Dynamic traveling salesman problem with stochastic release dates

C. Archetti[(1)] *D. Feillet*[(2)] *A. Mor*[(1)] * *M.G. Speranza*[(1)]

[(1)] *Department of Economics and Management, C.da S. Chiara 50*

University of Brescia, Brescia, Italy

{claudia.archetti, andrea.mor, grazia.speranza}@unibs.it

* Corresponding author

[(2)] *Ecole des Mines de Saint-Etienne and LIMOS UMR CNRS 6158*

CMP Georges Charpak, Gardanne, F-13541 France

feillet@emse.fr

August 14, 2019

Abstract

The dynamic traveling salesman problem with stochastic release dates (DTSP-srd) is a problem in which a supplier has to deliver parcels to its customers. These parcels are delivered to its depot while the distribution is taking place. The arrival time of a parcel to the depot is called its release date. In the DTSP-srd, release dates are stochastic and dynamically updated as the distribution takes place. The objective of the problem is the minimization of the total time needed to serve all customers, given by the sum of the traveling time and the waiting time at the depot. The problem is represented as a Markov Decision Process and is solved through a reoptimization approach. Two models are proposed for the problem to be solved at each stage. The first model is stochastic and exploits the entire probabilistic information available for the release dates. The second model is deterministic and uses an estimation of the release dates. An instance generation procedure is proposed to simulate the evolution of the information to perform computational tests. The results show that a more frequent reoptimization provides better results across all tested instances and that the stochastic model performs better than the deterministic model. The main drawback of the stochastic model lies in the computational time required to evaluate a solution, which makes an iteration of the heuristic substantially more time-consuming than in the case where the deterministic model is used.

Keywords: Routing, traveling salesman problem with release dates, dynamic problem, stochastic information, reoptimization.

1 Introduction

In a last mile distribution system, the supplier often faces the uncertainty related to the delivery of goods to its depot. Information therefore plays a crucial role in an efficient distribution planning. In this setting, we study the dynamic traveling salesman problem with stochastic release dates (DTSP-srd) that is a problem in which a supplier receives from its suppliers goods to be distributed to customers. The goods to be delivered are available for the distribution only after they have arrived to the depot of the supplier. The arrival time of a parcel to the depot is called its release date. In the DTSP-srd, release dates are considered to be stochastic and dynamically updated as the distribution takes place. The DTSP-srd finds application in many real world problems. The steadily growing interest in an environment-aware supply chain management has led to a significant research for an eco-friendly delivery to city centers. In this context, the distribution is implemented by means of distribution centers, where goods are unloaded from the trucks, consolidated, and delivered to the customers by means of hybrid or electric vehicles. The supplier has to face the uncertainty of the arrival times of the trucks to plan for an efficient distribution. The goal is to define an efficient distribution plan serving all customers. The DTSP-srd also finds application in cross docking operations, where the routing of outbound vehicles is planned according to the arrival time of inbound vehicles. In addition, as discussed in Klapp et al. [2018b] and Klapp et al. [2018a], the DTSP-srd finds application in the Same-Day Delivery (SDD) service, where the distribution of parcels, typically ordered online, from a distribution center to final customers, overlaps with their delivery from the suppliers to the distribution center.

While problems with release dates, defined as the earliest availabilities of jobs for processing, have been widely studied in the context of machine sequencing (see Pinedo [2016]), only in recent years the concept of release dates has been introduced for vehicle routing problems. Cattaruzza et al. [2016] introduces the multi-trip vehicle routing problem with time windows and release dates (MTVRPTW-R) in which all information is assumed to be static and deterministic. The authors propose a genetic algorithm for the problem with an underlying giant tour decomposition procedure. The complexity of the traveling salesman problem with release dates (TSP-rd) is studied in Archetti et al. [2015]. The authors introduce two variants of the problem, one considering a deadline for the completion of the distribution and seeking the minimization of the total travel time, referred to as TSP-rd(distance), and the other minimizing the total time required to complete the distribution, referred to as TSP-rd(time). The complexity of the problem is studied on special topologies of the graph. Reyes et al. [2018] extends this work to consider service guarantee, a fixed maximum delay between the release date and the delivery time. The TSP-rd(time) is studied in Archetti et al. [2018] where a mathematical programming formulation is described and an iterated local search is developed. Two versions of the heuristic are tested on instances derived from TSP instances by Solomon [1987] and from the TSPLIB (see

Reinelt [1991]).

Stochasticity has been widely studied in the literature on distribution problems. We refer to Gendreau et al. [2016] for a review of the recent advances and future directions of the stochastic vehicle routing literature and to Ritzinger et al. [2016] for a survey on the dynamic and stochastic vehicle routing problems. Despite this, problems related to the study of uncertain release dates in routing have only recently received scientific interest. Uncertain arrival times are considered in Klapp et al. [2018a], where the authors study the dispatch wave problem. At any stage (wave), requests are either known or potential. The stochasticity of potential requests considers both the probability of non-arrival and the probability of the arrival at a specific wave. The minimization of the expected vehicle operating costs and the penalties for unserved requests is sought on a special topology of the graph, i.e., the line. In Klapp et al. [2018b] the authors extend the work to a general network. A deterministic model is used to find an optimal a priori solution to the stochastic variant and two dynamic policies are developed. Finally, the trade-off between minimizing operational costs and maximizing the total order coverage is studied. Klapp et al. [2018b] is probably the work more closely related to the one presented in this paper. It is therefore worth highlighting the key differences with respect to the problem and the approach presented. In the problem discussed by these authors, probabilistic information is available for unknown customers to describe if and when a customer request will be placed. The distribution takes place over a finite planning horizon. Customers can be rejected or outsourced at the end of the planning horizon at the price of a penalty. The planning horizon is divided in a fixed and finite number of dispatch times, called waves, in which decisions are made on the customers to be served. In the problem discussed in this paper, in contrast, customers are known and probabilistic information is available on their release dates. Customers cannot be rejected and, while the objective is to minimize the completion time, the planning horizon is unbounded. A key difference in the approach to the problem is the fact that stages do not happen at fixed times but are dinamically set according to the information available for the release dates.

A related problem considering a fleet of vehicles is discussed in Voccia et al. [2017], where the same-day delivery problem is studied. Customer requests are described by time windows or delivery deadlines. The arrival rate and geographical distribution of customer requests are known. The authors compute the time a vehicle can wait at the depot without decreasing the ability to serve customer requests and propose a consensus function to identify when it is beneficial to wait at the depot. An approximate forward dynamic programming approach is presented. The same-day delivery problem is also discussed in Ulmer et al. [2016]. The authors investigate the case where preemptive returns are allowed, that is, the vehicles can return to the depot before their planned route is completed to load the parcels of new customers. A study on the dispatching problem in urban consolidation centers is presented in van Heeswijk et al. [2017], where the delivery dispatch problem with time windows is discussed. The authors consider the case where orders are stochastic

and dynamically revealed to the operator, either at the time of arrival at the distribution center or in advance, over a finite planning horizon. At each time the consolidation center must choose a subset of orders to be dispatched. Each customer request is characterized by a time window for the time of dispatch from the distribution center. The aim of the center is to find the cost-minimizing consolidation policy. As the focus of the paper is on the dispatching decisions, routing decisions are not considered as part of the optimization problem and routing costs are estimated by means of a cost function.

The contributions of the current work are the following:

1. We introduce the DTSP-srd as a dynamic and stochastic variant of the TSP-rd(time) where the information on the release dates is assumed to be stochastic and dynamically updated as the distribution takes place. A Markov Decision Process (MDP) is presented to model the dynamic and stochastic aspects of the problem.

2. A heuristic reoptimization approach is proposed. In particular, three strategies with increasing reoptimization frequency are introduced.

3. Two models for solving the problem at each stage of the reoptimization are proposed. The first model is stochastic and exploits the entire probabilistic information available for the release dates. The second model is deterministic and uses a point estimation of the release dates. Both models are solved with an iterated local search.

4. A myopic solution approach is proposed, serving customers as soon as their parcels arrive to the depot, without considering the information available about future release dates.

5. An instance generation procedure is defined simulating the evolution of the information about the release dates by mimicking the arrival of the vehicle to the depot.

6. An exhaustive computational study is presented to show the value of a higher reoptimization frequency and the performance of the deterministic and the stochastic model. The results, obtained on instances with 50 customers, show that a more frequent reoptimization provides better results across all tested instances. The deterministic and stochastic model have an average percentage gap from the best solution found across the three dynamic strategies of 2.92% and 1.16%, respectively. While the stochastic model performs better, its main drawback lies in the computational time required to evaluate any of the solution explored, which makes an iteration of the algorithm substantially more time-consuming. The myopic approach is shown to perform more than 12% worse than the best solution found by the two models.

The paper is organized as follows. In Section 2 the DTSP-srd is defined and modeled as an MDP. The reoptimization approach proposed for the solution of the DTSP-srd is introduced in Section 3 as well as the myopic solution approach. In Section 4 the stochastic and deterministic optimization models, to be solved at each reoptimization epoch, are

introduced, an illustrative example of the two models is presented and the heuristic is described. The instance generation procedure is described in Section 5 and the computational experiments are reported in Section 6. Finally, conclusions are drawn in Section 7.

2 The dynamic traveling salesman problem with stochastic release dates

The DTSP-srd is defined as follows. Let $G = (V, A)$ be a complete graph. A traveling time and a cost are associated with each arc $(i, j) \in A$. These two values are assumed identical and denoted by d_{ij}. It is also assumed that the triangle inequality is satisfied. The set of vertices V is composed by vertex 0, which identifies the depot, and the set N of customers, with $|N| = n$. Each customer is characterized by the arrival time of its parcel to the depot. We call this value its release date. A single vehicle is allowed to perform a sequence of tours to deliver the goods to the customers. Capacity constraints are not considered. We call *route* a tour starting and ending at the depot and not visiting the depot in between. $K = \{1, \ldots, |K|\}$ is the set of routes. The objective is to minimize the expected completion time to serve all customers, that is, the total travel time plus the expected waiting time at the depot. The waiting time is due to the fact that the vehicle has to wait at the depot until the latest release date of the customers that it is going to serve in the next route. Time is assumed to be discrete, i.e., $t \in \mathbb{N}$. While this is a modeling simplification, a sufficiently fine granularity of time should not hinder the quality of the conclusions. We assume that $t = 0$ is the starting of the distribution plan.

As the name implies, the DTSP-srd is characterized by a stochastic and a dynamic aspect. The stochastic aspect is found in the fact that the release dates of the parcels that have not yet been delivered to the depot are assumed to be stochastic. This represents the situation where the vehicle carrying the parcel to the depot is still en route and might encounter unforeseen situations, such as lighter or heavier traffic than expected at the beginning of the distribution. The dynamic aspect of the problem lies in the fact that the information available for each release date evolves over time. On one side, updated information about the stochastic release dates becomes available, on the other, parcels are delivered to the depot.

At any time t, as the distribution is being carried out, some customers might have already been served, that is their parcels have already been delivered to the depot and loaded on the vehicle that has left from the depot. We call $N_t^{served} \subseteq N$ the set of such customers. Note that $N_0^{served} = \varnothing$.

Unserved customers, i.e., customers in $N \setminus N_t^{served}$, are classified based on the information available for their release date. We call $N_t^{unserved}$ the set of these customers. If the parcel for customer has already been delivered to the depot, its release date is past and certain

and the customer can be served. We call $N_t^{known} \subseteq N_t^{unserved}$ the set of these customers. If the parcel is still to be delivered to the depot, the release date occurs in the future and so is stochastic. We call N_t^{future} the set of these customers. Two types of customers in N_t^{future} are defined, according to how such stochasticity is updated over time. The first type are those customers whose stochastic information about the release date is updated only when the vehicle reaches the depot. This can be seen as the case of non-GPS-equipped or non-monitored vehicles: as no new information is available about the state of the vehicle, the information about the release date is not updated until the parcel is delivered to the depot. We call $N_t^{static} \subseteq N_t^{future}$ the set of these customers. The second type of customers are those whose information about the stochastic release date is refined over time. This is the case of parcels delivered to the depot by means of GPS-equipped vehicles, transmitting updated information about their arrival time as they travel. We call $N_t^{dynamic} \subseteq N_t^{future}$ this set of customers. Note that, at any time, $N_t^{future} = N_t^{static} \cup N_t^{dynamic}$, $\{N_t^{known}, N_t^{future}\}$ is a partition of $N_t^{unserved}$ and $\{N_t^{unserved}, N_t^{served}\}$ is a partition of N.

The vehicles delivering the parcels to the depot are assumed to travel independently from one another, i.e., the release dates are assumed to be independent.

The release date of customer $i \in N_t^{unserved}$ at time t is denoted by $\tilde{r}_i^t$. Based on the type of customer, $\tilde{r}_i^t$ is defined according to the following:

- if $i \in N_t^{known}$, $\tilde{r}_i^t$ is the known release date, $\tilde{r}_i^t = r_i$, where r_i is the observed arrival time of the parcel for customer i, i.e., $r_i \leq t$. At time $t = 0$, the set N_t^{known} is composed by those customers whose parcel arrived before the starting time of the distribution plan, e.g., overnight, and $r_i = 0$;
- if $i \in N_t^{static}$, $\tilde{r}_i^t$ is a random variable describing the release date with the information (i.e. probability distribution) available at the beginning of the distribution, $\tilde{r}_i^t = \tilde{r}_i^0$, $\forall t > 0$, i.e., an estimation of the release date is done at time $t = 0$ and never updated as no newer information is available before the parcel reaches the depot;
- if $i \in N_t^{dynamic}$, $\tilde{r}_i^t$ is a random variable describing the release date at time t. In this case $\tilde{r}_i^t$ is dynamically updated with the information available at time t.

The random variables are assumed to be discrete and bounded, with the lower and upper bound of $\tilde{r}_i^t$ denoted as l_i^t and u_i^t, respectively. As the random variable represents a future release date, the lower bound is assumed to be greater than the time t, i.e, $l_i^t \geq t$.

The vehicle delivering parcels to customers can depart at any time t to serve a subset of customers in N_t^{known}. Once the vehicle departs from the depot, its route cannot be modified, that is, customers whose parcel has been loaded cannot be skipped. This excludes preemptive returns to the depot. While this is a modeling restriction, considering probabilistic information when planning the routes of the vehicles makes the benefits of al-

lowing preemptive returns marginal, as discussed in Klapp et al. [2018b]. This is especially true when decisions about route planning are revised with sufficient frequency and in an adaptive way with respect to the evolution of the distribution of the parcels to customers.

We model the problem as an MDP (see Powell et al. [2012], Puterman [2014] for a definition of MDPs and their components, and Secomandi and Margot [2009] for an application of MDPs to routing problems). The components of the MDP proposed for the DTSP-srd are described in the following.

Stages

At each stage a decision is taken based on the information available on the customers to be served. We denote a stage as $e \in \{0, \dots, E\}$ and the time in which stage e takes place as t_e. As preemptive returns are not allowed once the vehicle departs from the depot, that is, new information cannot modify the route of the vehicle, stages are defined in those times in which the vehicle is at the depot. The initial stage 0 happens at the beginning of the distribution, that is, $t_0 = 0$, and the final stage E happens at the end of the distribution, when the vehicle returns to the depot at the end of its final route. As the problem involves a finite number of customers with stochastic but bounded release dates, the number of stages is finite.

States

The state is defined as the information necessary to describe the process at any stage. We therefore describe the state of the process by the time of the stage, the set of customers with known (past) release date and the stochastic information available about the future release dates. Formally, the state of the process at stage e is denoted as:

$$s_e = (t_e, N_{t_e}^{known}, \{\tilde{r}_i^{t_e}\}_{i \in N_{t_e}^{future}})$$

where t_e is the time at which stage e takes place, $N_{t_e}^{known}$ is the set of customers with known release date at time t_e, and $\{\tilde{r}_i^{t_e}\}_{i \in N_{t_e}^{future}}$ is the information available for parcels that have still to reach the depot.

In the initial state $s_0 = (0, N_0^{known}, \{\tilde{r}_i^0\}_{i \in N_0^{future}})$ no customer has been served, that is, $N_0^{served} = \varnothing$ and $N_0^{known} \cup N_0^{future} = N$. At the final stage E, the process is at the terminal state s_E, where the vehicle has returned to the depot after its final route and no customer remains to be served: $s_E = (t_E, \varnothing, \varnothing)$.

Actions

At any stage, a decision is taken based on the state of the process. We call this decision an action. In the DTSP-srd, an action consists of the next route of the vehicle, that is, which subset of the unserved customers to serve and the order of visit of the selected customers. It must be highlighted that the action considers the entire set of customers that have still to be served, $N_{t_e}^{unserved}$, rather than those that can be served at time t_e, $N_{t_e}^{known}$. This models the fact that, at any stage, the best action might be to wait for parcels that are still en route to the depot before serving the next set of customers. In fact, if the chosen route includes some customers in $N_{t_e}^{future}$ then the vehicle will wait at the depot for the parcels of

these customers to arrive before leaving. In this case, it also implies that the starting time of the route is unknown, while the route will start at time t_e otherwise. It also means that the action might never be executed if another stage happens before the route is started and if a different action is taken at this stage. The action space at stage e is therefore defined as the set of all possible routes serving a subset of the unserved customers and is denoted as $X(s_e) = \Gamma_{N_{t_e}^{unserved}}$, where $\Gamma_{N_{t_e}^{unserved}}$ is the set of all possible routes serving a subset of the nodes in $N_{t_e}^{unserved}$. The action selected at stage e is denoted as $\chi_e \in X(s_e)$.

Transitions

Once an action is chosen from the action space, the process transition to a new state. The transition defines how the state is updated once an action is selected. In the proposed MDP for the problem, the transition is dependent on the customers selected to be served. If the parcels of the selected customers are at the depot, the vehicle departs for the selected route, and the next stage takes place once the vehicle has returned to the depot. Otherwise, the vehicle remains at the depot and the next stage is the next time instant. The state of the process is updated accordingly. Formally, we denote as $n(\chi_e) \in N_{t_e}^{unserved}$ the set of customers selected by action χ_e, i.e., to be served in the next route. The state s_{e+1} of the process at stage $e+1$ is described as $s_{e+1} = (t_{e+1}, N_{t_{e+1}}^{known}, \{\tilde{r}_i^{t_{e+1}}\}_{i \in N_{t_{e+1}}^{future}})$. If $n(\chi_e) \subseteq N_{t_e}^{known}$, route χ_e can start at time t_e, t_{e+1} is computed as $t_{e+1} = t_e + t(\chi_e)$, where $t(\chi_e)$ is the time required to perform route χ_e, $N_{t_{e+1}}^{known}$ is updated as $N_{t_{e+1}}^{known} := N_{t_{e+1}}^{known} \setminus n(\chi_e)$, that is, the customers whose parcel has arrived at the depot at time t_{e+1} minus those that have been served by route χ_e. Otherwise, if the action χ_e is to serve a set of customers that includes some in $N_{t_e}^{future}$, t_{e+1} is updated as $t_{e+1} = t_e + 1$, while $N_{t_{e+1}}^{known}$ is given. In both cases, the information about the future release dates, denoted as $\{\tilde{r}_i^{t_{e+1}}\}_{i \in N_{t_{e+1}}^{future}}$, is given.

Objective

The objective is the minimization of the completion time of the distribution, which corresponds to the time in which the vehicle returns to the depot after its final trip. This time is represented as the time t_E of the final state s_E. At any stage before $E-1$, that is, the stage in which the selected decision is to depart from the depot to perform the final route, the ending time of the distribution is a random variable. The objective of the problem at any stage is therefore defined as the minimization of the expected ending time of the distribution, computed as the total cost of the routes plus the expected cost of waiting at the depot for the parcels of the customers to arrive.

3 Solution approaches

In this section the solution approaches devised for the problem are described. First, the heuristic reoptimization approach based on the solution of the MDP described above. Then, the myopic algorithm based on the greedy strategy of sending out the vehicle as soon as there is a parcel ready to be distributed.

3.1 Heuristic reoptimization approach

The size of the stage space in the MDP proposed for the problem can be large, as, in any stage e, the number of remaining stages is bounded only by the maximum upper bound of the release dates of the customers in $N_{t_e}^{unserved}$. A reoptimization approach is proposed which heuristically reduces the number of stages. A rule is therefore required to identify the subset of stages in which a reoptimization is carried out. Henceforth, we call a rule to select the stages a *strategy* and the selected stages *reoptimization epochs*. A strategy can be seen as an alternative way, compared to the one described in Section 2, to transition from a state to another once an action is selected. At each reoptimization epoch e, a problem is solved with updated information about the release dates for the customers in $N_{t_e}^{unserved}$. Three different strategies are proposed.

The first and simplest one solves the problem at the beginning of the distribution and each time the vehicle returns to the depot at the end of a route. This means that a stage does not happen until the route selected at the previous stage has been performed. This strategy is identified as strategy R.

The second strategy reoptimizes at every return to the depot and every time the vehicle should depart to perform a route, i.e., at the earliest time in which the next route can start. If the first route of the newly obtained solution can start, this route is implemented. Otherwise, the process is repeated. This means that, given an action, if the selected route cannot start, the next stage happens, i.e., when all the parcels of the selected customers have reached the depot. This strategy is identified as strategy RD.

The last strategy reoptimizes at every time the vehicle returns to the depot, every time the vehicle should depart from the depot, and at intervals while the vehicle is waiting at the depot. During waiting times, the next reoptimization epoch is defined as the middle point between the current time and the expected starting time of the next route. If the interval between the current time and the next starting time is less than a minimum time span between reoptimization epochs, the next reoptimization is performed at the expected starting time of the next route. This means that, given an action, if the selected route cannot start, the next stage happens at the mid-point between the time when the action was taken and the expected starting time of the selected route. This strategy is identified as RDW strategy.

The flowchart of the RDW strategy is presented in Figure 1, where τ_{start}^{next} is the actual starting time of the next route, that is, the time at which all the parcels of the customers to be served in the next route have been delivered to the depot and thus the vehicle can start the route and serve the customers.

The highlighted box in Figure 1 identifies how the action is selected at any reoptimization epoch. In particular, in the reoptimization approach the DTSP-srd is solved with the information available at time t, i.e., the set of customers in $N_t^{unserved}$ and the information on the corresponding release dates. As such, the solution found by the optimization model

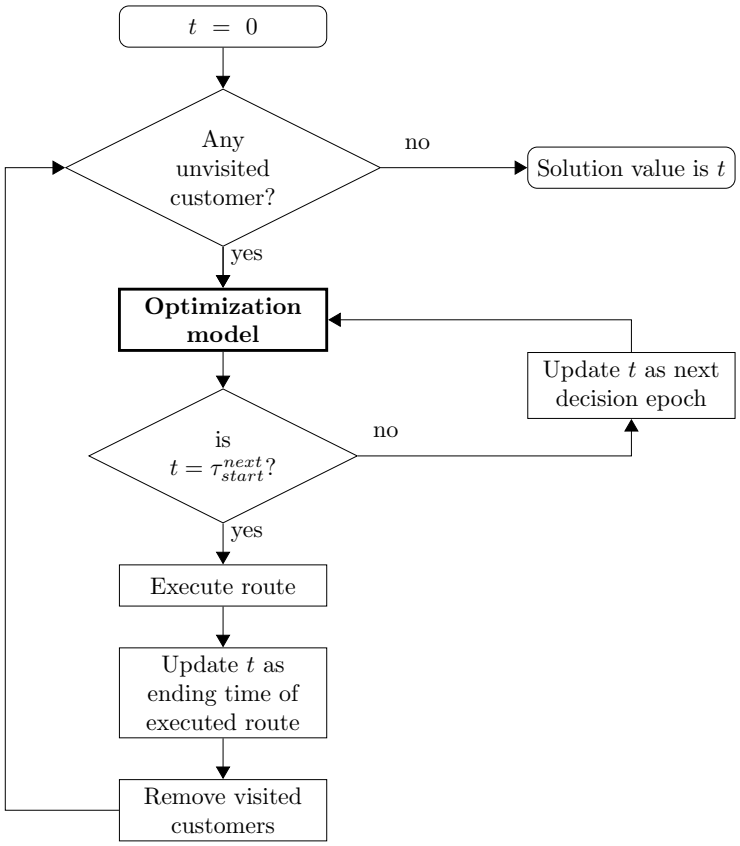


Figure 1: The RDW strategy.

may involve more than one route. However, only the first will be performed when the vehicle will depart from the depot. The following routes might be revisited in successive reoptimization epochs, i.e., when the vehicle is back to the depot. When the first route is performed, the customers that have been served are removed from the set N_t^{known} and added to the set N_t^{served}.

Concerning the optimization models, two different models are proposed to handle the information available at each stage. The first is a stochastic model that treats the release dates as random variables. The second is a deterministic model and makes use of a point estimation of the release date from the stochastic release date $\tilde{r}_i^t$ of customer $i \in N_t^{future}$. The models and the solution approach used to solve them are described in Section 4.

3.2 Myopic algorithm

In the myopic algorithm, the distribution is carried out without considering the information available for future release dates. In particular, the vehicle serves the customers as soon as their parcels reach the depot. If no customer can be served, the vehicle waits for the first parcel to arrive at the depot and immediately starts a route when a parcel arrives.

4 Optimization models

In this section the models used to select an action from the action space at each reoptimization epoch are presented. The stochastic model is introduced in Section 4.1. Then, in Section 4.2, the deterministic model is described. To illustrate the differences between the two models an example is presented in Section 4.3. Finally, the solution method proposed for the two models is described in Section 4.4.

4.1 Stochastic model

If the full stochastic information on the release dates is considered, the proposed stochastic mixed-integer linear program for the problem at time t is defined as follows. The formulation relies on the following decision variables: x_{ij}^k are arc variables, with $x_{ij}^k = 1$ if route k traverses arc (i, j) and $x_{ij}^k = 0$ otherwise, y_i^k are visit variables, with $y_i^k = 1$ if customer i is visited in route k and $y_i^k = 0$ otherwise. Flow variables u_{ij}^k are added to enforce subtour elimination.

$$\min \sum_{k \in K} \sum_{(i,j) \in A} d_{ij} x_{ij}^k + \mathbb{E}\left[Q(x, \tilde{r}^t)\right] \tag{1}$$

s.t.

$$\sum_{k \in K} y_i^k = 1 \qquad i \in N_t^{unserved}, \tag{2}$$

$$\sum_{j \in V} x_{ij}^k = \sum_{j \in V} x_{ji}^k = y_i^k \qquad i \in N_t^{unserved} \cup \{0\}, k \in K, \tag{3}$$

$$\sum_{j \in V} u_{ji}^k - \sum_{j \in V} u_{ij}^k = y_i^k \qquad i \in N_t^{unserved}, k \in K, \tag{4}$$

$$u_{ij}^k \leq (n-1) x_{ij}^k \qquad (i, j) \in A, k \in K, \tag{5}$$

$$x_{ij}^k \leq 1 - x_{00}^k \qquad (i, j) \in A, k \in K \setminus \{|K|\}, \tag{6}$$

$$x_{00}^k \geq x_{00}^{k+1} \qquad k \in K \setminus \{|K|\}, \tag{7}$$

$$x_{ij}^k \in \{0, 1\} \qquad (i, j) \in A, k \in K, \tag{8}$$

$$y_i^k \in \{0, 1\} \qquad i \in N_t^{unserved} \cup \{0\}, k \in K, \tag{9}$$

$$u_{ij}^k \geq 0 \qquad (i, j) \in A, k \in K. \tag{10}$$

The objective function (1) minimizes the total completion time as sum of the travel times and the expected waiting times at the depot, where $x = (x_{ij}^k)_{(i,j) \in A, k \in K \setminus \{|K|\}}$ and $\tilde{r}^t = (\tilde{r}_i^t)_{i \in N_t^{unserved}}$ is the random vector of the release dates. Constraints (2) ensure the visit of all customers. Constraints (3)-(5) impose that each route is a circuit connected to the depot. In particular, constraints (5) generate a flow that decreases while the vehicle visits customers, which prevents subtours. Constraints (6) and (7) are symmetry breaking constraints. In particular, constraints (6) set $x_{00}^k = 1$ if route k is empty, $x_{00}^k = 0$ otherwise,

and constraints (7) impose that all the empty routes, if any, must precede all the non-empty routes, i.e., if route k is empty then all routes k' must be empty, with $k' = 1, \ldots, k-1$.

Given a solution and a realization for the random vector $\tilde{r}^t$, $Q(x, \tilde{r}^t)$ denotes the waiting time at the depot. Waiting occurs if the starting time of a route is greater than the ending time of the previous one, or, in the case of the first route, if it is not starting at time $t = 0$. The ending time of a route is computed as its starting time plus its travel time. The starting time of a route must be greater than or equal to the release date of the customers in the route and the ending time of the previous route. Let τ^k_{start} and τ^k_{end} be the starting and ending time of route $k \in K$. Then:

$$\tau^k_{end} = \tau^k_{start} + \sum_{(i,j)\in A} d_{ij} x^k_{ij} \qquad k \in K, \tag{11}$$

$$\tau^k_{end} \leq \tau^{k+1}_{start} \qquad k \in K \setminus \{|K|\}, \tag{12}$$

$$\tau^k_{start} \geq \tilde{r}^t_i y^k_i \qquad i \in N, k \in K. \tag{13}$$

It follows that

$$\mathbb{E}\left[Q(x, \tilde{r}^t)\right] = \mathbb{E}\left[\sum_{k\in K\setminus\{|K|\}} \left(\tau^{k+1}_{start} - \tau^k_{end}\right)\right]. \tag{14}$$

Therefore, the objective function (1) can be computed as the expected ending time of the last route:

$$\sum_{k\in K} \sum_{(i,j)\in A} d_{ij} x^k_{ij} + \mathbb{E}\left[Q(x, \tilde{r}^t)\right] = \mathbb{E}\left[\tau^{|K|}_{end}\right]. \tag{15}$$

The model (1)-(10) is solved with the solution method presented in Section 4.4 where solutions are evaluated by the expected ending time of the last route. In the remainder of this section the computation of this expected value is described. Let $\mathcal{S}$ be a solution of (1)-(10), obtained at time t. The vehicle is assumed to be located at the depot and $\mathcal{S}$ provides the delivery routes for customers in $N^{unserved}_t$. The waiting time depends on the release dates $\tilde{r}^t_i$. In the remainder of this section, $\tilde{r}^t_i$ is assumed to be a random variable. Note that the known release date of customers in N^{known}_t can be seen as a random variable with bounds $l^t_i = u^t_i = r_i$ and $P(\tilde{r}^t_i = r_i) = 1$. In what follows the index t is removed to improve readability.

The expected completion time of $\mathcal{S}$ is computed as $\mathbb{E}\left[\tau^{|K_\mathcal{S}|}_{end}]\right]$, where $K_\mathcal{S} = \{1, \ldots, |K_\mathcal{S}|\}$ is the set of routes of $\mathcal{S}$. From equations (11) and (12) it follows that the starting time of a route can be computed as:

$$\tau^k_{start} = \max\left\{\tau^{k-1}_{end}, \max_{i\in N_k}\{\tilde{r}_i\}\right\}, \qquad k \in K_\mathcal{S} \setminus \{1\}, \tag{16}$$

where $N_k := \{i \in N^{unserved}_t | y^k_i = 1\}$ is the set of customers served in route k.

The computation makes use of the cumulative distribution function (cdf) of the maximum of the release dates of the customers in route k, $\max_{i\in N_k}\{\tilde{r}_i\}$, computed as:

$$P\left(\max_{i\in N_k}\{\tilde{r}_i\} \leq t'\right) = \prod_{i\in N_k} P\left(\tilde{r}_i \leq t'\right), \qquad k \in K_{\mathcal{S}}. \tag{17}$$

Equations (16) and (17) allow us to define the cumulative distribution function of τ^k_{start} as:

$$P\left(\tau^k_{start} \leq t'\right) = \begin{cases} P\left(\max_{i\in N_k}\{\tilde{r}_i\} \leq t'\right) & \text{for } k = 1 \\ P\left(\tau^{k-1}_{end} \leq t'\right) \cdot P\left(\max_{i\in N_k}\{\tilde{r}_i\} \leq t'\right) & \text{for } 2 \leq k \leq |K_{\mathcal{S}}|. \end{cases} \tag{18}$$

The support of the random variable τ^k_{start} is defined as follows. Let l^k_{start} and u^k_{start} be the lower and upper bound of τ^k_{start}, respectively. The two bounds are computed as:

$$l^k_{start} = \begin{cases} \max_{i\in N_k}\{l_i\} & \text{for } k = 1 \\ \max\left\{l^{k-1}_{end}, \max_{i\in N_k}\{l_i\}\right\} & \text{for } 2 \leq k \leq |K_{\mathcal{S}}| \end{cases} \tag{19}$$

$$u^k_{start} = \begin{cases} \max_{i\in N_k}\{u_i\} & \text{for } k = 1 \\ \max\left\{u^{k-1}_{end}, \max_{i\in N_k}\{u_i\}\right\} & \text{for } 2 \leq k \leq |K_{\mathcal{S}}|. \end{cases} \tag{20}$$

Because of equation (11), the bounds on τ^k_{end} are computed as $l^k_{end} = l^k_{start} + d_k$ and $u^k_{end} = u^k_{start} + d_k$, where d_k is the traveling time of route k.

The expected value of solution $\mathcal{S}$ is therefore computed as:

$$\mathbb{E}\left[\tau^{|K_{\mathcal{S}}|}_{end}\right] = \sum_{t=l^{|K_{\mathcal{S}}|}_{end}}^{u^{|K_{\mathcal{S}}|}_{end}} t \cdot \left[P\left(\tau^{|K_{\mathcal{S}}|}_{end} \leq t\right) - P\left(\tau^{|K_{\mathcal{S}}|}_{end} \leq t-1\right)\right]. \tag{21}$$

where $P\left(\tau^{|K_{\mathcal{S}}|}_{end} \leq t\right)$ follows from expressions (11) and (18). Thanks to (15) this allows us to compute the objective function for any solution.

The complexity of computing the expected value of the objective function is $\mathcal{O}(|S_{\tau^{|K_{\mathcal{S}}|}_{end}}| \cdot |N|)$, where $S_{\tau^{|K_{\mathcal{S}}|}_{end}}$ is the support of the random variable $\tau^{|K_{\mathcal{S}}|}_{end}$. The term $|S_{\tau^{|K_{\mathcal{S}}|}_{end}}|$ is due to the fact that the computation of the expected value involves a sum over all the elements of the support, as described in expression (21). The term $|N|$ is due to the fact that the computation of the cdf of $\max_{i\in N_k}\{\tilde{r}_i\}$, having cost $\mathcal{O}(|N_k|)$ as per expression (17), must be performed for all routes in K, as defined in expression (18), and therefore $\sum_{k=1}^{|K|} \mathcal{O}(|N_k|) = \mathcal{O}(|N|)$.

4.2 Deterministic model

The second of the two proposed models for the DTSP-srd makes use of a point estimation for the stochastic release dates. The information to be used in the model is the observed release date of customers in N^{known}_t, and the expected value of the stochastic release dates for customers in $i \in N^{future}_t$. The problem to be solved at any reoptimization epoch is a TSP-rd(time) as proposed in Archetti et al. [2018].

4.3 Illustrative example

An example is now shown to illustrate the differences between the two models. As the aim of the example is a comparison of the two models without taking into consideration the effect of the different reoptimization strategies and the effect of the dynamic information on the solution, the release dates are considered as static and only the solution obtained at time $t = 0$ is shown. Without affecting the validity of the example, in what follows the assumption that $t \in \mathbb{N}$ is waived to allow for a leaner text. The example can be made valid with $t \in \mathbb{N}$ by multiplying the arc costs and the values of the support of the release dates by one thousand.

Let there be two customers arranged as in Figure 2. The number close to each arc is d_{ij}. The random variables describing the release dates of the two customers, denoted as

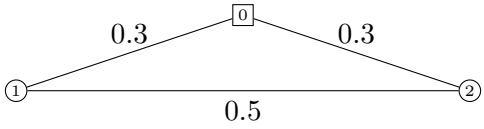


Figure 2: Example.

$\tilde{r}_1$ and $\tilde{r}_2$, are defined by the following probability density functions (pdf):

- Customer 1:

$$f_{\tilde{r}_1}(t) = \begin{cases} 0.5 & \text{if } t = 1 \\ 0.5 & \text{if } t = 2 \\ 0 & \text{otherwise.} \end{cases}$$

- Customer 2:

$$f_{\tilde{r}_2}(t) = \begin{cases} 0.5 & \text{if } t = 1 \\ 0.5 & \text{if } t = 2.1 \\ 0 & \text{otherwise.} \end{cases}$$

Let us consider two solutions for the DTSP-srd problem. The first solution, $\mathcal{S}_1$, consists of two routes, $\mathcal{S}_{1,1}$ and $\mathcal{S}_{1,2}$, visiting customer 1 and customer 2, respectively. The traveling time of the routes is $c_{\mathcal{S}_{1,1}} = c_{\mathcal{S}_{1,2}} = 0.6$. The second solution, $\mathcal{S}_2$, consists of a single route $\mathcal{S}_{2,1}$, visiting both customers 1 and 2, with travel time $c_{\mathcal{S}_{2,1}} = 1.1$.

The deterministic model evaluates the two solutions in the following way:

- solution $\mathcal{S}_1$ has an estimated value of 2.7, as route $\mathcal{S}_{1,1}$ has an estimated starting time $\tau_{start}^{\mathcal{S}_{1,1}} = \mathbb{E}[\tilde{r}_1] = 1.5$ and route $\mathcal{S}_{1,2}$ has an estimated starting time $\tau_{start}^{\mathcal{S}_{1,2}} = \max\{\tau_{end}^{\mathcal{S}_{1,1}}, \mathbb{E}[\tilde{r}_2]\} = 2.1$;

- solution $\mathcal{S}_2$ has an estimated value of 2.65 as route $\mathcal{S}_{2,1}$ has an estimated starting time $\tau_{start}^{\mathcal{S}_{2,1}} = \max\{\mathbb{E}[\tilde{r}_1], \mathbb{E}[\tilde{r}_2]\} = 1.55$.

The stochastic model evaluates the two solutions in the following way:

- solution $\mathcal{S}_1$ is evaluated according to the procedure described in Section 4.1:
  - the starting time of $\mathcal{S}_{1,1}$ is defined as $\tau_{start}^{\mathcal{S}_{1,1}} = \tilde{r}_1$, that is, $f_{\tau_{start}^{\mathcal{S}_{1,1}}} = f_{\tilde{r}_1}$;
  - because of equation (11), $\tau_{end}^{\mathcal{S}_{1,1}}$ is defined as $\tau_{end}^{\mathcal{S}_{1,1}} = \tau_{start}^{\mathcal{S}_{1,1}} + c_{\mathcal{S}_{1,1}}$. The pdf of $\tau_{end}^{\mathcal{S}_{1,1}}$ is defined as:
$$f_{\tau_{end}^{\mathcal{S}_{1,2}}}(t) = \begin{cases} 0.5 & \text{if } t = 1.6 \\ 0.5 & \text{if } t = 2.6 \\ 0 & \text{otherwise;} \end{cases}$$
  - the starting time of the second route is defined as $\tau_{start}^{\mathcal{S}_{1,2}} = \max\{\tau_{end}^{\mathcal{S}_{1,1}}, \tilde{r}_2\}$. The cumulative distribution function is obtained as described in (18): $F_{\tau_{start}^{\mathcal{S}_{1,2}}}(t) = F_{\tau_{end}^{\mathcal{S}_{1,1}}}(t) \cdot F_{r_2}(t)$. The resulting pdf of $\tau_{start}^{\mathcal{S}_{1,2}}$ is the following:
$$f_{\tau_{start}^{\mathcal{S}_{1,2}}}(t) = \begin{cases} 0.25 & \text{if } t = 1.6 \\ 0.25 & \text{if } t = 2.1 \\ 0.5 & \text{if } t = 2.6 \\ 0 & \text{otherwise;} \end{cases}$$
  - combining equation (11) with the pdf of $\tau_{start}^{\mathcal{S}_{1,2}}$ allows us to compute the expected ending time of $\mathcal{S}_{1,2}$ as $\mathbb{E}[\tau_{end}^{\mathcal{S}_{1,2}}] = 2.825$;
- solution $\mathcal{S}_2$ has expected value 2.9. The random variable of the starting time is defined as $\tau_{start}^{\mathcal{S}_{2,1}} = \max\{\tilde{r}_1, \tilde{r}_2\}$ and therefore has $l_{\tau_{start}^{\mathcal{S}_{2,1}}} = \max_{i\in\{1,2\}}\{l_{\tilde{r}_i}\} = 1$ and $u_{\tau_{start}^{\mathcal{S}_{2,1}}} = \max_{i\in\{1,2\}}\{u_{\tilde{r}_i}\} = 2.1$. The pdf of $\tau_{start}^{\mathcal{S}_{1,2}}$ is:
$$f_{\tau_{start}^{\mathcal{S}_{1,2}}}(t) = \begin{cases} 0.25 & \text{if } t = 1 \\ 0.25 & \text{if } t = 2 \\ 0.5 & \text{if } t = 2.1 \\ 0 & \text{otherwise,} \end{cases}$$
  thus, $\mathbb{E}[\tau_{start}^{\mathcal{S}_{2,1}}] = 1.8$ and $\mathbb{E}[\tau_{end}^{\mathcal{S}_{2,1}}] = 2.9$.

Table 1 reports the expected value of the two solutions according to the stochastic and the deterministic models.

Model	$\mathcal{S}_1$	$\mathcal{S}_2$
Deterministic	2.7	**2.65**
Stochastic	**2.825**	2.9

Table 1: Expected values of the solutions.

This result shows how, in the presented example, the deterministic model chooses the solution that waits for the two customers while the stochastic model chooses to split the

service of the two customers, anticipating the visit of customer 1. The two solutions have been evaluated in the four scenarios, originating from the combination of the realization of the random variables representing the release dates of the two customers. The value of each solution in every scenario is reported in Table 2.

Scenario	$\tilde{r}_2 = 1$	$\tilde{r}_2 = 2.1$
$\tilde{r}_1 = 1$	$\mathcal{S}_1 = 2.2$	$\mathcal{S}_1 = \mathbf{2.7}$
	$\mathcal{S}_2 = \mathbf{2.1}$	$\mathcal{S}_2 = 3.2$
$\tilde{r}_1 = 2$	$\mathcal{S}_1 = 3.2$	$\mathcal{S}_1 = \mathbf{3.2}$
	$\mathcal{S}_2 = \mathbf{3.1}$	$\mathcal{S}_2 = \mathbf{3.2}$

Table 2: Results.

When looking at the best solution for each scenario, it can be observed that $\mathcal{S}_1$, the solution chosen by the stochastic model, finds the best solution in 2 over the 4 scenarios and $\mathcal{S}_2$, the solution chosen by the deterministic model, is the best solution in all but one scenarios. However, when evaluating the difference in terms of average percentage gap from the best solution in each scenario, $\mathcal{S}_1$ has an average gap of 2% while $\mathcal{S}_2$ has an average gap of 4.63%. This result is explained by the fact that, as highlighted in Table 2, $\mathcal{S}_2$ is greatly penalized in the scenario in which customer 1 is early and customer 2 is late.

4.4 A heuristic algorithm

At each reoptimization epoch, the solution of the two models is obtained through a heuristic method derived from the one presented in Archetti et al. [2018], which we briefly recall. The authors present an iterated local search (ILS) composed by two phases: a perturbation, consisting in a destroy-and-repair (DR) procedure, followed by a local search (LS). The two phases are repeated iteratively as shown in Figure 3. The DR is carried out by removing a number of customers from the solution and adding them back in a randomly chosen route, different from the one they were originally included in. The LS considers a set of four neighborhoods: the relocation of a customer to a different route, the merging of two consecutive routes, the split of a route in two separate ones, and the anticipation or postponement of a depot visit. The vertices are randomly sorted and considered sequentially: if the vertex is not the depot the customer relocation move is considered, otherwise the remaining three moves are evaluated. The first improving move is implemented and the process is repeated until no improving move is found. The ending conditions for the heuristic are the maximum time allowed and the maximum number of iterations without improvement. Each LS iteration has complexity $\mathcal{O}(|N|^2)$, as the node insertion move is evaluated for all nodes in all the possible positions within a solution.

The algorithm from Archetti et al. [2018] was shown to have good performance, finding the optimal solution in all but one of the 24 instances tested with size $|N| = 20$, the largest that could be solved to optimality within one hour with CPLEX. In the current work, the

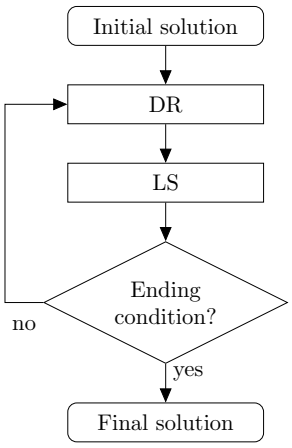


Figure 3: The heuristic scheme from Archetti et al. [2018].

algorithm has been modified to reflect the dynamic setting. In particular, the local search algorithm has been revised to become a best improvement local search where solutions will null improvement over the current best solution are implemented with the condition that the structure of the solution is improved, as explained in the following. When the deterministic model is solved, between two equivalent solutions, the one minimizing the sum over all routes of the difference between the expected starting time and the maximum release date is chosen. This criterion is expressed as $\sum_{k \in K} \left(\tau^k_{start} - \max_{i \in N_k}\{\hat{r}^t_i\}\right)$. This allows us to select the solution anticipating the visit to customers as much as possible. This property is intuitively useful in a dynamic setting, where the information about arrivals in near future is expected to be less uncertain than that of customers whose goods will arrive further away in time. Given a solution and a move, the evaluation of the neighboring solution of the deterministic model has complexity $\mathcal{O}(1)$. Each LS iteration for the deterministic model has therefore complexity $\mathcal{O}(|N|^2)$.

For the stochastic model, the solution is obtained by adopting the same heuristic scheme presented for the deterministic model modified to consider the stochastic component of the waiting times. In the local search, solutions are kept if they improve the sum of the total travel time and expected waiting times, i.e., the expected value of the solution, computed as described in Section 4.1. Similarly to the deterministic case, solutions with null improvement are kept only if the structure of the solution is improved. As the release dates are stochastic, this criterion is measured as the expected value of the sum, for all routes, of the difference between the starting time and the maximum release date, which are both random variables, i.e., $\mathbb{E}\left[\sum_{k \in K} \left(\tau^k_{start} - \max_{i \in N_k}\{\tilde{r}^t_i\}\right)\right] = \sum_{k \in K} \left(\mathbb{E}\left[\tau^k_{start}\right] - \mathbb{E}\left[\max_{i \in N_k}\{\tilde{r}^t_i\}\right]\right)$. The solution for which such value is minimum is kept.

As reported in Section 4.1, the evaluation of a solution of the stochastic model has complexity $\mathcal{O}(|S_{\tau_{end}^{|K_{\mathcal{S}}|}}| \cdot |N|)$. Each LS iteration has therefore complexity $\mathcal{O}(|S_{\tau_{end}^{|K_{\mathcal{S}}|}}| \cdot |N|^3)$.

5 Instance generation

The following sections describe the instance generation procedure, firstly reporting the instances from which the DTSP-srd instances are derived, and secondly defining how the release dates of each customer are generated and updated over time. In a brief overview, the release dates of customers are generated by simulating the traveling of the vehicles delivering the parcels to the depot. The dynamic release dates are updated as the vehicles travel to the depot. When the vehicle reaches the depot the release date becomes known.

All instances as well as detailed results are available at `http://or-brescia.unibs.it/instances`.

5.1 Input instances

Instances have been generated starting from the instances for the TSP-rd(time) described in Archetti et al. [2018], and in particular from the subset of instances derived from Solomon [1987]. Each instance is denoted as $\mathcal{I}_\beta$ where $\mathcal{I}$ is the name of the instance from which the vertices coordinates are taken, i.e., C101, C201, R101, and RC101, and β represents how spread out in time the release dates can be, compared to the TSP travel time of the instance. Denoting d_{TSP} the TSP travel time of the instance, each release date is uniformly sampled in the interval $[0, \beta \times d_{TSP}]$. We denote $\bar{r}_i$ the release date of customer i from the instance for the TSP-rd(time).

Starting from these input instances, the customers in the sets N_0^{known}, N_0^{static}, or $N_0^{dynamic}$ are partitioned. In particular, all customers for which $\bar{r}_i = 0$, that is, all customers whose parcel is at the depot at time $t = 0$ in the instance in Archetti et al. [2018], are included in the set N_0^{known}. Customers in $N \setminus N_0^{known}$ are assigned to N_0^{static} or $N_0^{dynamic}$ according to a parameter $\delta \in [0, 1]$ indicating the rate of customers with a dynamically updated release date, i.e., the customers in set $N_0^{dynamic}$ over the total number of customers in $N \setminus N_0^{known}$. In particular, $\delta \left\lceil |N \setminus N_0^{known}| \right\rceil$ randomly chosen customers are assigned to set $N_0^{dynamic}$ and the remaining customers are assigned to N_0^{static}.

Instances with $\mathcal{I} \in \{C101, C201, R101, RC101\}$, $\beta \in \{0.5, 1, 1.5\}$, and $\delta \in \{0, 0.5, 1\}$ have been tested.

5.2 Release dates generation and updating

In this section we present the procedure to generate the random variables describing the initial release dates for the customers in N_0^{future}, i.e., $\tilde{r}_i^0$. As defined in Section 2, the release dates are assumed to be bounded and discrete. The discrete and bounded distribution for each release date is generated by first choosing a continuous probability distribution which is then truncated and discretized.

Concerning the choice of the continuous probability distribution, various distributions have been considered in the vehicle routing literature modelling stochasticity in travel times, including uniform (see Malandraki and Daskin [1992]), Gaussian (see Li et al. [2010]),

log-normal (see Kaparias et al. [2008]), gamma (see Fan et al. [2005] and Taş et al. [2013]) and shifted gamma (see Russell and Urban [2008]). The computational results reported in the following sections have been obtained using a Gaussian distribution. Note, however, that the proposed models and the related solution methods are not distribution dependent. The parameters of the continuous Gaussian distribution are the expectation and the variance of the distribution, denoted as $\mu_{\tilde{r}_i^0}$ and $\sigma^2_{\tilde{r}_i^0}$, respectively. The Gaussian distribution is truncated to the 1^{st} and the 99^{th} percentile. Parameters $\mu_{\tilde{r}_i^0}$ and $\sigma^2_{\tilde{r}_i^0}$ are obtained by simulating the arrival of the vehicles delivering the parcels to the depot, assuming a speed in the interval $[10, 100]$.

In the case of customers in N_0^{static}, the information is not updated for $t > 0$. For these customers the actual release date r_i is assumed to fall within the interval $[l_i^0, u_i^0]$. For customers in $N_t^{dynamic}$, the release date is updated by simulating the traveling of the vehicles delivering the parcels to the depot. The distance of each vehicle is updated by decreasing its value by the distance travelled in the previous time unit. If the updated distance from the depot is null (or negative) the vehicle has reached the depot and the customer is included in the set N_t^{known} with release date $r_i = t$. Otherwise, the speed of the vehicle is updated and the new parameters of the release date are computed. The speed is updated as a truncated random walk process with Gaussian steps.

In order to give an idea of the role of parameters β and δ, we provide an example of their effect on release date generation in Figures 4 and 5. In particular, Figure 4 presents the effect of β on the distribution of the supports of the release dates in time. The plots refer to three instances with value of $\beta = 0.5, 1, 1.5$, from top to bottom, and with $\delta = 0$. The x-axis reports the time and the y-axis the customers of the instance. The horizontal lines of each plot represent the support of the distribution function of the release dates of customers. Each line corresponds to one customer. Customers are sorted by increasing value of the expected release date, from bottom to top. As β increases, the release dates are sampled in a broader interval with respect to d_{TSP}. This allows us to investigate both the cases in which it is better to wait for more customers to serve in longer routes, when β is small, and the one where it is more advantageous to serve customers as their parcel reaches the depot, when β increases.

Figure 5 shows the evolution of the expected value and bounds of the release dates when $\delta = 1$. The figure reports the evolution of the release date of six customers, sorted by the value of the initial expected release dates. At the beginning of the distribution, the release dates have a given variance, based on the distance from the depot. As time goes on, the expected arrival time each vehicle might be advanced or delayed, based on various conditions, e.g., traffic and deliveries to other distribution centers. As the vehicle travels, the potential causes of uncertainty decrease and the estimation of the arrival time gets more precise, until the moment in which the vehicle reaches the depot.

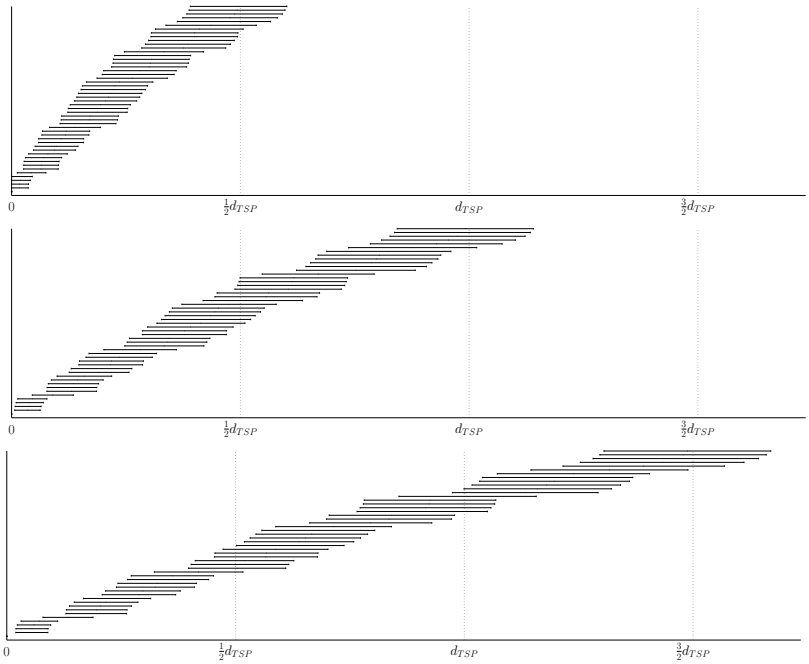


Figure 4: The effect of different values of β on the release dates. $\beta = 0.5, 1, 1.5$ from top to bottom.

6 Computational experiments

In this section we describe the computational experiments that have been carried out to assess the performance of the two models and the three reoptimization strategies. The aim is to understand whether, in the presented reoptimization and stochastic setting, it is better to consider a point estimation or the entire stochastic information for the release dates and if increasing the number of reoptimization epochs improves the quality of the solution. We proceed as follows. In Section 6.1, the results of preliminary experiments are presented where the aim is to tune the parameters of the algorithm presented in Section 3.1. Section 6.2 is devoted to final results. First, the value of reoptimization is shown by comparing the results of a 'static' approach where the problem is solved at time 0 and never reoptimized afterwards. Then, we compare the stochastic versus the deterministic model and show that the stochastic model provides better solution quality on average, at the expense of a higher computing time. In addition, the R, RD and RDW reoptimization strategies are compared to show the value of a higher reoptimization frequency. Finally, results are presented for instances where the release dates follow a different distribution with respect to the one used for the other tests. The distribution mimics situations where peak hours occur followed by low activity time intervals. The results show that the performance of the algorithm is not affected by a different probability distribution of release dates.

It must be noted that, throughout this paper, the solution of the TSP has been achieved with the Lin and Kernighan [1973] algorithm in the implementation described in Helsgaun [2000].

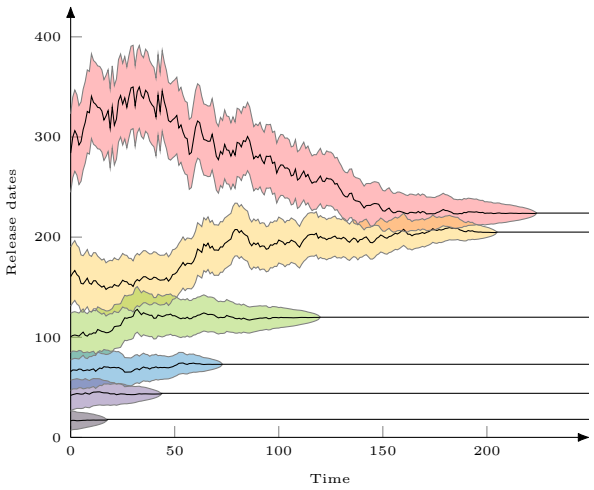


Figure 5: The evolution of the release dates when $\delta = 1$.

6.1 Preliminary experiments

In this section preliminary experiments are reported to identify the time limit for each reoptimization and investigate the relationship between the instance size and the number of ILS iterations performed.

The time limit $T_{\max}$ for each reoptimization, expressed in minutes, has been obtained through a set of preliminary experiments. Tests have been carried out for $T_{\max} \in \{1, 5, 10\}$ on the instances with $\mathcal{I} = C101$, $\beta \in \{0.5, 1, 1.5\}$, and $\delta = \{0, 0.5, 1\}$. It must be noted that, as the time required for each reoptimization is not taken into account when computing the value of the objective function, the implicit assumption is that the considered time unit is larger than the value of $T_{\max}$. The maximum number of iterations without improvement has been set to $100 \cdot T_{\max}$. The results are reported in Table 3. Column "Avg" shows, for each tested value of $T_{\max}$, the average gap of each model from the best solution found by that model across all the tested values of $T_{\max}$. As the deterministic model has shown some outlying results for the instance with $\beta = 1.5$ and $\delta = 0.5$, the average percentage gap excluding such instance, for both the deterministic and stochastic models, is reported in the column "Avg*". This value shows an improvement in the quality of the solution found by the deterministic model as the time allowed for each reoptimization increases, while the stochastic model shows no benefit when $T_{\max}$ is increased from 5 to 10 minutes. The column "Max Sim. Time" shows the maximum time required to run each simulation, i.e., all the reoptimizations required to solve an instance, for each value of $T_{\max}$, across all model-strategy combinations. The value $T_{\max} = 5$ has been chosen as it allowed to run each simulation in less than two hours without significantly undermining the performance of the two models.

The performance of the ILS depends on the number of iterations performed and therefore on the time that each iteration requires. Table 4 reports for instance $\mathcal{I} = C101$, $\beta = 1$, $\delta = 0.5$, the average number of iterations, over the three strategies, performed by

$T_{\max}$ (minutes)	Deterministic		Stochastic		Max Sim. Time (minutes)
	Avg	Avg*	Avg	Avg*	
1	1.63	1.35	1.37	1.37	21
5	1.43	1.27	0.66	0.61	102
10	1.68	1.21	0.66	0.62	201

Table 3: The results of the tests for each value of $T_{\max}$.

each model by size of the subgraph of customers in $N_t^{unserved}$. The "-" symbol indicates that the model has not been solved for subgraphs of that size. The results show how the number of iterations required when solving the problem considering the full probabilistic information for the release dates is substantially smaller than in the deterministic case, especially in the early stages of the simulation, when the size of the subgraph considered in the reoptimization is larger.

	Number of iterations	
Size	Deterministic	Stochastic
50-46	718	18
45-41	715	63
40-36	-	201
35-31	613	226
30-26	2248	500
25-21	1765	500
5-1	500	500

Table 4: Results on the average number of ILS iterations for the two models.

As the number of ILS performed within $T_{\max}$ appears to be related to the size of the instance to be solved, tests have been conducted to measure the time to perform one ILS iteration for different instance sizes. Within the single ILS iteration examined, the focus has been on the time required to perform the LS phase of the ILS. The results are reported in Table 5 for instance $\mathcal{I} = C101$, $\beta = 1$, $\delta = 0.5$. Column "t_{LS}" reports the total time required for the LS, "n_{LS}" the number of LS iterations performed and "$\bar{t}_{LS}$" the average time required for one LS iteration. It is shown how the average time required to perform a LS iteration for the deterministic model appears to be constant with respect to the instance size while it increases when the stochastic model is solved. It must also be noted that when the instance size is greater or equal to 70 nodes the cost of one LS iteration for the stochastic model becomes greater than $T_{\max}$.

	Deterministic			Stochastic		
Size	t_{LS}	n_{LS}	$\bar{t}_{LS}$	t_{LS}	n_{LS}	$\bar{t}_{LS}$
50	0.11	8	0.014	49.14	16	3.07
60	0.14	8	0.017	46.34	14	3.31
70	0.28	22	0.013	335.93	36	9.33
80	0.41	58	0.007	406.88	15	27.13
90	0.25	42	0.006	557.44	15	37.16
100	0.34	31	0.011	310.07	16	19.38

Table 5: Time in seconds to perform one ILS iteration.

6.2 Final results

Considering the results in Section 6.1, all computational results are performed on instances with 50 customers. The six model-strategy combinations have been compared on the entire instance set described in Section 5. The results are shown by comparing the performance of the two models (deterministic vs. stochastic) and the three reoptimization strategies (R, RD and RDW), and of the myopic approach. An analysis of the solutions is also carried out. As the instance generation process is dependent on the sampling of random variables, five different seeds are generated for each combination of $\mathcal{I}$, β, and δ. The results in this section are computed as averages over these five different scenarios.

Performance of models and strategies

We start the analysis by studying the value of reoptimization, i.e., to investigate how the reoptimizaton approach that dynamically solves the problem taking benefit of the updated information behaves with respect to a static approach. In particular, the reoptimization strategies presented in Section 3.1 are compared with a solution approach in which the problem in solved at time 0 and never reoptimized afterwards. Thus, the value of the solution of this latter approach corresponds to the actual value determined at time 0, i.e., the solution is constructed on the basis of the estimations of the release dates at time 0, and the corresponding value depends on the actual realizations of the release dates. Tests are made on 9 instances where the customer locations are the ones of instance C101. Concerning release dates, one instance for each value of δ and β is tested. The results are reported in Table 6 which shows the gap of the solution of the static approach with respect to the best solution determined by the reoptimization approach for the deterministic and the stochastic model, respectively. The results show that reoptimization is indeed beneficial providing an average improvement of solution quality of 10.06% and 14.01% for the deterministic and the stochastic model, respectively. In addition, the table shows that benefits increases with a higher degree of dynamism ($\delta = 1$) and with a wider dispersion of release dates ($\beta = 1.5$).

	% gap Deterministic	% gap Stochastic
$\delta = 0$	3.13	6.81
$\delta = 0.5$	13.14	16.85
$\delta = 1$	13.92	18.36
$\beta = 0.5$	7.02	11.35
$\beta = 1$	10.10	14.48
$\beta = 1.5$	13.07	16.21
Avg.	10.06	14.01

Table 6: Results of the static approach vs best result found by the respective model in the reoptimization approach.

Given the assessment of the value of reoptimization provided by the results above, we then focused on detailed analysis of the behavior of the algorithms presented in Section 3.

Table 7 shows the average of the percentage gap of each solution from the best solution found by all model-strategy combinations. In the top left corner the performance of the two models is shown with respect to the parameter δ. On average, the stochastic model is shown to find better solutions across all values of δ. A higher rate of dynamic customers appears to have a negative impact on the performance of the two models: the percentage gap of the deterministic model shows a positive trend with respect to δ, while the performance of the stochastic model worsens as δ goes from 0 to 0.5 and shows only a slight reduction when δ is increased to 1. The bottom left corner reports the performance of the models with different values of β. Again, the stochastic model finds on average the best solution for all tested values. Interestingly, $\beta = 1$ appears to be the value for which the stochastic model has the largest advantage over the deterministic one. This could be explained by the fact that $\beta = 1$ is the case in which it is harder to find the best trade-off between serving customers and waiting at the depot, as no component of the objective function, i.e., the waiting time at the depot caused by the release dates and the cost of traveling, outweighs the other. It appears that, by exploiting the full probabilistic information, the stochastic model better copes with the increased challenge.

The top central part of Table 7 allows us to analyze the different reoptimization strategies for different values of δ. It is shown that the quality of the solution improves as the number of reoptimizations increases and the benefit increases for an increasing value of δ. The bottom central part shows a comparison of the three strategies for each value of β. The RDW strategy is the one finding the best solution across each value of β. Furthermore, with the sole exception of the R strategy, the case with $\beta = 1$ appears to be the most challenging one, reflecting the result found when comparing the two models.

To show the benefits of the proposed reoptimization approach, the instances have also been solved with the myopic approach. The results are reported in the rightmost column

of Table 7. On average, the solutions found by the myopic approach are shown to be more than 12% worse than the best solution found. The performance of the myopic approach improves as β increases. This is explained by the fact that as β increases, it gets more beneficial to serve customers in shorter routes, i.e., serving less customers per route with a higher number of routes and reducing waiting times.

	Avg. Deterministic	Avg. Stochastic	Avg. R	Avg. RD	Avg. RDW	Avg. Myopic
$\delta = 0$	2.69	0.72	1.77	1.68	1.67	11.96
$\delta = 0.5$	2.96	1.40	2.35	2.17	2.02	14.74
$\delta = 1$	3.10	1.34	3.03	2.40	2.25	10.85
$\beta = 0.5$	1.91	1.28	1.67	1.61	1.51	24.22
$\beta = 1$	3.79	0.90	2.37	2.56	2.11	11.10
$\beta = 1.5$	3.05	1.29	2.40	2.09	2.02	2.23
Avg.	2.92	1.16	2.14	2.09	1.88	12.52

Table 7: Average percentage gap from the best solution.

More detailed results of the average percentage gap are reported for each model-strategy pair and for the myopic approach in Table 8. It is worth noting that, although the results of the myopic approach are on average worse than the ones found by the reoptimization approach, the case when $\delta = 0$ and $\beta = 1.5$ is best solved by serving customers as soon as their parcel is delivered to the depot. This is due to the fact that when release dates are sufficiently spread out in time to make serving each customer individually become the best solution, the myopic approach finds the best solution. Table 8 also highlights how, on average, the best results are found by the stochastic model in combination with the RDW approach.

Analysis of the solutions

To allow for a better comparison of the proposed models and to better understand the differences in the solutions found by the different models and strategies, Tables 9, 10, 11, 12, and 13 are presented, replicating the structure of Table 7. Table 9 reports the average total time required to run each simulation, Table 10 reports the number of reoptimizations carried out, Table 11 reports the average number of routes across all tested instances, Table 12 shows the average reoptimization time per route, and Table 13 the average percentage of waiting time over the total cost of the solution for each model and reoptimization strategy and for all values of δ and β. The results of the myopic approach are omitted when not relevant.

Table 9 reports the average total time required to run each simulation. Notably, when comparing the two models for different values of δ, it is shown that the stochastic model consistently requires about 5 minutes more than the deterministic one. When comparing the two models with respect to the parameter β, the time required on average for each

		Deterministic			Stochastic			
δ	β	R	RD	RDW	R	RD	RDW	Myopic
0	0.5	0.61	0.41	0.46	0.76	0.82	0.73	25.19
	1	4.79	4.55	4.24	0.48	0.22	0.46	10.69
	1.5	2.92	3.07	3.19	1.05	1.05	0.92	0
	Avg.	2.77	2.67	2.63	0.76	0.70	0.70	11.96
0.5	0.5	2.29	2.33	2.31	1.76	1.03	0.99	25.89
	1	4.10	3.69	2.97	0.85	2.20	1.49	15.09
	1.5	3.15	2.57	3.19	1.92	1.21	1.18	3.26
	Avg.	3.18	2.87	2.82	1.51	1.48	1.22	14.74
1	0.5	2.47	3.44	2.85	2.11	1.61	1.71	21.58
	1	3.12	3.57	3.06	0.86	1.12	0.42	7.53
	1.5	3.71	2.71	2.94	1.65	1.94	0.68	3.43
	Avg.	3.10	3.24	2.95	1.54	1.56	0.94	10.85
Avg.		3.02	2.93	2.80	1.27	1.24	0.95	12.52

Table 8: Detailed results for the average percentage gap from the best solution.

simulation increases for both models. It is shown, however, that the difference in the required time for the stochastic model to the deterministic one decreases as β increases. The total time is larger when the reoptimization is carried out more frequently. In particular, the RDW strategy takes more than twice the time required by the R strategy.

	Avg. Deterministic	Avg. Stochastic	Avg. R	Avg. RD	Avg. RDW
$\delta = 0$	40.36	46.29	23.20	36.11	70.65
$\delta = 0.5$	32.12	37.04	22.28	33.55	47.90
$\delta = 1$	37.79	43.57	25.64	40.04	56.35
$\beta = 0.5$	16.62	26.39	14.21	21.39	28.92
$\beta = 1$	37.69	43.37	25.10	38.95	57.54
$\beta = 1.5$	55.96	57.13	31.81	49.36	88.45
Avg.	36.75	42.30	23.71	36.57	58.30

Table 9: Average time required for each simulation (in minutes).

Table 10 shows a comparison of the models and strategies with respect to the number of reoptimizations. The two models show a very similar number of reoptimizations across all values of δ and β. As expected, the number of reoptimizations increases from the R strategy to the RD and RDW strategies. Such increase is shown to be correlated with the

parameter β. This is due to the fact that the more release dates are spread out in time, the more it is beneficial to serve customers in a higher number of shorter routes. As the reoptimization epochs are dependent on the number of routes and on the starting time of such routes, when β increases more reoptimizations are performed.

	Avg. Deterministic	Avg. Stochastic	Avg. R	Avg. RD	Avg. RDW
$\delta = 0$	10.71	10.63	5.98	9.12	16.90
$\delta = 0.5$	9.07	9.11	5.89	8.79	12.58
$\delta = 1$	10.41	10.69	6.66	10.10	14.88
$\beta = 0.5$	6.37	6.33	4.33	6.27	8.46
$\beta = 1$	10.49	10.78	6.61	10.12	15.18
$\beta = 1.5$	13.32	13.31	7.59	11.63	20.73
Avg.	10.06	10.14	6.18	9.34	14.79

Table 10: Average number of reoptimizations.

Table 11 reports a comparison of the number of routes for each model and strategy. It is observed that, on average, the stochastic model finds solutions where the distribution is completed with less routes than the ones found by the deterministic model. This result holds across all values of δ. When the models are compared with respect to the parameter β it only holds for $\beta = 1.5$. Table 11 confirms the positive correlation between the parameter β and the average number of routes in each solution. This result also validates the increasing competitiveness of the myopic approach, highlighted in Table 7. While this result might seem contradictory with the fact that the solutions found by the myopic approach have on average less routes than the ones found with any of the proposed reoptimization strategies, it must be noted that the case in which the myopic approach performs better, i.e., $\beta = 1.5$, is also the case in which the difference in the number of routes with any of the reoptimization strategies is minimum. Finally, the number of routes appears to be consistent across the three different reoptimization strategies.

Table 12 allows us to investigate the behavior of the ratio between the total time required to run each simulation over the number of routes. This value can be interpreted as the average time spent solving the model in order to obtain the next route to be implemented. When comparing the two models, the stochastic one is shown to require more time per route across all values of β and δ. Looking at the results for different values of δ, the case of $\delta = 0$ is the one requiring the longest time both when considering the two models and when comparing the three strategies. Both the two models and the three strategies show a positive trend with respect to the parameter β.

Further analysis of the structure of the solutions found has been carried out by comparing the percentage of time spent waiting at the depot over the total time required for the distribution. The results are reported in Table 13. It can be observed that the stochastic

	Avg. Deterministic	Avg. Stochastic	Avg. R	Avg. RD	Avg. RDW	Avg. Myopic
$\delta = 0$	6.11	5.79	5.98	5.86	6.01	5.25
$\delta = 0.5$	6.13	5.93	5.92	5.96	6.21	4.97
$\delta = 1$	7.08	6.33	6.66	6.60	6.85	5.38
$\beta = 0.5$	4.15	4.53	4.33	4.33	4.36	3.70
$\beta = 1$	6.58	6.67	6.61	6.58	6.69	5.08
$\beta = 1.5$	8.59	6.84	7.62	7.52	8.02	6.82
Avg.	6.44	6.01	6.19	6.14	6.36	5.20

Table 11: Average number of routes.

	Avg. Deterministic	Avg. Stochastic	Avg. R	Avg. RD	Avg. RDW
$\delta = 0$	6.28	7.90	3.75	6.00	11.51
$\delta = 0.5$	5.12	6.28	3.61	5.63	7.87
$\delta = 1$	5.19	6.82	3.71	6.00	8.31
$\beta = 0.5$	4.06	6.01	3.19	4.97	6.94
$\beta = 1$	5.75	6.58	3.74	5.94	8.81
$\beta = 1.5$	6.79	8.41	4.14	6.72	11.93
Avg.	5.53	7.00	3.69	5.88	9.23

Table 12: Average time (in minutes) over the number of routes.

model has on average a higher rate of waiting time over the deterministic model. This result confirms those found when looking at the number of routes of the solutions: as the stochastic model tends to serve customers with less routes compared to the deterministic model, more time is spent waiting at the depot for the parcels of the customers. The results found for the myopic approach reflect its strategy of serving customers as soon as possible: as this approach only waits at the depot if there are no customers that can be served, the percentage waiting time is shown to be lower compared to the solutions found when solving the models considering the information available about future release dates.

Finally, an additional set of tests have been performed to evaluate the performance of the proposed solution approaches when the distribution of the release dates is different from the one used in the former tests. In particular, a 'double triangular' distribution is assumed for the release dates, as shown in Figure 6, where the two peaks mimic 'burst of activities' where the realizations of release dates concentrate. The idea is to simulate practical situations where there is a concentration of deliveries during peak hours. Tests are made on 9 instances where the customer locations are the ones of instance C101. Concerning release dates, we tested one instance for each value of δ and β. The results are shown in Table 14. We note that the change of the distribution of release dates has

	Avg. Deterministic	Avg. Stochastic	Avg. R	Avg. RD	Avg. RDW	Avg. Myopic
$\delta = 0$	9.13	11.00	9.79	10.23	10.18	1.67
$\delta = 0.5$	5.88	6.53	6.08	6.13	6.40	1.43
$\delta = 1$	6.64	8.37	7.49	8.02	7.00	2.28
$\beta = 0.5$	2.81	2.13	2.32	2.51	2.57	0
$\beta = 1$	7.91	6.93	7.13	7.87	7.25	0.77
$\beta = 1.5$	10.93	16.85	13.91	14.00	13.76	4.60
Avg.	7.22	8.63	7.79	8.13	7.86	1.79

Table 13: Average percentage waiting time.

no impact on the performance of the algorithms which follows the same trend as for the results presented above.

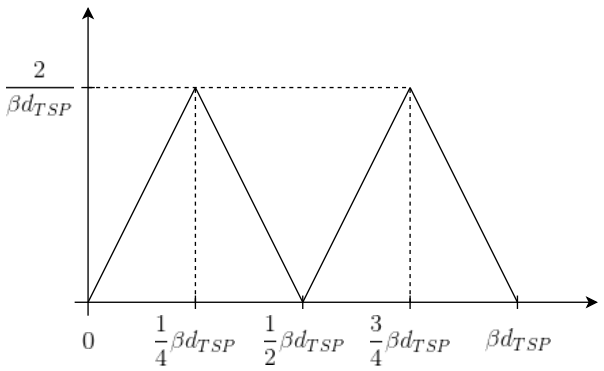


Figure 6: The double triangular distribution of release dates.

7 Conclusions

In this paper the dynamic traveling salesman with stochastic release dates (DTSP-srd) is introduced. A solution approach is proposed based on reoptimization. Three strategies are introduced to define the reoptimization epochs, with increasing frequency of reoptimization, and two models are proposed for the solution of the problem at each epoch. The first one considers a point estimation of the release dates and the second one makes use of the entire probabilistic information available. The models are solved with an iterated local search heuristic. Extensive computational tests are carried out on a large instance set. The results show that a more frequent reoptimization provides better results across all tested instances. Furthermore, it has been observed that, on average, better solutions are found when the problem is solved with the stochastic model. In addition, a myopic approach serving customers as soon as possible is shown to perform more than 12% worse than the best solution found by the two models. The deterministic and stochastic model have an

	Deterministic			Stochastic			
	R	RD	RDW	R	RD	RDW	Myopic
$\delta = 0$	6.80	1.18	0.25	2.16	1.57	0.18	23.60
$\delta = 0.5$	1.19	1.12	2.30	1.26	0.63	0.98	9.26
$\delta = 1$	1.82	0.11	0.37	0.00	0.00	0.59	1.23
$\beta = 0.5$	0.76	0.17	0.60	1.08	0.14	1.06	8.61
$\beta = 1$	6.46	1.93	1.18	1.57	1.57	0.00	11.24
$\beta = 1.5$	2.58	0.32	1.14	0.77	0.49	0.68	14.24
Avg.	3.27	0.80	0.98	1.14	0.73	0.58	11.36

Table 14: Results with the double triangular distribution.

average percentage gap from the best solution found across the three dynamic strategies of 2.92% and 1.16%, respectively. When considering this result together with the number of iterations performed by the heuristic for each of the two models, the deterministic model appears to be a good compromise, especially when the number of customers increases.

The analysis of the results offers some interesting insights. First, they show that the dynamic behavior of release dates has an important impact on the type of solution approach that should be used to solve the problem. In fact, on one side, a static approach provides non satisfactory results. On the other side, a reoptimization approach which subsequently solves the problem on the basis of the updated information is able to provide good quality solutions, and the quality is proportional to the frequency of reoptimization. However, this comes at the expense of an increasing computing time. Thus, the decision maker has to evaluate the trade-off between quality and time. Similar considerations are related to the kind of model used to solve the problem, i.e., deterministic versus stochastic. The results show that the stochastic model provides better results. However, the computational burden is much higher than the one for the deterministic model. Thus, again, the trade-off must be evaluated.

Several research directions remain open. As in the proposed approach no action is considered when the vehicle is not at the depot, such option could be considered by reoptimizing while the vehicle is at customers location and allowing for an anticipated return to the depot, as a further effort towards a better distribution. Furthermore, different probability distributions could be considered for the release dates and the case where the distribution assumed is different from the actual distribution of the release dates could be investigated. The DTSP-srd considers one uncapacitated vehicle to perform the distribution. A natural extension of the problem would allow multiple capacitated vehicles to be deployed.

References

C Archetti, D Feillet, and M G Speranza. Complexity of routing problems with release dates. *European Journal of Operational Research*, 247:797–803, 2015.

C Archetti, D Feillet, A Mor, and M G Speranza. An iterated local search for the traveling salesman problem with release dates and completion time minimization. *Computers & Operations Research*, 98:24–37, 2018.

D Cattaruzza, N Absi, and D Feillet. The multi-trip vehicle routing problem with time windows and release dates. *Transportation Science*, 50:676–693, 2016.

Y Y Fan, R E Kalaba, and J E Moore II. Arriving on time. *Journal of Optimization Theory and Applications*, 127:497–513, 2005.

M Gendreau, O Jabali, and W Rei. 50th anniversary invited article—future research directions in stochastic vehicle routing. *Transportation Science*, 50:1163–1173, 2016.

K Helsgaun. An effective implementation of the Lin–Kernighan traveling salesman heuristic. *European Journal of Operational Research*, 126:106–130, 2000.

I Kaparias, M G H Bell, and H Belzner. A new measure of travel time reliability for in-vehicle navigation systems. *Journal of Intelligent Transportation Systems*, 12:202–211, 2008.

M A Klapp, A L Erera, and A Toriello. The one-dimensional dynamic dispatch waves problem. *Transportation Science*, 52:402–415, 2018a.

M A Klapp, A L Erera, and A Toriello. The dynamic dispatch waves problem for same-day delivery. *European Journal of Operational Research*, 271:519–534, 2018b.

X Li, P Tian, and S C H Leung. Vehicle routing problems with time windows and stochastic travel and service times: Models and algorithm. *International Journal of Production Economics*, 125:137–145, 2010.

S Lin and B W Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, 21:498–516, 1973.

C Malandraki and M S Daskin. Time dependent vehicle routing problems: Formulations, properties and heuristic algorithms. *Transportation science*, 26:185–200, 1992.

M L Pinedo. *Scheduling: theory, algorithms, and systems*. Springer, 2016.

W B Powell, H P Simao, and B Bouzaiene-Ayari. Approximate dynamic programming in transportation and logistics: a unified framework. *EURO Journal on Transportation and Logistics*, 1:237–284, 2012.

M L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

G Reinelt. TSPLIB - a traveling salesman problem library. *ORSA journal on computing*, 3:376–384, 1991.

D Reyes, A L Erera, and M W P Savelsbergh. Complexity of routing problems with release dates and deadlines. *European Journal of Operational Research*, 266:29–34, 2018.

U Ritzinger, J Puchinger, and R F Hartl. A survey on dynamic and stochastic vehicle routing problems. *International Journal of Production Research*, 54:215–231, 2016.

R A Russell and T L Urban. Vehicle routing with soft time windows and erlang travel times. *Journal of the Operational Research Society*, 59:1220–1228, 2008.

N Secomandi and F Margot. Reoptimization approaches for the vehicle-routing problem with stochastic demands. *Operations Research*, 57:214–230, 2009.

M M Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35:254–265, 1987.

D Taş, N Dellaert, T Van Woensel, and T De Kok. Vehicle routing problem with stochastic travel times including soft time windows and service costs. *Computers & Operations Research*, 40:214–224, 2013.

M W Ulmer, B W Thomas, and D C Mattfeld. Preemptive depot returns for a dynamic same-day delivery problem. *Technical Report, Technische Universität Braunschweig*, pages 1–39, 2016.

W.J.A. van Heeswijk, M.R.K. Mes, and J.M.J. Schutten. The delivery dispatching problem with time windows for urban consolidation centers. *Transportation science*, 53:203–221, 2017.

S A Voccia, A M Campbell, and B W Thomas. The same-day delivery problem for online purchases. *Transportation Science*, 2017. doi: https://doi.org/10.1287/trsc.2016.0732.