} because the duration of J_{3n+1} is different. □

In view of Lemma 2.3, one can compute the total delivery time in the Base scenario, i.e., when option B is *always* chosen. From (7), one has that the vehicle delivering the last 2 jobs of T_i always returns to M exactly at time C_{3i} (see Fig.2). Therefore, the last time the vehicle arrives at M (before delivering the jobs of T_{n+1}) is $C_{3n} + 4x_n + b_n$. Because of the definition of p_{3n+1}, and because J_{3n+1} starts at time C_{3n}, we have $C_{3n} + 4x_n + b_n = C_{3n+1} - Q/2$. In this case, the vehicle will stay idle from $C_{3n+1} - Q/2$ to C_{3n+1}, when job J_{3n+1} can be finally loaded and delivered at time C_{3n+1}. The 3 jobs of T_{n+1} have zero travel times and the last 2 jobs have also zero durations, so all jobs of T_{n+1} can be delivered at $C_{3n+1} = C_{3n+2} = C_{3n+3}$. Finally, recalling that $C_1 = C_2 = C_3 = 0$, the first job of each triple T_i, for $i \geq 1$, is delivered at $C3i + x_i$, the second job is delivered at $C_{3i} + 3x_i$ and the last job at time $C_{3i} + 3x_i + b_i + x_i$, as illustrate in Fig.(2). Hence, we have:

$$f^{BASE} = \sum_{i=1}^{n} (3C_{3i} + 7x_i + b_i) + C_{3n+1} + C_{3n+2} + C_{3n+3} \tag{8}$$

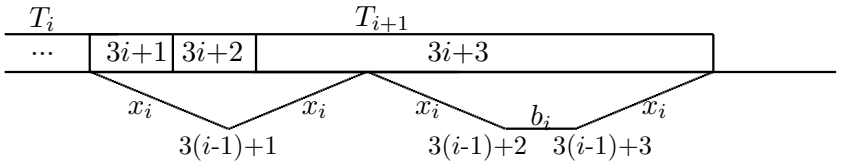

Figure 2: The base schedule (i.e., B is always chosen).

Contribution to total delivery time. Before computing the contribution of a certain triple to the total delivery time, let us consider schedules in which the delivery of the the last three jobs J_{3n+1}, J_{3n+2} and J_{3n+3} start exactly at their release time, i.e., at time $C_{3n+1} = C_{3n+2} = C_{3n+3}$ (options A and B are equivalent). Let us call *regular* a schedule in which such a condition holds.

Expression (8) refers to the scenario in which for all triples, the option B is chosen. We want now to compute the objective function of an arbitrary solution. Let us first consider the contribution of triple T_i to the objective function in the Base schedule, i.e., assuming that *the delivery of T_i started at time C_{3i}*, and let us denote this contribution as TDT_i^A and TDT_i^B

depending of the selected option for T_i.. One has:

$$\begin{aligned} TDT_i^A &= (C_{3i} + x_i) + (C_{3i} + x_i + a_i) + (C_{3i} + 3x_i + a_i) = 3C_{3i} + 5x_i + 2a_i \\ TDT_i^B &= (C_{3i} + x_i) + (C_{3i} + 3x_i) + (C_{3i} + 3x_i + b_i) = 3C_{3i} + 7x_i + b_i \end{aligned}$$

Note that

$$\begin{aligned} TDT_i^B - TDT_i^A &= 2x_i + b_i - 2a_i \\ &= (3a_i - 2b_i + 3(n-i)(a_i - b_i)) + b_i - 2a_i \\ &= a_i - b_i + 3(n-i)(a_i - b_i) \end{aligned} \tag{9}$$

which is positive, remembering that $a_i > b_i$. This means that choosing option A over B brings a benefit in terms of total delivery time. However, such favorable situation for option A is mitigated by the fact that, with option A, one has a longer round trip time than with option B, by the amount $(a_i - b_i)$ (Fig.3). In a regular schedule, such increased round trip time will be "paid" by all subsequent jobs, except the last jobs J_{3n+1}, J_{3n+2} and J_{3n+3}. Hence, in a regular schedule the total effect (in favor of option B) on the subsequent jobs of choosing option A for T_i is given by

$$3(n-i)(a_i - b_i) \tag{10}$$

In conclusion, the *net benefit* of choosing option A over B for T_i in terms of objective function value is obtained subtracting (10) from (9), and in view of the definition of x_i (4), one has therefore that

$$\begin{aligned} NetBenefit_i &= (2x_i + b_i - 2a_i) - 3(n-i)(a_i - b_i) \\ &= a_i - b_i \end{aligned} \tag{11}$$

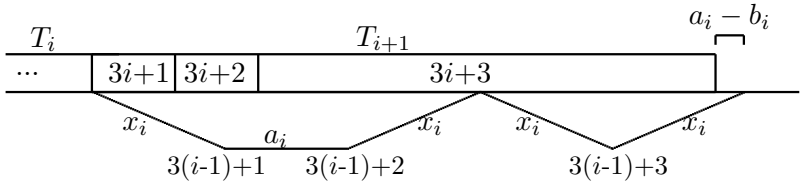

Figure 3: T_i is the first triple choosing option A.

In conclusion, it turns out that, when A is chosen over the Base option, one has a larger round trip time, by $(a_i - b_i)$, but also a smaller contribution to total delivery time (also by the amount $(a_i - b_i)$)(see Fig. 4). So, given any regular triple-oriented schedule in which the last three jobs depart at their completion time, let T_A be the set of triples for which the option A is chosen. Then, from the above considerations, the value f of the objective function is given by

$$f = f^{BASE} - \sum_{i \in T_A} (a_i - b_i) \tag{12}$$

On the other hand, the time at which the vehicle returns to M before loading the last three jobs (J_{3n+1}, J_{3n+2} and J_{3n+3}) is given by

$$C_{3n} + 4x_n + b_n + \sum_{i \in T_A} (a_i - b_i) \tag{13}$$

Now, in a regular schedule the delivery of job J_{3n+1} (and also J_{3n+2} and J_{3n+3}) starts at time $C_{3n+1} = C_{3n} + 4x_n + b_n + Q/2$. Hence, from (13), in a regular schedule, it must hold:

$$\sum_{i \in T_A} (a_i - b_i) \leq Q/2$$

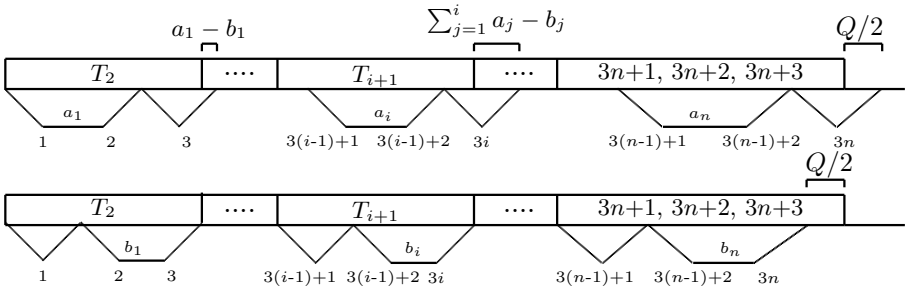

Figure 4: Round trips with option A only and option B only

On the other hand, comparing (5), (8) it turns out that

$$f^* = f^{BASE} - Q/2$$

and hence, from (12), a regular schedule is feasible precisely if a subset T_A of indices exists such that $\sum_{i \in T_A}(a_i - b_i) = Q/2$, i.e., if and only if a feasible partition exists in the instance of EOP. To conclude the proof, it is left to show that f^* can be attained only by a regular schedule. In fact, if a schedule is not regular, the departure time of the last batch is delayed by the amount $(\sum_{i \in T_A}(a_i - b_i) - Q/2)$ with respect to C_{3n+1}. As a consequence, the expression of f in (12) must be modified to take account of such delay of the last three jobs, i.e. it comes

$$f = f^{BASE} - \sum_{i \in T_A}(a_i - b_i) + 3(\sum_{i \in T_A}(a_i - b_i) - Q/2) = f^{BASE} + 2\sum_{i \in T_A}(a_i - b_i) - 3Q/2 \quad (14)$$

Since, in a nonregular schedule,

$$\sum_{i \in T_A}(a_i - b_i) > Q/2,$$

from (14) one has

$$f = f^{BASE} + 2\sum_{i \in T_A}(a_i - b_i) - 3Q/2 > f^{BASE} + Q - 3Q/2 = f^{BASE} - Q/2$$

and hence it cannot be feasible. □

2.3 Pseudopolynomial time algorithm for $P1(\sum_j f_j(D_j))$

Theorem 2.1 implies that no optimal polynomial time algorithm can be found for $P1(\sum_{j=1}^{n} D_j)$, and hence for more general objective functions, unless P=NP. In what follows, we show that $P1(\sum_j f_j(D_j))$ can be solved in pseudopolynomial time, hence settling the complexity status of P1.

We denote by $\{i, j\}$ the batch consisting of jobs $J_i, \ldots, J_j$. As usual, C_j is the completion time of job J_j (known because σ is known), and hence the release time for delivery. We denote by $M(i, j)$ the duration of the round trip of batch $\{i, j\}$, and, if the batch starts at time t, we call $K(i, j, t)$ its contribution to the objective function. Also, we assume that at the beginning, the vehicle is at the manufacturing location.

We denote by $F(i, j, t)$ the value of the optimal solution of the problem restricted to the first j jobs, in which the first job of the last batch is J_i, and such that the delivery of the batch starts at time t. Then, $F(i, j, t)$ can be computed by means of a simple recursive formula. In the optimal solution of the subproblem, if the second last batch is $\{p, i-1\}$, and if it starts at time s, then we have:

$$F(i, j, t) = F(p, i-1, s) + K(i, j, t)$$

Note that, if the vehicle starts at time s, it must be back before or at time t, i.e., the following constraint must hold:

$$C_{i-1} \leq s \leq t - M(p, i-1)$$

In conclusion, the problem is solved by means of:

$$F(i,j,t) = \min_{\substack{\max(i-c,1)\leq p\leq i-1 \\ C_{i-1}\leq s\leq t-M(p,i-1)}} \{F(p,i-1,s)\} + K(i,j,t) \tag{15}$$

Let T be an upper bound on the latest possible departure time for the last batch. As long as the triangle inequality holds, this is given, for instance, by:

$$T = \max\left(\max_{1\leq i\leq n-1}\{C_i + 2\sum_{h=i}^{n-1} t_{hM}\}, C_n\right)$$

The optimal solution is given by:

$$z^* = \min_{n-c+1\leq i\leq n, C_n\leq t\leq T} \big(F(i,n,t)\big)$$

A few boundary conditions must be imposed:

$$F(i,j,t) = +\infty \text{ for all } j < i \tag{16}$$

$$F(1,j,t) = K(1,j,t) \text{ for all } j,t \tag{17}$$

Condition (16) is obvious. Condition (17) allows to initialize the algorithm.

Let us turn to complexity. First, consider the computation of values $M(i,j)$ and $K(i,j,t)$. Both can be simply computed adding the contribution of the next job in the batch either to the round trip time (for $M(i,j)$) or to the objective function (for $K(i,j,t)$). More precisely, the delivery time d_h of job J_h with respect to the departure time of the batch is simply given by:

$$d_h = \begin{cases} d_{h-1} + t_{h-1,h}, & \text{if } i < h \leq j \\ t_{M,h}, & \text{if } h = i \text{ (in this case the vehicle starts from the manufacturer location)} \end{cases}$$

Hence, $M(i,j)$ is simply given by $d_j + t_{j,M}$. Note that $M(i,j+1) = M(i,j) - t_{j,M} + t_{j,j+1} + t_{j+1,M}$. This means that all $M(i,j)$ can be computed in $O(nc)$ assuming $j \leq i+c-1$. Similarly, if batch $\{i,j\}$ starts indeed at time t, the contribution of job J_h to the objective function is given by:

$$f_h(t + d_h), \forall i \leq h \leq j$$

$K(i,j,t)$ is given by $\sum_{h=i}^{j} f_h(t+d_h)$. Again, assuming that $f_j(\cdot)$ can be computed in constant time, note that $d_{j+1} = d_j + t_{j,j+1}$ and $K(i,j+1,t) = K(i,j,t) + f_{j+1}(d_{j+1})$. So, all values $K(i,j,t)$ can be computed in $O(ncT)$.

Once all values $M(i,j)$ and $K(i,j,t)$ are known, one can compute formula (15) for all feasible triples (i,j,t). Each such computation requires comparing nT values. Finally, $O(cT)$ values are compared to find the optimal solution. Since the feasible triples are $O(ncT)$, the computation of all values $F(i,j,t)$ clearly dominates the other phases, and the following result is proved.

Theorem 2.4 *Problem $P1(\sum_j f_j(D_j))$ can be solved in $O(nc^2T^2)$.*

3 More NP-hardness results

In this section we address the special case in which the number of locations is limited for the problem $P1(\sum_j D_j)$, this problem is denoted by $P3(\sum_j D_j)$. We denote by l_j the location of job j and consider that we have K successive locations, i.e., two successive jobs j and $j+1$ are whether in the same location then $l_{j+1} = l_j$ or in the successive locations then $l_{j+1} = l_j + 1$ (figure 5). Note that this problem is NP-hard and is equivalent to $P1(\sum_j D_j)$ if the number of locations

$K = n$. In Sect. 3.1 we completely characterize the complexity of $P3(\sum_j D_j)$, showing that when the objective function is the total delivery time minimisation with the number of locations lower than n, the problem remains NP-hard. For this purpose, we use a similar reasoning to that employed to characterize the complexity of $P1(\sum_j D_j)$. Then we focus in Sect. 3.2 on the case where delivery preemption is allowed, and we establish the NP-hardness of this case.

3.1 Problem $P3(\sum_j D_j)$ with limited number of locations

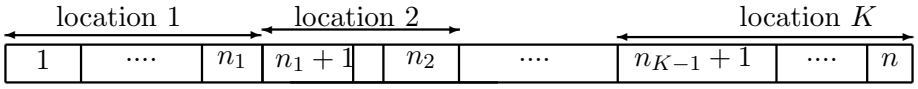

Figure 5: Special case with K locations

Theorem 3.1 *Problem $P3(\sum_j D_j)$ is NP-hard.*

Proof. Given an instance of MEOP, we build an instance of P3 as follows. There are $3(n+1)l$ jobs, $3(n+1)$ locations, l jobs for each location and the vehicle capacity is $c = 2l$. We denote by J^k the sequence of jobs $(J_1^k, J_2^k, \ldots, J_l^k)$ corresponding to location k $(k = 1, \ldots, 3(n+1))$. The job J_j^k is at a position $l(k-1)+j$ in the schedule.

Since the reasoning is similar to that employed to prove the complexity of $P_1(\sum_j D_j)$, we do not recall the global structure of the proof but we rather detail how the reduction used is extended to the limited number of location case.

The processing times of the jobs are defined as follows:

$$
\begin{aligned}
&p_j^1 = 0, p_j^2 = 0, p_j^3 = 0, && \forall j = 1, \ldots, l \\
&p_1^{3i+1} = 1, p_j^{3i+1} = 0, && \forall i = 1, ..., n-1, \forall j = 2, ..., l \\
&p_1^{3i+2} = 1, p_j^{3i+2} = 0, && \forall i = 1, ..., n-1, \forall j = 2, ..., l \\
&p_1^{3i+3} = 4x_i + b_i - 2, p_j^{3i+3} = 0, && \forall i = 1, ..., n-1, \forall j = 2, ..., l \\
&p_1^{3n+1} = 4x_n + b_n + Q/2, p_j^{3n+1} = 0, && \forall j = 2, \ldots, l \\
&p_j^{3n+2} = p_j^{3n+3} = 0, && \forall j = 1, \ldots, l
\end{aligned}
$$

where the x_i are defined by:

$$x_i = \big((a_i - b_i)/l - b_i + 2a_i + 3(n-i)(a_i - b_i)\big)/2, \quad \forall i = 1, \ldots, n \tag{18}$$

and $x_{n+1} = 0$.

Notice that $\sum_{j=1}^{l}(p_j^{3i+1} + p_j^{3i+2} + p_j^{3i+3}) = 4x_i + b_i$, $\forall i = 1, \ldots, n-1$.

The travel times are those used in Sect. 2.2, but now the *triple* T_{i+1}, $i = 0, \ldots, n$, corresponds to the set of sequences of l jobs $(J^{3i+1}, J^{3i+2}, J^{3i+3})$.

Finally, the problem consists in determining whether a solution exists, such that the total delivery time does not exceed

$$f^* = l\sum_{i=1}^{n}(3C_{3li} + 7x_i + b_i) + \sum_{j=1}^{3l} C_{3n+j} - Q/2. \tag{19}$$

For shortness, we call "*feasible*" a schedule satisfying (19).

Lemma 3.2 *If a feasible schedule exists, then it satisfies the following property: for any k, $k = 1, \ldots, 3(n+1)$, all the jobs J_j^k with $j = 1, \ldots, l$ are in the same batch.*

Proof. Suppose that a feasible schedule exists. We denote by k the first location where there exists jobs J_i^k and J_{i+1}^k $(i < l)$ which are not delivered in the same batch. All jobs of the *triple* $T_{\lceil k/3 \rceil}$ – which contains the l jobs of J^k – have been completed when the vehicle starts to deliver the first i jobs of J^k. We distinguish two cases. First, suppose that the capacity of the vehicle allows to take J_{i+1}^k. If we denote by τ the delivery time of J_i^k, job J_{i+1}^k cannot be delivered before time $\tau + 2x_{\lceil k/3 \rceil}$. The total delivery time cannot decrease if J_{i+1}^k is delivered in the same batch as J_i^k at time τ, therefore J_{i+1}^k is assigned to the same batch and the schedule remains feasible. Second, suppose that the capacity of the vehicle does not allow to take J_{i+1}^k. This case is not possible because before location k, the vehicle only delivered multiples of l jobs (either l or $2l$ jobs). So either the vehicle cannot take any job of J^k, or it can take all the jobs of J^k. By applying this reasoning each time there exists a split of jobs that have to be delivered to the same location, we finally obtain a feasible schedule satisfying the lemma. □

In view of Lemma 3.2, we can consider the set of jobs J^k, $k = 1, \ldots, 3(n+1)$, as a single job and the rest of the proof is similar to that given for problem $P1$. □

3.2 Problem $P4(\sum Dj)$ with delivery preemption

In this section, we consider a case where preemption is allowed during the delivery: one can deliver a part of a job, come back to the depot, and take the reminder part of the job for another delivery. We remark that in this case, this problem is equivalent to the NP-hard problem $P1(\sum Dj)$.

Remark 3.3 *The problem $P4(\sum Dj)$ with delivery preemption is NP-hard.*

Proof. Consider that the preemption is allowed on the instance used to prove the complexity of $P1(\sum Dj)$. One can see that each time the vehicle returns to the depot, the number of jobs already completed and ready for delivery is higher than the capacity c. Therefore, delivering a job in two successive trips has no interest and always increases the value of the objective function. □

The result can be extended to the case where the number of sites is fixed to $K < n$.

4 Polynomial cases

In this section, we investigate two particular cases. First, the case where travel times are constant between successive locations. Then, we consider the case where the number of locations is bounded (problem NP-hard, see Section 3.1) and fixed. For both cases, we propose dynamic programming algorithms.

4.1 A particular case: constant travel times

In this section we address the special case in which all travel times are identical. More precisely, we will deal with the following problems:

- problem $P5(\sum_j D_j)$, which consists of problem $P1(\sum_j D_j)$ with constant travel times,
- problem $P6(\sum_j D_j)$, which consists of problem $P2(\sum_j D_j)$ with constant travel times.

For both problems we propose polynomial time algorithms.

4.1.1 Problem $P1(\sum_j D_j)$ with constant travel times

In the following, we investigate the problem of minimizing a general sum-type objective function when the sequence is fixed for both production and delivery and when travel times are constant. We start by analyzing some properties of the optimal solution.

Clearly, every time the vehicle is back at the depot, it can restart immediately with a new batch consisting of jobs already completed, or it can wait for the completion of some jobs to be delivered (see Fig. 6).

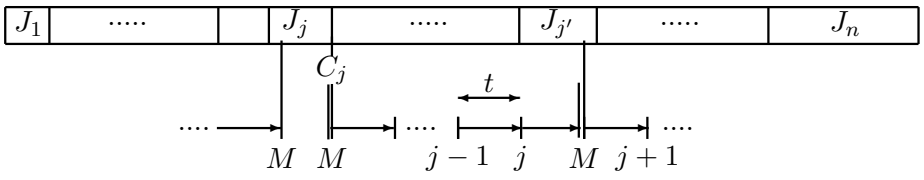

Figure 6: Start of a new tour

Following Li et al.(2005), we call NSS (*Non Stop Shipment*) a sequence of consecutive round trips during which the vehicle is never waiting at the depot (see Fig. 7), followed by a waiting time. We denote by $NSS[i, n_i, j]$ a NSS starting at time C_i, i.e. J_i is the last job of the first round trip of the NSS containing n_i jobs and ending before C_j (when another NSS will start).

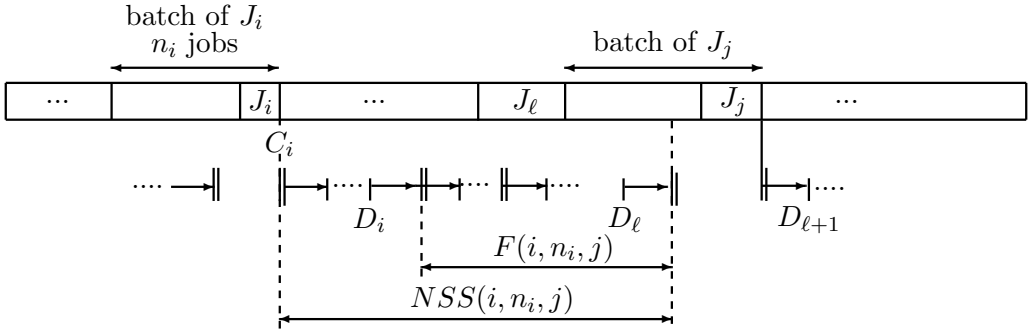

Figure 7: Illustration of a NSS

Suppose that a vehicle starts a round trip at a certain time τ. Let $\mathcal{J}$ be the set of jobs completed before τ (or at time τ) and not delivered. The round trip starting at time τ is called *maximal* if either (i) the batch contains c jobs of $\mathcal{J}$, or (ii) it contains all the jobs of $\mathcal{J}$. The following proposition gives a key feature of the optimal solutions.

Proposition 4.1 *There exists an optimal solution in which all round trips are maximal.*

Proof. Let us denote by R_q the round trip starting at τ. Let $J_i, \ldots, J_j$ be the jobs completed at or before τ and not yet delivered. The round trip R_q is maximal if it contains $\min\{c, j-i+1\}$ jobs. Suppose that R_q is not maximal, i.e. $|R_q| = k - i + 1 < \min\{c, j - i + 1\}$, i.e., the last job in R_q is job J_k, $k \leq j - 1$ (see Fig. 8(a)). This means that job J_{k+1} is delivered in the next batch R_{q+1}. One can move job J_{k+1} from R_{q+1} to R_q because R_q is not maximal. As a consequence, the delivery time of job J_{k+1} decreases by t, without changing the delivery times of *all* the subsequent jobs (see Fig. 8(b)). Hence, the new solution is better than the previous one. Suppose that R_q is maximal, then we are done. One can repeat the whole process for $k + 2, \ldots, \min\{c, j - i + 1\}$, and the proposition follows.

From this, we can see that a solution of the problem is composed by successive NSS. However, a round trip may have to wait for some additional jobs, before starting its route. Let consider the following example.

Example: Let consider an instance with $n = 4$ jobs, processing times equal to $p = (1, 1, 10, 6)$, a travel time equal to 5 and a capacity of 2. The solution without waiting times where the vehicle starts after job J_1 has a total delivery time of 79, the solution without waiting times where the vehicle starts after job J_2 has a total delivery time of 73. The optimal solution consists in waiting for the completion of J_4 and in delivering $\{J_1, J_2\}$ in the same batch and $\{J_3, J_4\}$ in the same batch too.

From the previous properties, one can propose a polynomial time algorithm to build the different NSS.

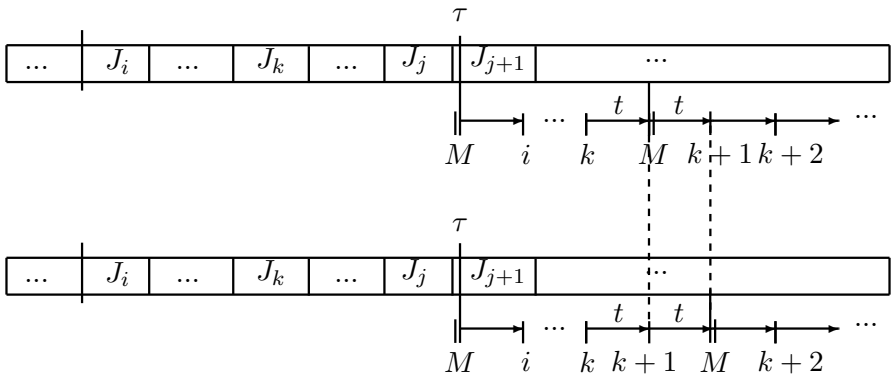

Figure 8: Maximal round trips

We denote by $F(i, n_i, j)$ the contribution to the objective function of the jobs $J_{i+1}, \ldots, J_\ell$ ($j - c \leq \ell < j$) delivered as follows: an NSS denoted $NSS[i, n_i, j]$ starts exactly at time C_i, delivers the n_i jobs of the batch of J_i in the first trip, then the jobs J_{i+1} to J_ℓ in consecutive maximal round trips, where J_ℓ is the last job that can be delivered, so that the vehicle is back at the depot before or at C_j. Notice that once J_i, n_i and J_j are known, J_ℓ is unique and we denote by $\nu_{i,n_i,j}$ the number of undelivered jobs from $J_{\ell+1}$ to J_j, i.e. $\nu_{i,n_i,j} = j - \ell$. At time C_j, another NSS starts in which the first round trip delivers the undelivered jobs $J_{\ell+1}$ to J_j (see Fig. 7). We denote by $B_{i,j}$ the number of round trips of $NSS[i, n_i, j]$. The number of jobs delivered by the round trip number k ($1 \leq k \leq B_{i,j}$) is denoted by n'_k (we have $n'_1 = n_i$). The starting time of the delivery of the first round trip of $NSS[i, n_i, j]$ is equal to C_i. It delivers the jobs in a single tour and J_i is delivered last. The starting time of the delivery of the second round trip of $NSS[i, n_i, j]$ is equal to $C_i + t(n_i + 1)$, of the third round trip is equal to $C_i + t(n_i + 1 + n'_2 + 1)$, the starting time of the round trip number k is equal to $C_i + t\sum_{k'=1}^{k-1}(n'_{k'} + 1)$.

Then, we have:

$$\begin{aligned} F(i, n_i, j) &= \sum_{k=2}^{B_{i,j}} \sum_{r=1}^{n'_k} \left(C_i + t \sum_{k'=1}^{k-1} (n'_{k'} + 1) + rt \right) \\ &= \sum_{k=2}^{B_{i,j}} \left(n'_k \big(C_i + t \sum_{k'=1}^{k-1} (n'_{k'} + 1) \big) + t \frac{n'_k (n'_k + 1)}{2} \right) \end{aligned} \tag{20}$$

When n_i, the number of jobs in the first round trip starting at C_i is known, and according to proposition 4.1, we can construct in polynomial time $NSS[i, n_i, j]$ for any $j > i$, and it follows that $F(i, n_i, j)$ can be calculated in polynomial time.

We define $F(0, n_0, j)$ ($\forall j,\ 1 \leq j \leq c$) as the contribution to the objective function of the first n_0 jobs $J_1, \ldots, J_j$ such that these jobs are delivered in the same first round trip, starting at time C_j. We have:

$$\begin{aligned} F(0, n_0, j) &= \sum_{l=1}^{j} (C_j + lt),\ \forall j, 1 \leq j \leq c, n_0 = j \\ &= \infty,\ \text{otherwise} \end{aligned}$$

Two particular cases are identified, where an NSS cannot exist between two jobs J_i and J_j:

1. $C_i + t(n_i + 1) > C_j$: in this case, the round trip delivering job J_i finishes after the completion time of C_j and thus $NSS[i, n_i, j]$ cannot exist.
2. $\nu_{i,n_i,j} > c$: job J_j cannot be delivered in the first tour of the next NSS starting at time C_j.

In both cases, we set:

$$F(i, n_i, j) = +\infty \tag{21}$$

We define now $f(j, n_j)$ the minimum total cost of the jobs $\{J_1, ..., J_j\}$ where the batch delivering J_j contains n_j jobs and starts at time C_j.

We have:

$$f(j, n_j) = \min_{\substack{0 \leq i < j \\ 1 \leq n_i \leq c \\ \nu_{i,n_i,j} \equiv n_j}} \left\{ f(i, n_i) + F(i, n_i, j) + \sum_{r=1}^{n_j} (C_j + rt) \right\} \tag{22}$$

with

$$\begin{aligned} f(0, n_j) &= 0, \quad \text{if } n_j = 0 \\ &= \infty, \quad \text{otherwise} \end{aligned}$$

Because in the optimal solution, the last round trip does not necessarily start at time C_n, we introduce a dummy job J_{n+1} at the last position with $C_{n+1} = C_n + 2tn$ and $t_{Mn+1} = t$. Clearly, this implies that there exists an optimal solution in which the last round trip delivers only this dummy job at time $D_{n+1} = C_{n+1} + t$.

Finally, the optimal solution is equal to

$$z^* = f(n + 1, 1) - D_{n+1} \tag{23}$$

This dynamic programming algorithm can be implemented in $O(c^2 n^2)$.

Theorem 4.2 *If the travel times are constant, then the problem $P5(\sum_j D_j)$ is solved to optimality with dynamic programming algorithm in polynomial time $O(c^2 n^2)$.*

4.1.2 Problem $P6(\sum_j D_j)$ with constant travel times

In this section, we consider the case where the sequence of jobs is not fixed and travel times are constant.

Proposition 4.3 *There exists an optimal schedule in which the jobs are ordered according to the Shortest Processing Time (SPT) rule.*

Proof. Suppose first that the jobs are numbered according to the SPT rule. It means that $\sigma^{SPT} = \{J_1, J_2, \ldots, J_n\}$. Suppose that the SPT sequence is not optimal, then there exists in the optimal solution σ^* two jobs J_i and J_j with J_i before J_j and $i > j$. We show by using a pairwise interchange reasoning that by maintaining the same round trips, i.e. the same number of jobs and the same starting time of the round trips, σ^{SPT} is such that $\sum D_j^{SPT} \leq \sum D_j^*$ and thus is optimal.

Suppose that J_i and J_j are in the same round trip. The swap of these two jobs, leading to σ' does not change the sum of the delivery dates, and thus $\sum D'_j = \sum D_j^*$. Suppose that these jobs are not in the same round trip. It means that the round trip of J_i terminates with J_i and the round trip of J_j starts with J_j. After the swap of these jobs, the departure times of these two round trips are still feasible, and thus the quality of the solution is unchanged. But putting J_j before J_i may allow to deliver J_j earlier, and then to reduce the total delivery time. We deduce that σ^{SPT} is optimal. □

4.1.3 Problem $P7$ with processing times equal to zero

We consider in this section the case where the processing times of jobs are equal to zero: $p_j = 0$, $\forall j$, $1 \leq j \leq n$. The vehicle has a capacity equal to c and travel times are arbitrary (satisfying the triangular inequality).

One can see that if the vehicle capacity is equal to n, in the optimal solution all the jobs are delivered in a single tour. This solution is called "ideal" and denoted by σ^{id} and the total delivery time is equal to $\sum D_j^{id}$. Clearly, σ^{id} is not feasible when $c < n$. From the ideal solution σ^{id}, a feasible and optimal solution can be obtained in polynomial time.

We define a graph $G = (V, A)$ with $n + 1$ vertices in V $(0 \leq j \leq n)$ where vertex 0 is a dummy job denoted by M (manufacturer site) and vertex j, $1 \leq j \leq n$ corresponds to J_j. There is an edge in A between i and j if $i + 1 \leq j \leq i + c$, corresponding to a round trip with jobs $\{J_{i+1}, ..., J_j\}$ (see Fig. 9). This round trip generates an additional cost to $\sum D_j^{id}$ equal to $t_{j,M} + t_{M,j+1} - t_{j,j+1}$ for each job scheduled after J_j, i.e. $(n - j)$ times. The cost of edge (i, j) is equal to $c_{i,j} = (t_{j,M} + t_{M,j+1} - t_{j,j+1})(n - j)$. Notice that this cost does not depend on J_i.

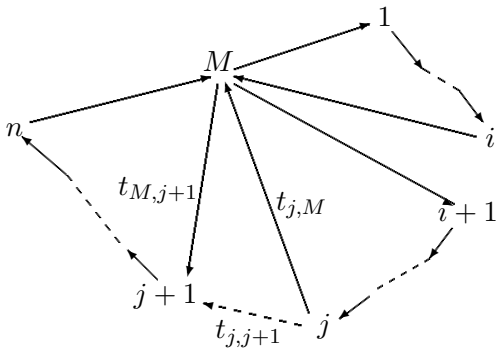

Figure 9: Construction of a feasible and optimal solution from σ^{id}

We define now $f(j)$ the value of the smallest possible degradation until J_j of the "ideal" solution which makes the restricted problem to the j first jobs feasible and optimal. Because there is no edge between (i, j) with $j > i + c$, $f(j)$ is equal to the potential of j in a shortest path in G from 0 to j. Then, we have:

$$f(j) = \min_{i+1 \leq j \leq i+c} \{f(i) + c_{i,j}\} \tag{24}$$

with $f(0) = 0$. The optimal value of the shortest path is given by $f(n)$.

Finally, the optimal solution z^* is equal to:

$$z^* = \sum_{j=1}^{n} D_j^{id} + f(n) \tag{25}$$

This dynamic programming algorithm can be implemented in $O(cn)$. The above reasoning immediately yields the following theorem.

Theorem 4.4 *If the processing times are equal to zero, then the problem $P7(\sum_j D_j)$ is solved to optimality by a dynamic programming algorithm in $O(cn)$ time.*

5 Conclusion

In this paper, we focus on the coordination of a single machine production scheduling problem and a single vehicle delivery problem. The jobs are processed on a single machine and delivered in batches to customers by a single vehicle with limited capacity. The production sequence is

given, and is supposed to be the same as the delivery sequence. The problem is to form batches of jobs and the objective is to minimize the sum of the delivery times. We prove that the problem is NP-hard and propose a pseudopolynomial time dynamic programming algorithm. Some particular cases of the problem are proved to be NP-hard (The limited number of sites and preemptive delivery cases) and polynomial time algorithms are proposed for some more restricted problems.

In the future, we propose to investigate the general case where the sequence is not fixed and has to be determined. We will propose exact algorithms as well as heuristic approaches.

ACKNOWLEDGEMENT

This work was supported by the financial support of the ANR ATHENA project, grant ANR-13-BS02-0006 of the French Agence Nationale de la Recherche.

References

[Agnetis et al., 2014] Agnetis, A., Aloulou, M. A., and Fu, L.-L. (2014). Coordination of production and interstage batch delivery with outsourced distribution. *European Journal of Operational Research*, 238(1):130 – 142.

[Agnetis et al., 2015] Agnetis, A., Aloulou, M. A., Fu, L.-L., and Kovalyov, M. Y. (2015). Two faster algorithms for coordination of production and batch delivery: A note. *European Journal of Operational Research*, 241(3):927 – 930.

[Armstrong et al., 2008] Armstrong, R., Gao, S., and Lei, L. (2008). A zero-inventory production and distribution problem with a fixed customer sequence. *Annals of Operations Research*, 159(1):395–414.

[Chang and Lee, 2004] Chang, Y.-C. and Lee, C.-Y. (2004). Machine scheduling with job delivery coordination. *European Journal of Operational Research*, 158(2):470 – 487. Methodological Foundations of Multi-Criteria Decision Making.

[Chen and Lee, 2008] Chen, B. and Lee, C.-Y. (2008). Logistics scheduling with batching and transportation. *European Journal of Operational Research*, 189(3):871 – 876.

[Chen, 2010] Chen, Z.-L. (2010). Integrated production and outbound distribution scheduling: Review and extensions. *Operations Research*, 58(1):130–148.

[Fan et al., 2015] Fan, J., Lu, X., and Liu, P. (2015). Integrated scheduling of production and delivery on a single machine with availability constraint. *Theoretical Computer Science*, 562:581 – 589.

[Gao et al., 2015] Gao, S., Qi, L., and Lei, L. (2015). Integrated batch production and distribution scheduling with limited vehicle capacity. *International Journal of Production Economics*, 160:13 – 25.

[Garey et al., 1988] Garey, M. R., Tarjan, R. E., and Wilfong, G. T. (1988). One-processor scheduling with symmetric earliness and tardiness penalties. *Mathematics of Operations Research*, 13(2):330–348.

[Hall et al., 2001] Hall, N. G., Lesaoana, M., and Potts, C. N. (2001). Scheduling with fixed delivery dates. *Operations Research*, 49(1):134–144.

[Hurink and Knust, 2001] Hurink, J. and Knust, S. (2001). Makespan minimization for flow-shop problems with transportation times and a single robot. *Discrete Applied Mathematics*, 112(13):199 – 216. Combinatorial Optimization Symposium, Selected Papers.

[Lee and Chen, 2001] Lee, C.-Y. and Chen, Z.-L. (2001). Machine scheduling with transportation considerations. *Journal of Scheduling*, 4(1):3–24.

[Lenté and Kergosien, 2014] Lenté, C. and Kergosien, Y. (2014). Problème de livraison a séquence fixée. In *10ème conférence internationale de modélisation, optimisation et simulation (MOSIM'14), Nancy*.

[Li and Ou, 2005] Li, C.-L. and Ou, J. (2005). Machine scheduling with pickup and delivery. *Naval Research Logistics (NRL)*, 52(7):617–630.

[Li et al., 2005] Li, C.-L., Vairaktarakis, G., and Lee, C.-Y. (2005). Machine scheduling with deliveries to multiple customer locations. *European Journal of Operational Research*, 164(1):39 – 51.

[Tsirimpas et al., 2008] Tsirimpas, P., Tatarakis, A., Minis, I., and Kyriakidis, E. (2008). Single vehicle routing with a predefined customer sequence and multiple depot returns. *European Journal of Operational Research*, 187(2):483 – 495.

[Viergutz and Knust, 2014] Viergutz, C. and Knust, S. (2014). Integrated production and distribution scheduling with lifespan constraints. *Annals of Operations Research*, 213(1):293–318.

[Wang and Cheng, 2009] Wang, X. and Cheng, T. (2009). Production scheduling with supply and delivery considerations to minimize the makespan. *European Journal of Operational Research*, 194(3):743 – 752.