

Look-ahead strategies for dynamic pickup and delivery problems

Martijn Mes · Matthieu van der Heijden · Peter Schuur

Published online: 1 August 2008
© The Author(s) 2008

Abstract In this paper we consider a dynamic full truckload pickup and delivery problem with time-windows. Jobs arrive over time and are offered in a second-price auction. Individual vehicles bid on these jobs and maintain a schedule of the jobs they have won. We propose a pricing and scheduling strategy based on dynamic programming where not only the direct costs of a job insertion are taken into account, but also the impact on future opportunities. Simulation is used to evaluate the benefits of pricing opportunities compared to simple pricing strategies in various market settings. Numerical results show that the proposed approach provides high quality solutions, in terms of profits, capacity utilization, and delivery reliability.

Keywords Dynamic vehicle routing · Multi-agent systems · Auctions · Scheduling

1 Introduction

An important trend in external logistics is the increased focus on real-time decision making due to continuing developments in telecommunication and information technologies. These technologies, such as Internet and global positioning systems (GPS), enhance the planning capability of freight carriers and provide the necessary information required to perform real-time decision making. Also, the possibilities of internet trade of products in the business-to-business area will increase the complexity of the

This research is supported by the BSIK project “Transition Sustainable Mobility” (TRANSUMO).

M. Mes (✉) · M. van der Heijden · P. Schuur
Department of Operational Methods for Production and Logistics,
School of Management and Governance, University of Twente,
P.O. Box 217, 7500 AE Enschede, The Netherlands
e-mail: m.r.k.mes@utwente.nl

physical distribution in the near future. As a consequence, new operations research techniques are needed that allow for real-time decision making in a dynamic environment. Dynamic environments are characterized by frequent and unpredictable modifications in the relevant planning information and the ability (or even necessity) to update the planning based on this additional information. For example, carriers may know only a fraction of the shipments to be served when the first plan is constructed whereas additional (rush) transportation jobs arrive during execution of the original plan. A proper transport planning approach should be able to construct initial plans taking into account all such uncertainties and to update the plans reacting on real-time information updates.

In this paper we specifically focus on decisions regarding a single vehicle. The motivation for this focus is that the single vehicle may be a part of a distributed planning system, or more specifically of a multi-agent system (MAS). Such a system consists of a group of intelligent and autonomous computational entities (agents) which coordinate their capacities in order to achieve certain (local or global) goals (Wooldridge 1999). An example of a MAS for the control of multiple vehicles can be found in Mes et al. (2007). Here each vehicle is represented by an agent who is responsible for the planning and scheduling of its corresponding vehicle. Coordination is achieved through the use of an auction protocol. We compared the performance of this agent-based system with two traditional scheduling heuristics. We find that a properly designed multi-agent approach performs as good as or even better than the two traditional methods. In this paper we aim to enhance the performance of this agent-based transportation system by improving the pricing and scheduling techniques of individual vehicles. Besides this motivation from the perspective of a distributed planning system, the single vehicle case may serve as a base for the multi-vehicle problem.

In the literature, our transportation problem is known as a dynamic pickup and delivery problem with time windows, full truckloads, and stochastic arrivals of jobs. We consider a special case where vehicles participate in sequential transportation procurement auctions. In recent years, there has been a surge of interest in the use of online market mechanisms to match supply and demand for transportation services dynamically. Especially due to the development of Internet sites that match shippers' demand for transportation with the transport capacity of carriers. Besides that, auctions are known to be an efficient way to allocate items among agents, both in terms of process and outcome (Sandholm 2002). As a result, auctions are frequently proposed and used as a coordination mechanism for distributed scheduling, see, e.g., (Wellman et al. 2001).

Given the transportation procurement auction, vehicles not only have to decide about routing and scheduling, but also on acceptance decisions through pricing of jobs. Also from the perspective of multiple vehicles we need a proper pricing mechanism to optimize the system-wide performance in terms of logistical costs. In this paper we focus on three key questions. First, how can we use information on historic job patterns and historic auction results to improve bid pricing? Second, how can we use this information to improve planning and scheduling? Third, what is the impact of such additional intelligence on the profit of an individual vehicle as well as on the overall logistical performance?

The remainder of this paper is structured as follows. In the next section, we give an overview of related literature. Our model of the transportation network is presented in Sect. 3, and the scheduling and bid pricing behavior of the vehicles in Sect. 4. Our solution method to include probabilistic knowledge about future job arrivals is presented in Sect. 5. In Sect. 6 we describe how the parameters can be estimated. The experimental settings and simulation results are presented in Sect. 7. We end up with conclusions in Sect. 8.

2 Literature

Our problem is well known in the area of dynamic vehicle routing problems (DVRP). Here a number of vehicles have to satisfy transportation requests that arrive dynamically over time. This requires real-time replanning in order to include the new jobs in the vehicle schedules.

Although many papers have been devoted to the DVRP, there are still some issues that have not been addressed yet (Ghiani et al. 2003). Especially regarding look-ahead policies that incorporate the future consequences of certain decisions. In the literature one can find two main approaches to cope with the problem changes caused by new customer arrivals. The two strategies are reoptimization and dispatching (Branke et al. 2005).

The first approach is to renounce planning to a certain extent, and, instead of scheduling all known tasks, only decide on a vehicle's next task. Here, decisions that take into account future job arrivals, are where to wait and for how long. We indicate this type of decisions by *operational waiting decisions*.

An early example can be found in Bertsimas and van Ryzin (1991). They consider a dynamic traveling repairman problem (DTRP) where service demands arrive according to a Poisson process at uniformly distributed locations in a Euclidean plane. They show for the case of a single vehicle and light traffic conditions, that it is optimal to reposition the vehicle to the center of the service region whenever there are no customers left to be served. Returning to the center anticipates future customer arrivals by positioning the vehicle so that the expected distance to the next arriving customer is minimized. Similar results are shown in Bertsimas and van Ryzin (1993) for the multiple-vehicle traveling repairman problem and in Swihart and Papastavrou (1999) for the single vehicle pickup and delivery problem.

Larsen et al. (2004) consider the dynamic traveling salesman problem. The goal is to find a minimum-costs tour through a set of dynamic requests with soft time-windows. The issue is to relocate empty vehicles to predefined resting locations in anticipation of future demand. They consider the following online policies: (1) the vehicle goes to the nearest resting location, (2) the vehicle goes to the resting location that belongs to a region with the largest arrival rate, (3) the vehicle goes to the resting location that belongs to a region with the highest expected number of customers within a certain time period depending on future obligations of this vehicle.

Thomas and White (2004) consider the case of a single vehicle which has to travel from a given origin to a given destination. Within the network there are a few known locations of potential customers that might issue a request while the vehicle is under

way. The vehicle anticipates these possible future customer requests both by waiting at a location and by visiting non-customer locations. They present an optimal policy for route construction by modeling their routing problem as a Markov decision process. In a later paper (Thomas 2007), the waiting decisions—when and where to wait—are studied in more detail. Based on analytical results for the case with a single dynamic customer, they propose a real-time waiting heuristic and demonstrate its effectiveness compared with a series of other intuitively appealing waiting heuristics.

Ichoua et al. (2006) consider a DVRP with soft time-windows for serving customers and a hard time-window for returning at the depot. The problem consists of using probabilistic knowledge about future request arrivals to minimize the cost of vehicle routes. The cost of a route depends on the lateness penalties and on the total distance traveled. The original algorithm, a parallel tabu search heuristic with adaptive memory, is extended by allowing a vehicle to wait at its current position if the probability of new, nearby requests is high.

The main distinction between the papers mentioned above and our research, is that we schedule multiple jobs in advance. As a consequence we have to anticipate future job arrivals in the sequencing of jobs currently in the schedule.

The second approach to handle real-time events in DVRP is to reoptimize whenever a new job arrives. The main decision here is to anticipate future job arrivals through scheduling of jobs. We indicate these decisions by *tactical waiting decisions*.

Mitrović-Minić and Laporte (2004) consider the uncapacitated dynamic pickup and delivery problem with time windows. They present a methodology that captures the need to take future events into consideration, even when the events are not stochastically modeled or predicted. They examine the benefits of several tactical waiting strategies on the total detour and number of vehicles, where they distinguish between the impact of decisions on the short-term and the long-term basis. Their work differs from ours because we do not focus on detour but profits (i.e., probability and profitability of new job insertions).

Branke et al. (2005) consider a DVRP where one additional customer arrives at a beforehand unknown location when the vehicles are already under way. Each vehicle has a fixed route along different locations. They optimize the waiting times of the vehicles at each location such that the probability that the new customer can be inserted is maximized. The main difference is that we determine the optimal pickup time for one job at a time, but evaluate multiple future jobs insertions and their impact on the profitability.

Besides the differences mentioned above, there are two additional differences. First, we consider the combination of tactical and operational waiting decisions. Second, we also focus on acceptance decisions through dynamic pricing of jobs. All these decisions are supported by the same opportunity valuation method, which therefore does not require additional computation time.

Another related line of research is on dynamic fleet management problems (DFMP), or more generally the dynamic assignment problem. These problems ask for a dynamic assignment of resources (trucks) to tasks (loads). Truly stochastic models decompose the DFMP with respect to time periods and assess the impact of the current decisions on the future through a recourse or value function. An early example can be found in Powell et al. (1988), who study a dynamic assignment problem where a fleet of

vehicles is assigned to a set of locations with dynamically occurring demands. They show that it is advantageous to take forecasted demands into account when deciding where the vehicles should drive next, compared with a model that only reacts after new demands have arrived. More recent examples can be found in [Carvalho and Powell \(2000\)](#), [Godfrey and Powell \(2002\)](#). We cannot use the DFMP algorithms directly, because jobs have to be accepted early to avoid losing them to competitors, and jobs are scheduled in a distributed manner by the vehicle agents. Also the price of a job is not given externally but subject to negotiation. Moreover, the arrival intensity of jobs at a company is not described by an exogenous information process, but can be influenced by better repositioning of vehicles.

An interesting and closely related line of research can be found in [Figliozzi \(2004\)](#), [Figliozzi et al. \(2006\)](#). They present an optimal bid analysis for carriers participating in sequential transportation procurement auctions. Each arriving job triggers an auction where carriers compete with each other to win the right of servicing the job. The bid price of a carrier consists of the marginal costs (in terms of distance traveled by its fleet of vehicles to service all jobs) of adding the new job to its current set of jobs. However, winning a job also has an effect on the marginal costs of future jobs because (1) it reduces the carriers' capacity and (2) it changes the current schedule. To capture this effect they use *one-step* look-ahead opportunity costs, defined as the expected increase in marginal costs of the next job to be auctioned due to accepting the current job. Using simulation, they show that the carrier that accounts for opportunity costs can significantly improve its profitability compared to the naive carrier. The main differences with our work are (1) we work with opportunity costs that are defined over *multiple* jobs to be acquired within a certain time horizon, and (2) these functions are used to price a new job, but also to sequence the jobs, and to support pro-active move decisions (moves to attractive locations if vehicles are idle).

3 Network description

Our transportation network is a directed graph $(\mathcal{N}, \mathcal{A})$, i.e., it consists of a set of nodes \mathcal{N} and a set of arcs \mathcal{A} connecting these nodes. Jobs, consisting of unit loads (full truckloads), have to be transported between these nodes. We assume that jobs arrive one-by-one according to a stationary Poisson process with arrival intensity λ_{kl} of jobs from node k to node l . We define a job φ by an announcement time $a(\varphi)$, an origin node $o(\varphi)$, a destination node $d(\varphi)$, and a latest pickup time $e(\varphi)$ of the load at its origin. The announcement time is the time at which the job is offered in an auction. We introduce a soft time-window $z(\varphi) = e(\varphi) - a(\varphi)$ within which transportation should be started.

A set of vehicles \mathcal{V} , not necessarily identical, is available to transport the loads. The time to transport a load from node i to node j is given by τ_{ij}^f . This includes travel time, and the handling time to load- and unload the job. The time to drive empty from node i to node j is given by τ_{ij}^e . Both times are deterministic and vehicle independent. We use the shorthand notation τ_{ikl} for $\tau_{ik}^e + \tau_{kl}^f$. We further assume that a job in process cannot be interrupted (no preemption).

Objective of each vehicle is to maximize its profits. These profits are given by the income from all transportation jobs minus the transportation costs and costs for lateness. We consider two costs functions, namely the travel costs $c^r(t)$ as function of the travel time t and the penalty costs $c^p(t)$ in case of tardiness ($t > 0$) with respect to the latest pickup time.

Jobs are offered in sequential transportation procurement auctions. In these auctions, shippers typically put out a request for quotes from a set of carriers (Song and Regan 2002). This process is similar to a simple sealed-bid auction in which each bidder submits a sealed bid for a single job. Without loss of generality, we choose here for a reverse second-price sealed-bid auction, also known as the Vickrey auction. In this auction mechanism the bidder with the highest bid will receive the object at the price of the second highest bid. The Vickrey auction also has been widely used for MASs because it forces bidders to bid their true valuation of the object, and thus avoids many bidding problems. In this paper we ignore possible limitations of the Vickrey auction (see, e.g., Brandt and Weiß 2002) by assuming that individual bidders try to maximize their profits without caring for the profits made by the others. As a consequence we assume that bidders bid their true valuation, which is given by the estimated costs of a new job. Note that bidding the true valuation is not as simple as it seems as we aim to take into account opportunity costs in this paper. As it will appear, this is not a trivial matter. We further assume that the final price—the second best bid—is made known to all participants. Generalization toward a closed auction, where bidders only receive information about whether they win or lose an auction, is straightforward.

4 Vehicle scheduling and bid calculation

At each point in time, a vehicle has a job schedule, i.e., a list of jobs with scheduled starting times. Vehicles use this schedule only to support their job sequencing decisions and bid pricing decisions. Vehicles are not restricted by these pickup times, but can simply decide to insert new jobs or to wait at some node after delivery of a job.

Formally, we define a schedule Ψ by an ordered list of 2-tuples $\Psi_n = (\varphi_n, \omega_n)$ where φ_n refers to the n th job in the schedule and ω_n to the scheduled pickup time of this job. The loaded move for job φ_n goes from $o(\varphi_n)$ to $d(\varphi_n)$, starting at time ω_n and being delivered at time $\omega_n + \tau_{o(\varphi_n)d(\varphi_n)}^f$. In the remainder we denote this delivery time of job n by ρ_n . We further denote (1) the number of jobs in a schedule by N , (2) the destination of the last job in the schedule by *schedule destination*, and (3) the time until the expected arrival time at the schedule destination by *length of a schedule*. All times mentioned here and in the sequel are relative to the current time θ .

To avoid excessive computation times we consider an insertion scheduling heuristic. Here a vehicle contemplates the insertion of a new job at any position in the current schedule without altering the order of execution for the other jobs. Given a sequence of jobs, each job is scheduled as early as possible. A new job is either inserted between two jobs currently in the schedule or added after the last job in the schedule. We denote these two cases by insertion in a *gap* and insertion in the *end-gap*, respectively.

The end-gap denotes the period given by the difference between the planning period T and the length of a schedule. A gap denotes the potential time period between two

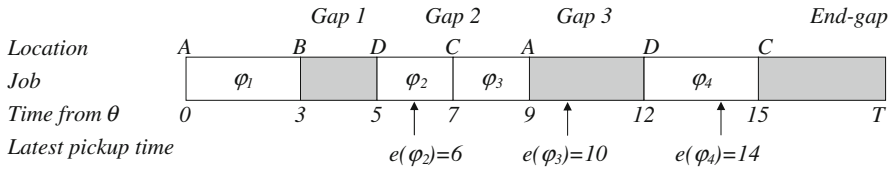


Fig. 1 Example of a vehicle schedule

consecutive jobs that can be used for new job insertions. The node at the beginning and end of a gap will be referred to as *start-node* and *end-node*, respectively. We define t_n as the flexibility of the n th gap (the gap after job n). This flexibility consists of an empty travel time τ_{ij}^e from the start-node i to the end-node j , plus a time slack s_n . This time slack is the maximum amount of time that this job can be postponed without causing an increase in penalties for this job or one of the succeeding jobs. We calculate the gap flexibilities recursively as shown in Algorithm 1.

Algorithm 1 Calculating the gap flexibilities

```

init:  $s_N = \infty$ 
for  $n = N - 1$  down to 1 do
     $s_n = \min((e(\varphi_{n+1}) - \omega_{n+1})^+, s_{n+1})$ 
     $t_n = s_n + (\omega_{n+1} - \rho_n)$ 
end;
```

An illustration of a possible schedule can be found in Fig. 1. Here, the gaps consists of the required empty travel times between the locations. The time slack of the third, second, and first gap are given by $s_3 = \min((14 - 12)^+, \infty) = 2$, $s_2 = \min((10 - 7)^+, 2) = 2$, and $s_1 = \min((6 - 5)^+, 2) = 1$, respectively. The gap flexibilities are now given by: $t_3 = 2 + (12 - 9) = 5$, $t_2 = 2$, and $t_1 = 1 + (5 - 3) = 3$.

Each vehicle wants to maximize its profits. However, estimation of the expected profit of a new job is not straightforward because the destination of this job may have an effect on the expected empty travel distance for the next job and the likelihood of winning a next job. The same holds for the origin which has an effect on the insertion of future jobs before this job. Therefore we focus on maximizing the profits within a certain planning horizon T . We denote the expected profits during period T , given a schedule Ψ , by $V(\Psi, T)$. These profits consist of the payments for all jobs in schedule Ψ , minus the costs for these jobs, plus the expected profits for future jobs to be inserted in schedule Ψ (see Sect. 5 for a formal expression).

When an auction for a new job φ is started, each vehicle creates a temporal schedule $\Psi \cup \varphi$ combining its current schedule with the new job in such a way that the value $V(\Psi \cup \varphi, T)$ is maximized. Note that at this point we do not know the payment for the new job which is therefore set to zero. To construct a temporal schedule, a vehicle evaluates all possible insertions of the new job φ in the current schedule Ψ . Because the first job of the schedule is always in execution, the number of insertion positions equals N . We indicate each alternative schedule by Ψ_φ^n . The temporal schedule is

given by an alternative schedule with the highest profit:

$$V(\Psi \cup \varphi) = \max_{n=1 \dots N} V(\Psi_\varphi^n) \tag{1}$$

A vehicle updates its schedule when (1) an auction for a new job is won and (2) the first loaded move in a schedule has been completed. In the first case, the vehicle replaces its current schedule with the temporal schedule that had been constructed for the auction. In the second case, the vehicle faces an *operational waiting decision* (see Sect. 5.3).

As mentioned in Sect. 3, we focus on bidding true valuations. The true valuation of a new job insertion consists of the increase in transportation and penalty costs, but also of the expected decrease in future profits. The sum of these two costs components is given by the decrease in expected profits $V(\Psi, T)$ due to the insertion of the new job, assuming we did not receive a payment for this new job. Bidding less is not optimal since then winning this job will result in loss of profit. Bidding more reduces the likelihood of winning the shipment while the expected final price remains the same. As a consequence, the optimal bid price $b(\varphi, \Psi)$ for a new job φ in the current vehicle schedule Ψ is given by the value of the current schedule minus the value of the temporal schedule with the new job:

$$b(\varphi, \Psi) = V(\Psi, T) - V(\Psi \cup \varphi, T) \tag{2}$$

As shown in the next section, this bid price not only includes the increase in transportation and penalty costs but also the change in value of future job opportunities. For a more in depth analysis on optimal bid prices we refer to Vickrey (1961), Figliozzi (2004).

5 Value functions

In the previous section we introduced the expected profits $V(\Psi, T)$ during a period T , given the current schedule Ψ . This profit is given by the sum of the payments p_φ for all jobs φ in the schedule Ψ , minus the direct costs for all jobs, plus the value of all gaps between subsequent jobs, plus the value of the end-gap.

The direct costs of a job φ with pickup time ω are given by the travel costs for the loaded move and possibly penalty costs:

$$C^d(\varphi) = c^r \left(\tau_{o(\varphi)d(\varphi)}^f \right) + c^p \left((\omega - e(\varphi))^+ \right) \tag{3}$$

The costs for empty moves are not included in the direct costs because they might be replaced by loaded moves. Instead we include these costs in the value of gaps between subsequent jobs. To quantify the values of gaps we introduce two value functions, namely a gap-value and an end-value. The *gap-value* $V^g(i, j, \sigma, t)$ is the expected profit of all future moves in a gap defined by start-node i , end-node j , an expected time σ until arrival at node i and the gap flexibility t . Note that this value is only defined for $t \geq \tau_{ij}^e$. For future reference we put $V^g(i, j, \sigma, t) = -\infty$ for $t < \tau_{ij}^e$. The

end-value $V^e(i, \sigma, t)$, is the expected profit during a period $t = T - \sigma$ after arrival at schedule destination i at time σ from now on. As an example consider the schedule of Fig. 1. For this schedule, the value of the third gap is given by $V^g(A, D, 9, 5)$ and the end-value is given by $V^e(C, 15, T - 15)$. In the remainder we use the word *time-to-go* to indicate the time σ from now till the arrival at the schedule destination i or the start-node i of a gap.

The expected profit $V(\Psi, T)$ during period T given schedule Ψ is now given by:

$$V(\Psi, T) = \sum_{n=1}^N \left(p_{\varphi_n} - C^d(\varphi_n) \right) + \sum_{n=1}^{N-1} V^g(d(\varphi_n), o(\varphi_{n+1}), \rho_n, t_n) + V^e(d(\varphi_N), \rho_N, T - \rho_N) \tag{4}$$

We also use this expression to calculate the expected profit $V(\Psi \cup \varphi, T)$ of the temporal schedule with the new job φ . Because we do not know the payment for this new job φ , we set p_φ equal to zero.

5.1 Recursive formulation

In this section we derive recursive relations for the value functions V^e and V^g . These relations are described by four types of information: state space, decision set, transition probabilities, and expected rewards. We use the state variables to capture all necessary information to value the future behavior of the system to be controlled. Here we assume that, in each iteration of the recursive relations, a job is scheduled immediately after the job received in the previous iteration. As a consequence, we may represent the state within a gap by $\{i, j, \sigma, t\}$ and the state within an end-gap by $\{i, \sigma, t\}$. Below, we present the recursion for end-values only, since the gap-values are derived in a similar—albeit slightly more complicated—manner. To clarify our presentation we proceed in a number of steps: (1) we present the waiting decisions; (2) we present a partial value function which forms an element of the end-value; (3) we distinguish three cases for winning a job depending on the waiting decisions, provide the partial value function for each case, and derive the end-value function by combining the three partial value functions.

5.1.1 Waiting decisions

Given that subsequent jobs are scheduled immediately after each other, a vehicle will always drive immediately toward the origin of the next job whenever it finishes a job and its schedule is not empty. Whenever it finishes a job and its schedule is empty, it has to decide where to wait. Therefore we consider the decision $\delta(i) \in \mathcal{N}$ to move pro-actively to node $\delta(i)$ directly upon arrival at node i . If $\delta(i) = i$, the decision is to wait at the current node i . In the remainder we will use the shorthand notation δ instead of $\delta(i)$. Of course, a vehicle will only make this decision when it is waiting at some node ($\sigma = 0$). So if its current state is given by $\{i, 0, t\}$ then its next state, immediately after making the decision δ , is given by $\{\delta, \tau_{i\delta}^e, t - \tau_{i\delta}^e\}$.

5.1.2 Partial value functions

To derive a recursion for the end-values $V^e(i, \sigma, t)$, we introduce a partial value function $V^P(i, \sigma, \eta, t)$, which is defined as the expected future revenue during a period t for a vehicle ending in location i given it wins a job at time η during its time-to-go σ .

We define $p_{ikl}(\sigma - \eta)$ as the conditional probability that a vehicle ending in location i will have a trip from k to l as next job, given that it wins a job at time η from now on. Here $\sigma - \eta$ is the time between winning a job and the earliest time it can actually start this job. Note that the probability of winning a specific job may depend on this time because jobs have time-window restrictions and these time-windows may differ per route.

We define $r_{ikl}(\sigma - \eta)$ as the expected reward of a job from k to l that is won at time η from now on. Now the next job from k to l ends at time $\sigma + \tau_{ikl}$ from now on. If $\tau_{ikl} \leq t$ (the job is handled within the time horizon $T = \sigma + t$), then we include the full profit in the value function. Otherwise we include a fraction $\alpha_{ikl}(t) = t/\tau_{ikl}$ corresponding to the percentage of the job that is completed within the time horizon. The next end-value is the value at the new schedule destination l at time $\sigma + \tau_{ikl}$ from now on. So the time-to-go from η on is $\sigma + \tau_{ikl} - \eta$ and the remaining time horizon is $\max\{t - \tau_{ikl}, 0\}$. By summation over all possible routes kl we get the following partial value function:

$$\begin{aligned}
 V^P(i, \sigma, \eta, t) = & \sum_{k,l \in \mathcal{N}} p_{ikl}(\sigma - \eta) \left[\alpha_{ikl}(t) r_{ikl}(\sigma - \eta) \right. \\
 & \left. + V^e(l, \sigma + \tau_{ikl} - \eta, t - \tau_{ikl}) \right] \tag{5}
 \end{aligned}$$

Here we use the boundary constraints $V^e(i, \sigma, t) = 0$ and $V^P(i, \sigma, \eta, t) = 0$ for $t \leq 0$.

5.1.3 End-value functions

We consider the following three cases for winning a new job: (a) we win a job during the time-to-go σ , (b) otherwise we end up at node i and decide to move pro-actively to node δ and we win a job during this time $\tau_{i\delta}^e$, (c) otherwise we end up at node δ and we wait until we win the next job over there. An illustration of the three cases can be found in Fig. 2. In the remainder of this section we explain how the partial value functions, as displayed in this figure, are derived for each case.

In case (a), the next job is won within the time-to-go σ , say at time $\theta + \eta$ ($0 \leq \eta \leq \sigma$). Therefore, the partial value function is simply given by $V^P(i, \sigma, \eta, t)$. By weighing over the time at which the next job is won, which we describe using a continuous probability density function $f_{i\sigma}(\eta)$ and corresponding distribution function $F_{i\sigma}(\eta)$, we find that the first part of the value function for the end-gap is given by $\int_0^\sigma f_{i\sigma}(\eta) V^P(i, \sigma, \eta, t) d\eta$. Note that $f_{i\sigma}(\eta)$ is an exponential density because we assumed Poisson arrivals, see Sect. 3.

In case (b) and (c), we do not win a job during the time-to-go σ . This happens with probability $1 - F_{i\sigma}(\sigma)$. Then the vehicle is at location i at time $\theta + \sigma$, and we update

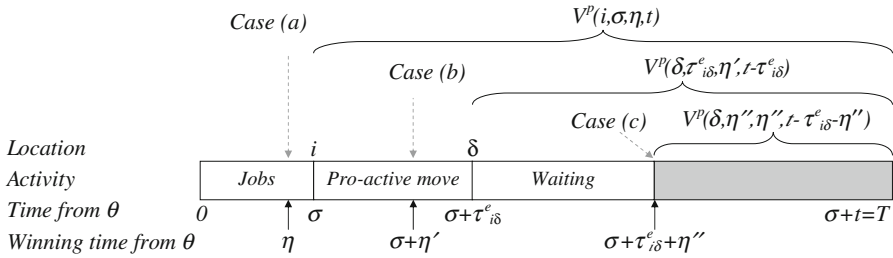


Fig. 2 Three cases for the partial value function depending on the winning time

the current time θ by making it the arrival time at node i . Now we have to find the best option for the pro-active move to location δ which takes a time $\tau_{i\delta}^e$ and which costs $c^r(\tau_{i\delta}^e)$ (if $\delta = i$ we wait at node i without costs for a pro-active move). Therefore, we compute the expected revenues and costs if we move pro-actively to δ and we select the option with maximum revenues in our recursion:

- In case (b), we win the next job at time $\theta + \eta'$ before arrival at node δ ($0 \leq \eta' \leq \tau_{i\delta}^e$). The remaining time horizon directly after arrival at the end-node δ is given by $t - \tau_{i\delta}^e$. Therefore, we find that the partial value function is given by $V^P(\delta, \tau_{i\delta}^e, \eta', t - \tau_{i\delta}^e)$.
- In case (c) we update the current time θ by making it the arrival time at node δ to which we have moved pro-actively. We win the next job at time $\theta + \eta''$. The new time-to-go is therefore also η'' and the remaining time horizon after winning this new job is $t - \tau_{i\delta}^e - \eta''$. The partial value function for this case is given by $V^P(\delta, \eta'', \eta'', t - \tau_{i\delta}^e - \eta'')$.

By combining the value functions for the three cases, we find the following relation for the end-values:

$$\begin{aligned}
 V^e(i, \sigma, t) &= \int_0^\sigma f_{i\sigma}(\eta) V^P(i, \sigma, \eta, t) d\eta \\
 &+ (1 - F_{i\sigma}(\sigma)) \max_\delta \left\{ -c^r(\tau_{i\delta}^e) + \int_0^{\tau_{i\delta}^e} f_{\delta\tau_{i\delta}^e}(\eta') V^P(\delta, \tau_{i\delta}^e, \eta', t - \tau_{i\delta}^e) d\eta' \right. \\
 &\left. + \left(1 - F_{\delta, \tau_{i\delta}^e}(\tau_{i\delta}^e)\right) \int_0^\infty f_{\delta 0}(\eta'') V^P(\delta, \eta'', \eta'', t - \tau_{i\delta}^e - \eta'') d\eta'' \right\} \quad (6)
 \end{aligned}$$

Again we use the boundary constraints $V^e(i, \sigma, t) = 0$ and $V^P(i, \sigma, \eta, t) = 0$ for $t \leq 0$.

From (6) we see that the end-value functions $V^e(i, \sigma, t)$ depend on the partial value functions with a time parameter less than or equal to t . From (5) we see that the partial value functions depend on the end-value functions with a time parameter strictly smaller than t . Now, if we discretize time, we can use (5) and (6) alternately to find

all end-value functions. To be precise, starting with the before mentioned boundary constraints, we iterate on the remaining time horizon t . For each t , we calculate the end-values $V^e(i, \sigma, t)$ for all i and σ , using the end-values found in previous iterations. However, even with perfect knowledge of the states, solving this recursion is a complex and time-consuming process. Especially, because the state space can be very large and for each state we have to calculate the partial value functions a large number of times. Therefore we propose some approximations.

5.2 Value function approximations

We apply the following approximations: discretization of time and replacement of the time-to-go σ with its expectation. For clarity of exposition we decided to perform two additional approximations. First, we approximate the gap-values by using the same parameters for the gap-values as for the end-values. This way we reduce the number of parameters that we have to estimate. In Sect. 6.4 we relax this assumption. Second, we use a time-to-go of zero. This way we keep the algorithms compact (because we do not have to consider the case of winning a job during the time-to-go) and relaxation of this assumption is straightforward as we will illustrate in the next subsection. Besides the notational convenience, the latter two approximations also reduce the computation times as we will see later on in our simulation experiments.

5.2.1 End-value approximation

As a first approximation we discretize time, from the current time θ on, into intervals of length ε . For ease of notation and without loss of generality, we assume that the time dimension is chosen such that $\varepsilon = 1$ while $\tau_{ij}^e \geq 1$ for $i \neq j$. Then, we use a discrete probability density $q_{i\sigma}(\eta)$ instead of the continuous density $f_{i\sigma}(\eta)$, where $q_{i\sigma}(\eta)$ denotes the probability that the vehicle will receive a job in the time interval $(\theta + \eta, \theta + \eta + 1]$, given schedule destination i and time-to-go σ . Similarly, we use $Q_{i\sigma}(\eta)$ instead of $F_{i\sigma}(\eta)$.

In the next approximation we replace the time-to-go σ with $\bar{\sigma}$. Because the state is now independent of the time-to-go σ , we can reduce this state to $\{i, t\}$ and we are able to derive the approximate value functions recursively. We denote the approximate end-value by \tilde{V}^e . This new value function is given by:

$$\begin{aligned} \tilde{V}^e(i, t) = & \sum_{\eta=0}^{\bar{\sigma}} q_{i\bar{\sigma}}(\eta) \tilde{V}^P(i, \bar{\sigma}, \eta, t) \\ & + (1 - Q_{i\bar{\sigma}}(\bar{\sigma})) \max_{\delta \in \mathcal{N}} \left\{ -c^r(\tau_{i\delta}^e) + \sum_{\eta'=0}^{\tau_{i\delta}^e} q_{\delta\tau_{i\delta}^e}(\eta') \tilde{V}^P(\delta, \tau_{i\delta}^e, \eta', t - \tau_{i\delta}^e) \right. \\ & \left. + \left(1 - Q_{\delta\tau_{i\delta}^e}(\tau_{i\delta}^e)\right) \sum_{\eta''=0}^{\infty} q_{\delta 0}(\eta'') \tilde{V}^P(\delta, \eta'', \eta'', t - \tau_{i\delta}^e - \eta'') \right\} \end{aligned} \tag{7}$$

where \tilde{V}^p is the approximate partial value function. This function is exactly the same as in (5), with the only exception that the end-value $V^e(i, \sigma, t)$ is replaced by the approximate end-value $\tilde{V}^e(i, t)$ where the time-to-go σ is replaced with its expectation $\bar{\sigma}$.

The approximate end-values based on an average time-to-go $\bar{\sigma}$ can be obtained using a simple backward stochastic dynamic programming recursion. At each single (discrete) point in time, starting with a planning horizon $t = 1$ until $t = T$, we calculate $\tilde{V}^e(i, t)$, using (7), for all schedule destinations $i \in \mathcal{N}$. However, as mentioned in the beginning of this section, we decided to use $\bar{\sigma} = 0$ in our recursions. Basically this means that a vehicle never expects to win a job during a certain time-to-go. So, at each point in time we calculate the probability that we win an auction during this unit time interval. If we do not win an auction, we make a proactive move (ending multiple time-units ahead) or we wait a single period at this node. The backward recursion for the end-values is given in Algorithm 2.

Algorithm 2 Calculating the approximate end-values

```

init:
  given a planning horizon  $T$ 
   $\tilde{V}^e(i, t) = 0$  for all  $i$  with  $t \leq 0$ 
for  $t = 1$  to  $T$  do
  for  $\forall i \in \mathcal{N}$  do
    
$$\tilde{V}^e(i, t) = Q_{i0}(1)\tilde{V}^p(i, t) + (1 - Q_{i0}(1)) \max_{\delta} \left\{ -c^r(\tau_{i\delta}^e) + \tilde{V}^e(\delta, t - \max(\tau_{i\delta}^e, 1)) \right\}$$

    
$$\tilde{V}^p(i, t) = \sum_{k,l \in N} P_{ikl}(0) \left[ \alpha_{ikl}(t)r_{ikl}(0) + \tilde{V}^e(l, t - \tau_{ikl}) \right]$$

  end;
end;
```

5.2.2 Gap-value approximation

We approximate the gap-values analogously to the end-values. First, we replace the gap-value function $V^g(i, j, \sigma, t)$ by $\tilde{V}^g(i, j, t)$ where time is discretized and the time-to-go σ is replaced by $\bar{\sigma}$. Recall that now t denotes the gap flexibility (see Sect. 4). As mentioned at the beginning of Sect. 5.2 we further apply two approximations for clarity of presentation. First we use a time-to-go of zero. Second, we use the same parameters (transition probabilities p_{ikl} , revenues r_{ikl} , and winning time distribution $Q_{i\sigma}$) for the gap-values as for the end-values. The algorithm for the approximate gap-values differs in four aspects from the approximate end-values.

- Because we use the same parameters for the gap-values as for the end-values, we may overestimate the winning probabilities, especially if we face a transition that involves a significant risk that we will violate the restrictions at the end of the gap. Then, it is possible that we make a non-profitable transition, i.e., if we had taken into account the restrictions at the end of the gap we would not have made a certain transition. To overcome this we multiply the transition probabilities with a decision variable δ_{kl}^g . This variable equals 1 if we accept the transition from k to

Algorithm 3 Calculating the approximate gap-values

init:

 given a schedule destination d

$t_N = \infty$

for $n = N - 1$ down to 1 do

 given $j = o(\varphi_{n+1})$ (end-node of the n th gap) and t_n (flexibility of the n th gap)

$\tilde{V}^g(i, j, s) = -\infty \forall i \neq j$ with $s < \tau_{ij}^e$ and $\tilde{V}^g(j, j, s) = 0 \quad \forall s$

 for $\forall i \in \mathcal{N}/j$ do

 for $s = \tau_{ij}^e$ to t_n do (only if $\tau_{ij}^e \leq t_n$)

$$\tilde{V}^g(i, j, s) = \max_{\delta_{kl}^a | k, l \in \mathcal{N}} \left\{ Q_{i0}(1) \tilde{V}^p(i, j, s) + (1 - Q_{i0}(1) + u(i)) \times \right. \\ \left. \max_{\delta \in \mathcal{N}} \left(\begin{array}{l} -c^r(\tau_{i\delta}^e) + \tilde{V}^g(\delta, j, s - \max(\tau_{i\delta}^e, 1)) \\ -\Delta \tilde{V}^e(d, \max(\tau_{i\delta}^e, 1) + \tau_{\delta j}^e - \tau_{ij}^e) \\ -\Delta \tilde{V}^g(n, \max(\tau_{i\delta}^e, 1) + \tau_{\delta j}^e - \tau_{ij}^e) \end{array} \right) \right\}$$

$$\tilde{V}^p(i, j, s) = \sum_{k, l \in \mathcal{N}} \delta_{kl}^a P_{ikl}(0) \left\{ r_{ikl}(0) - \Delta \tilde{V}^e(d, \tau_{ikl} + \tau_{ij}^e - \tau_{ij}^e) \right. \\ \left. - \Delta \tilde{V}^g(n, \tau_{ikl} + \tau_{ij}^e - \tau_{ij}^e) + \tilde{V}^g(l, j, s - \tau_{ikl}) \right\}$$

$$u(i) = Q_{i0}(1) (1 - \sum_{k, l \in \mathcal{N}} \delta_{kl}^a P_{ikl}(0))$$

 end;

end;

end;

l and otherwise it is zero. The logic behind this decision variable is that a vehicle always has the option to wait a single time unit if this seems to be more profitable than making the transition.

- Because the state of the approximate gap-value is given by (i, j, t) , we also have to pass the end-node j to the next iteration.
- We use another boundary condition $\tilde{V}^g(i, j, t) = -\infty$ for $i \neq j$ and $t < \tau_{ij}^e$, to force vehicles not to use more flexibility than there is available. Note that the fraction $\alpha_{ikl}(t)$ is not needed for the approximate gap-values, because vehicles are bounded by the gap flexibility.
- When a new job is inserted in a gap, it reduces the value of succeeding gaps and of the end-value. For the dynamic programming recursion this will mean that we have to subtract these values in every iteration and that the gap-values are defined recursively. Regarding the latter we have to calculate the value of the end-gap first, then the value of the gap before the last job, then working backwards until the gap after the first job. We denote the decrease in end-value due to a delay t of the scheduled delivery time of the schedule destination d by $\Delta \tilde{V}^e(d, t)$. We use $\Delta \tilde{V}^g(n, t)$ to denote the sum of the decreases in gap-values for all gaps after the n th gap, given the scheduled pickup time of the job $n + 1$ is postponed a time t .

The approximate gap-values are calculated using Algorithm 3. Here $u(i)$ is the probability that we did not accept a transition for a job won after node i . To approximately calculate the decrease $\Delta \tilde{V}^e(d, t)$ in end-value, we do not have to know the time-to-go σ until arrival at the schedule destination d . One can argue that the expected profit on

day t equals the expected profit on day $t + 1$, for t large enough (see Sect. 7). Therefore we calculate the change in end-value as follows:

$$\Delta \tilde{V}^e(d, t) = \tilde{V}^e(d, T) - \tilde{V}^e(d, T - t) \tag{8}$$

To calculate the decrease $\Delta \tilde{V}^g(n, t)$ in gap-values, we have to recalculate the gap flexibilities t_m for all gaps $m \geq n + 1$. We denote the updated flexibilities by t'_m . We calculate these flexibilities using Algorithm 1 with the only exception that we increase all times ω_m and ρ_m for $m \geq n + 1$ with time t . We calculate the change in gap-values as follows:

$$\Delta \tilde{V}^g(n, t) = \sum_{m=n+1}^{N-1} \left(\tilde{V}^g(d(\varphi_m), o(\varphi_{m+1}), t_m) - \tilde{V}^g(d(\varphi_m), o(\varphi_{m+1}), t'_m) \right) \tag{9}$$

A major disadvantage of the dependencies between gaps is that we cannot simply calculate the value of all possible gaps in advance. Therefore we also propose a variant in which we ignore the dependencies between gaps. Here we treat each gap in the schedule as if it were the last gap. So, each new move that is inserted in this gap only reduces the value of the end-gap. Because we can calculate the end-values in advance for all possible schedule destinations, we can also calculate these gap-values for all possible gaps in advance. As we will show in Sect. 7, this reduces the computation time and still yields reasonable results.

5.3 Using the value functions

As mentioned in Sect. 4, vehicles face three types of decisions: bid pricing, scheduling, and waiting decisions. These decisions are supported by the value functions as introduced in the previous sections. Note that all decisions have to be made in real-time whereas in principle the value functions can be calculated offline (see Sect. 7).

After substitution of the approximate value functions in (4), we are able to calculate the optimal bid price using (2) and determine the optimal insertion position for the new job using (1). The waiting decisions are supported by the parts of the value functions for which we assume we have to wait. For example in case of the end-gap we use (7), where the best waiting decision $\delta^*(i)$ at node i is given by:

$$\delta^*(i) = \arg \max_{\delta \in \mathcal{N}} \left\{ \begin{array}{l} -c^r(\tau_{i\delta}^e) + \sum_{\eta'=0}^{\tau_{i\delta}^e} q_{\delta\tau_{i\delta}^e}(\eta') \tilde{V}^P(\delta, \tau_{i\delta}^e, \eta', t - \tau_{i\delta}^e) + \\ \left(1 - Q_{\delta\tau_{i\delta}^e}(\tau_{i\delta}^e) \right) \sum_{\eta''=0}^{\infty} q_{\delta 0}(\eta'') \tilde{V}^P(\delta, \eta'', \eta'', t - \tau_{i\delta}^e - \eta'') \end{array} \right\} \tag{10}$$

The waiting decision in a gap is derived in a similar manner. However, we then have three possibilities: (1) immediately move to the origin of the next job and start this job, (2) immediately move to another node, (3) wait at most a certain maximum amount of time at the current location. An optimal waiting strategy can be found using the approximate gap-values which are calculated with Algorithm 3. In this recursion we established the best waiting decision for each node i and remaining gap flexibility s .

Searching these values will provide the maximum amount of time a vehicle should wait before starting on the next job currently in its schedule, given that it does not win another job during this time.

6 Parameter estimation

Because travel times are deterministic, we assume that all vehicles are aware of the travel times τ_{ij}^e and τ_{ij}^f for all routes $i, j \in \mathcal{N}$. This leaves us with the following parameters that we need to estimate:

- The average time-to-go $\bar{\sigma}$.
- The conditional probability $p_{ikl}(\sigma - \eta)$ that a vehicle ending in location i will have a trip from k to l as next job, given that it wins a job at time $\theta + \eta < \theta + \sigma$.
- The expected reward $r_{ikl}(\sigma - \eta)$ of a job from k to l that is won at time $\theta + \eta < \theta + \sigma$ with start-node i .
- The probability $q_{i\sigma}(\eta)$ that we win a new job in the time interval $(\theta + \eta, \theta + \eta + 1]$ if the schedule ends at node i .
- The distribution function $Q_{i\sigma}(\sigma)$ that we win a new job during the time-to-go σ if the schedule ends at node i .

We estimate these parameters and functions based on historic data. A possibility is to store all historic waiting times, revenues, travel times and transition percentages for all possible states. Even when these parameters do not depend on the time-to-go σ , we must store a lot of information. Therefore we propose to estimate these parameters based on auction data for certain routes and the job arrival intensity for these routes. The auction data can be used to estimate (1) the sample mean \bar{x}_{ij} and sample variance s_{ij}^2 of all observations of the winning price for all routes $i, j \in \mathcal{N}$, (2) the average job arrival intensities λ_{ij} for all routes $i, j \in \mathcal{N}$, (3) the average time-window length z_{ij} for jobs on all routes $i, j \in \mathcal{N}$, and (4) the average time-to-go $\bar{\sigma}$.

In the next section we describe how the vehicles estimate the distribution of the lowest bid using the sample mean and variance of the winning prices. In Sect. 6.2 we describe how the vehicles calculate the transition probabilities $p_{ikl}(\sigma - \eta)$ and expected revenues $r_{ikl}(\sigma - \eta)$. Calculation of the winning probabilities $q_{i\sigma}(\eta)$ and distribution $Q_{i\sigma}(\sigma)$ of winning moments is given in Sect. 6.3. In Sect. 6.4 we present the necessary modifications to deal with intrinsic parameters for the gap-values.

6.1 Distribution of lowest bids

The estimation of the distribution parameters of the lowest bids for jobs on this route, depends on the structure of the auction. Here, we consider the second-price auction where only the winning price, i.e., the one but lowest bid, is published. To estimate the distribution of the lowest bid, vehicles can use information about all winning prices together with their own bid history. Because your own bids provide little information about the bids of your competitors we only use information about the winning prices. This causes a problem, because we need the probability distribution of the *lowest* price whereas we only have observations of the *one but lowest* price.

To deal with this we use the theory of the so-called extreme value distributions (EVD), (see [Reiss and Thomas 1997](#)). The EVD are a class of probability distributions for the order statistics of a large set of random observations from the same (arbitrary) distribution (m th order statistic is the m th smallest value). These limiting distributions have the same set of distribution parameters. This enables us to estimate the parameters of the limiting distribution for the one but lowest bid, and use these parameters for the limiting distribution of the minimum bid.

Suppose that the bids b_c for a single job from competitor c ($c = 1 \dots C$) are independent and identically distributed with a cumulative distribution function $H(x)$. We denote the probability distribution functions of the corresponding order statistics by $H_m(x)$. Except for special cases, it is not possible to express the distributions of these order statistics as closed form expressions. However, it is shown in [Gumbel \(1958\)](#) that for any well-behaved initial distribution (i.e., $H(x)$ is continuous and has an inverse), limiting distributions for $n \rightarrow \infty$ can be derived, even if the distributions $H(x)$ are not exactly known or the iid condition of the bids fails ([Reiss and Thomas 1997](#)).

Here we use the so-called Gumbel distribution $G_m(x)$ for the m th order statistic (see [Reiss and Thomas 1997](#)), which only requests that the tail of the distribution of $H(x)$ declines exponentially. We calculate the parameters of $G_2(x)$ using the sample mean and sample variance of all historical winning prices (the second-lowest bids) for jobs on route k, l . Next we use these parameters in the distribution $G_1(x)$ of the lowest bid. In the remainder we will indicate this distribution by $H_{kl}^{\min}(x)$, which in fact describes the probability that a vehicle will lose an auction given its bid price x .

6.2 Estimating revenues and transition probabilities

To estimate the revenues and transition probabilities, we use so-called winning intensities. The winning intensities provide for all routes the intensity at which we expect to win jobs given a certain state. We define the winning intensity $\xi_{ikl}(\sigma)$ as the mean number of winning jobs per time unit from k to l after arrival at node i with time-to-go σ . The winning intensities are given by:

$$\xi_{ikl}(\sigma) = \lambda_{kl} \cdot p_{ikl}^{\text{win}}(\sigma) \tag{11}$$

where $p_{ikl}^{\text{win}}(\sigma)$ is the probability of winning a job from k to l with start-node i and time-to-go σ , which is given by:

$$p_{ikl}^{\text{win}}(\sigma) = 1 - H_{kl}^{\min}(b_{ikl}(\sigma)) \tag{12}$$

where $b_{ikl}(\sigma)$ is the expected bid price for a job on route kl , given schedule destination i and time-to-go σ . Using (2) and (4), and given that new jobs can only be added to the end of the schedule, we see that this bid price is given by the direct costs of this job and a decrease in end-value. We denote the decrease in end-value due to the insertion of a new job from k to l after arrival at node i with time-to go σ , by so-called opportunity

costs $OC_{ikl}(\sigma)$. This enables us to rewrite the bid price (2) as follows:

$$b_{ikl}(\sigma) = c_{kl}(\sigma) + OC_{ikl} \tag{13}$$

where—as in (3)—the direct costs are given by $c_{kl}(\sigma) = c^r(\tau_{kl}^f) + c^p((\sigma + \tau_{ik}^e - z_{kl})^+)$. The opportunity costs however are not known yet. On the contrary, because we are about to calculate the end-values, this whole approach is focused on finding them. Therefore we use an estimate based on value functions that we have calculated earlier for this state. In case of end-values, we approximate the opportunity costs by:

$$OC_{ikl} = \tilde{V}^e(i, T) + c^r(\tau_{ik}^e) - \tilde{V}^e(l, T - \tau_{ikl}) \tag{14}$$

An alternative is to perform multiple iterations of the dynamic programming recursions. At the beginning of each iteration we calculate all parameters, using the results of the previous iteration to calculate the opportunity costs. Again we can start the first iteration using the value functions calculated in the previous auction round or period. We can stop the iteration when the difference in value-function is sufficiently small.

In the end-value function (6) we integrate the expected revenues over all possible winning moments η . Given that we win a job during time-to-go σ at time η , the remaining time-to-go is $\gamma = \sigma - \eta$. The expected revenue as a function of the winning time-to-go γ is given by the difference between the expected lowest bid of the competitors (given that the own bid is lower) and the costs that are made:

$$r_{ikl}(\gamma) = \frac{1}{1 - H_{ikl}^{\min}(b_{ikl}(\gamma))} \int_{x=b_{ikl}(\gamma)}^{\infty} x dH_{ikl}^{\min}(x) - c_{kl}(\gamma) - c^r(\tau_{ik}^e) \tag{15}$$

In our simulation experiments (see Sect. 7) we calculate this function by numerical integration. The conditional probability $p_{ikl}(\gamma)$ that the winning job has origin k and destination l given location i is given by:

$$p_{ikl}(\gamma) = \frac{\xi_{ikl}(\gamma)}{\sum_{k'l'} \xi_{ik'l'}(\gamma)} \tag{16}$$

Note that when time-windows, penalty costs, or travel costs differ per route, then also the transition probabilities will be time dependent. So, at different points in time, different jobs will have the highest winning probability.

6.3 Distribution of winning moments

We described the winning moments η by a distribution function $Q_{i\sigma}(\eta)$. For $\eta \leq \sigma$, these winning moments follow a so-called non-homogeneous Poisson process (NHPP). The distribution function of the random amount of time η until the first time we win

an auction is given by:

$$Q_{i\sigma}(\eta) = 1 - e^{-\int_0^\eta \sum_{kl} \xi_{ikl}(\sigma-u)du}, \quad 0 \leq \eta \leq \sigma, \sigma > 0 \tag{17}$$

Note that when $\sigma < z_{kl} - \tau_{ik}^e$ for all k , then the penalties in the expected bid $b_{ikl}(\sigma)$ of (13) are zero. As a consequence, the winning moments follow a homogeneous Poisson process $Q_{i\sigma}(\eta) = 1 - e^{-\eta \sum_{kl} \xi_{ikl}(\sigma)}$. This situation generally occurs in our simulation experiments.

6.4 Special parameters for the gap-values

To deal with intrinsic parameters for the gap-values, we have to (1) make all functions that depend on the time-to-go σ also dependent on the end-node j and the gap flexibility t and (2) we have to use another opportunity costs function (14). The basic idea regarding the latter is that we explicitly incorporate the estimation of opportunity costs in the dynamic programming recursion itself. More precisely, we estimate all required parameters at each stage s of the dynamic programming recursions, where s is the remaining gap flexibility, by using the gap-values that are calculated in earlier stages $t < s$. For a given s we can estimate the opportunity costs as follows:

$$OC_{ikl}(j, s) = \tilde{V}^g(i, j, s - 1) + c^r(\tau_{ik}^e) - \tilde{V}^g(l, j, s - 1 - \tau_{ikl}) \tag{18}$$

These opportunity costs can be calculated using the gap-value functions $\tilde{V}^g(i, j, t)$ for $t < s$, which were already calculated in earlier steps of the dynamic programming recursion.

7 Simulation

In this section we discuss the results of a concise simulation study. For a more in depth analysis, including an illustration of the shape of the value functions and a sensitivity analysis with respect to a whole range of experimental factors, we refer to Mes (2008).

The goal of this study is to provide some insight into the performance of opportunity based bid pricing and scheduling and the effect of various approximations. We compare the opportunity based bid pricing and scheduling strategy with a naive strategy where only the direct costs of a job insertion are taken into account. We perform this comparison using three simplified market structures. First, we consider a *virtual market* where a single vehicle uses opportunity based bid pricing and scheduling. The lowest bid of other vehicles—indicated by external market—is generated from a given distribution function and the single vehicle has perfect knowledge of this distribution. We use this market type to study our approach under ideal circumstances. Second, we consider an *open market* which is a more realistic situation where we again consider a single vehicle using opportunity based bid pricing and scheduling, but where the other vehicles use the naive bid pricing and scheduling strategy. Third, we consider a *closed market* where all vehicles are using the same

pricing and scheduling strategy. The closed market structure can be seen as an internal application of our approach. Examples of closed environments include (1) shippers that control their own fleet of vehicles, (2) collaborative carriers, and (3) factories that allocate internal transportation tasks to automatic guided vehicles (see, e.g., Mes et al. 2008).

7.1 Settings

We consider unbalanced transportation networks, in the sense that some nodes are more popular than others. We consider two network settings, a 3 node network and a 9 node network. In the 3 node network, the nodes are spanning up an equilateral triangle. The distances between the nodes are 50 km. We label the nodes A, B, and C. The probability of being an origin node is 0.1, 0.3, 0.6 for node A, B, C, respectively. In the 9 node network, the nodes are the grid points of a 3×3 square grid. Distances are Euclidian and such that the horizontal and vertical distances between adjacent nodes are 25 km. The probability of being an origin for nodes on row 1 is 5 times higher than row 2 and 25 times higher than row 3. Per row all probabilities are equal. In both network settings, the destination node is drawn randomly from all nodes other than the origin node.

We use ten vehicles, each having a travel speed of 50 km/h (both empty and loaded). The travel cost function is given by $c^r(t) = t$ and the penalty cost function by $c^p(t) = 10t$. The loading- and unloading times are 5 min each. For the dynamic programming recursions we discretize time into periods of 1 min and use a planning horizon of 12,000 min. Jobs have a time-window length of 10 h, and arrive according to a Poisson process. In the 3 node network, the mean interarrival time of jobs is 900 s. In the 9 node network we consider two settings for the mean interarrival time between jobs: quiet (1,200 s) and busy (800 s).

We evaluate the approximate end-values and gap-values in combination with the following relaxations: the use of the average time-to-go $\bar{\sigma}$ (see Sect. 5.2.1) and the use of custom parameters for the gap-values (see Sect. 6.4). In addition we consider the proposed gap-values defined recursively over all gaps in the schedule as well as the approximation where we treat each gap in the schedule as if it were the last gap (see Sect. 5.2.2). Hence, we consider the following policies:

VE	End-values based on $\sigma = 0$
VEA	End-values based on $\sigma = \bar{\sigma}$
VG	Gap-values based on $\sigma = 0$
VGC	Gap-values with their own parameters based on $\sigma = 0$
VGAC	Gap-values with their own parameters based on $\sigma = \bar{\sigma}$
VGS	Gap-values calculated recursively over all gaps in the schedule based on $\sigma = 0$
VGAS	Gaps-values calculated recursively over all gaps in the schedule based on $\sigma = 0$

For the virtual market we use a Gumbel G_1 distribution to generate the lowest bid of the external market. The parameters of this distribution are estimated based

on a learning phase of 500 days. In this learning phase we have ten vehicles using the naive pricing and scheduling strategy. At each auction round, we store the lowest bid of a fixed group of nine vehicles (which represent the external market). At the end of the learning phase we estimate the parameters of the Gumbel distribution from the mean and standard deviation of the lowest bid data. The single vehicle calculates the different value functions in advance at the end of the learning phase, except for the gap-values which are defined over all gaps in a schedule. In the open market, we have a single vehicle using opportunity based bid pricing and scheduling, and nine vehicles using the naive strategy. Although prices from the external market are possibly influenced by the behavior of the individual vehicle, we decided to use the same value functions as in the virtual market. As a consequence, we have to calculate the value functions only once for the virtual or open market. To determine the value functions, we perform 10 iterations of the proposed dynamic programming recursions (see Sect. 6.2). As shown in Mes (2008), this number is sufficient to converge to a stable level.

In the closed market, we have ten vehicles using opportunity based bid pricing and scheduling. Because now all players can change their bid pricing behavior, we cannot simply calculate all value functions in advance. Instead we calculate them periodically, each 50 days, using the observations of the last 50 days. Here we use only one iteration of the dynamic programming recursions, but we calculate the opportunity costs (14) based on the value functions calculated in the previous period.

As primary performance indicator we consider the relative profit (RP). The definition of RP differs per market structure. In the virtual market, the RP of a certain policy is given by the realized profit under this policy compared to the profit of using the naive policy. In the open market, the RP of a certain policy is given by the realized profit under this policy compared to the average profit of the nine vehicles that are using the naive policy. In the closed market, the RP of a certain policy provides the change in total net costs compared to the situation in which all vehicles are using the naive policy. Here the total net costs consist of costs for traveling empty and penalty costs. The loaded move costs are not included because they always have to be made and cannot be reduced. To explain the RPs we consider two secondary performance indicators: the percentage of the time vehicles are driving loaded (DL) and the service level (SL) being the percentage of jobs that are picked up before the latest pickup time.

We use a replication/deletion approach for our simulations, cf. (Law and Kelton 2000), where each experiment consists of a number of replications (each with different seeds) and a warm-up period. The length of each simulation run for the closed market setting is 710 days, including a warm-up period of 210 days. The length of each simulation run for the virtual and open market consists of 510 days including 10 days as a warm-up period. Here the learning phase is not included in the run length, but is done separately for multiple experiments in advance. To determine the number of replications we consider all three performance indicators of all experiments. The maximum number of replications needed with a confidence level of 95% and a relative error of 5% is 5. To facilitate comparison we use five replications for all experiments. Only the learning period consists of one replication.

Table 1 Simulation results for the virtual market

Policy	RP	DL	SL
Naive	0	79.5	95.1
VE	56.9	89.4	99.0
VE + VG	70.0	91.5	99.0
VE + VGC	71.3	91.8	99.0
VE + VGS	72.9	91.9	99.0
VEA	57.6	89.4	99.8
VEA + VGAS	74.1	92.0	99.3

7.2 Results

Here we present the results of our simulation study. First, we consider the combinations of the 3 node network with the virtual and open market structure and all policies. Next, we consider the combinations of the 9 node network with all market structures and a subset of policies. After that we present the computation times of the various policies.

7.2.1 3 Node network

First we consider the virtual market. The results can be found in Table 1. We see that the opportunity valuation policies always lead to an increase in performance for all three performance indicators. The more precise policies (adding gap-values, an average time-to-go, special gap-parameters, gap dependencies) always result in higher RPs. In most cases, this is the result of an increase in both secondary performance indicators (DL and SL), with the only exception that with the policy VEA the percentage of DL is relatively low and the SL is the highest. We further see that the relatively simple policies, such as VE and VE + VG, perform remarkably well. The advantage of these approximate policies is that they work much faster, see Sect. 7.2.3.

To explain the results, let us consider the route between C and B. Theoretically, 15% of the jobs from a vehicle go from B to C and 30% from C to B. Under the naive policy we see that for the individual vehicle, 15.4% of the jobs go from B to C and 24.5% go from C to B. Under the policy VEA + VGAS, these numbers are 37.3 and 39.4%, respectively. Obviously, this results in fewer empty miles for the individual vehicle and hence in more profits.

Next we consider the open market. Here we compare the performance of the individual vehicle with the other nine vehicles. The results can be found in Table 2. For the performance indicators DL and SL, we first show the results of the individual vehicle, and after the slash we show the average result of the other nine vehicles. We further use the RP to denote the profit of the individual vehicle compared to the average profit of the other nine vehicles.

An interesting result is that the RP is much higher than in the virtual market. This can be explained because in the open market not only the individual vehicle is better off, but the others are worse off because the most profitable jobs are taken out by the

Table 2 Simulation results for the open market

Policy	RP	DL	SL
Naive	-3.2	69.7/70.0	99.7/99.8
VE	560.8	86.4/67.9	99.1/99.9
VE + VG	726.2	85.6/67.5	98.9/99.8
VE + VGC	730.3	86.4/67.3	98.8/99.8
VE + VGS	762.7	86.8/67.3	98.9/99.9
VEA	575.7	88.5/67.8	100.0/99.8
VEA + VGAS	769.7	90.6/67.6	98.2/99.8

individual vehicle. As an example consider the jobs toward the most popular node C. With the policy VE, 68.5% of the jobs from the individual vehicle are between B and C, whereas the other nine vehicles have on average 41.8% of their jobs on the route BC, which is less than the average total of 45% of jobs on this route. Again, this results in fewer empty miles for the individual vehicle and hence in more profits.

7.2.2 9 Node network

Next we consider the 9 node network using the virtual market, the open market, and also the closed market setting. We consider the two opportunity valuation policies that are based on the average time-to-go. Because the policies require considerably more computation time in the 9 node network, we decided to ignore the dependencies between gaps and therefore use the gap-values VGAC. The results can be found in Table 3.

The results for the virtual and open market are more or less the same as in the 3 node network. The opportunity valuation policies always lead to an increase in performance, and in some cases, the profit of the individual vehicle is more than the total profits of the nine other vehicles. We further see that the percentage of DL is most of the time higher in the busy networks because there are more opportunities to avoid empty moves. The SLs are most of the time lower in the busy networks.

In the closed network settings, the opportunity valuation policies also perform very well. The total costs for driving empty and lateness are reduced by 11.1% by using the policy VEA + VGAC in the quiet network setting. In the virtual and open networks, the benefits of opportunity valuation are merely caused by selecting the most profitable jobs. This no longer holds in the closed environments because all vehicles use the same policy, and all jobs have to be transported. So, the only explanation is that vehicles are scheduling the jobs more efficiently. To illustrate this, let us consider the policy VE + VGC. Here jobs from the first row to the last row, are scheduled 5.5% later in time, and jobs from the last row to the first row 19.7% earlier in time. So jobs with a higher probability of an empty move afterwards are scheduled later in time, increasing the probability that the empty move is replaced by a loaded move. We also see that the average schedule length (difference between the expected delivery time of the last

Table 3 Simulation results for the 9 node network

Market	Policy	Network	RP	DL	SL
Virtual	Naive	Quiet	0	63.8	94.3
		Busy	0	64.0	92.9
	VEA	Quiet	79.5	75.5	100.0
		Busy	96.2	78.3	99.9
	VEA + VGAC	Quiet	98.6	75.7	100.0
		Busy	103.4	78.5	100.0
Open	VEA	Quiet	395.0	69.6/61.5	99.7/98.8
		Busy	254.7	71.1/62.2	99.5/98.1
	VEA + VGAC	Quiet	822.1	71.9/60.9	99.6/99.0
		Busy	498.0	73.8/61.8	99.5/98.4
Closed	Naive	Quiet	0	62.6	98.4
		Busy	0	63.0	97.7
	VEA	Quiet	10.1	64.8	99.7
		Busy	8.9	64.8	99.5
	VEA + VGAC	Quiet	11.1	65.0	98.7
		Busy	10.1	65.0	98.5

Table 4 Computation times in seconds

Policy	3 Nodes	9 Nodes
VE	0.35	15.19
VEA	1.12	129.21
VG	0.05	8.79
VGA	0.16	75.52
VGC	0.15	23.46
VGAC	0.46	200.47

job and the current time) is decreased by 9.3%. This in turn provides more flexibility to schedule jobs on time.

7.2.3 Computation times

The computation time required for insertion scheduling and marginal costs calculation for bid pricing is very small. The major concern here is the computation time required to calculate the approximate value functions. For our experiments we used the simulation software eM-Plant 7.5 and an Intel Pentium 4 processor at 3.4 GHz. The end-values are calculated for the whole planning horizon of 12,000 min. The gap-values are calculated in advance for all possible end-nodes and a maximum gap flexibility of 600 min. The results can be found in Table 4.

Clearly, the computation time increases drastically when going from 3 nodes to 9 nodes. However, this increase is much smaller in the approximations where we are using the zero time-to-go. Besides, we can calculate these functions off-line (e.g., periodically) such that, given the insertion scheduling strategy, we are able to respond within milliseconds (except for the policy VGAS which has to be calculated upon each announcement because it depends on the whole schedule).

8 Conclusions

In this paper, we presented an opportunity based bid pricing and scheduling concept. This method makes it possible to determine the value of a schedule, more specifically the value of gaps in a schedule and the value of a schedule destination. This enables us to calculate the opportunity costs which are defined as the loss in expected future revenues due to a new job insertion. By including this value in our bid price we not only cover the direct costs of a job insertion, but also its future implications. Moreover, by using this value in our scheduling decisions we prevent less profitable moves and increase our opportunities by better repositioning of vehicles.

From our simulation experiments, we conclude that an individual player using opportunity based bid pricing will perform significantly better than other players who use a naive pricing strategy (4–9 times higher profits). But also in closed environments (internal use of our approach), opportunity based bid-pricing and scheduling can reduce the system-wide logistical costs (an average reduction of 10% in costs for driving empty and tardiness).

We explain these results from the following behavior. First, vehicles tend to schedule an unattractive job later in time, increasing the probability of combining this job with another job. Second, the average schedule length is reduced such that there is more flexibility of scheduling jobs on time. Third, in case of open markets, the individual vehicle tends to select the most profitable jobs, leaving the less profitable jobs for the external market. We further showed that the approximations (a time-to-go of zero and gap-values using the same parameters as the end-values) perform remarkably well in most cases. The advantage of using the approximations is that they require less computation time and are better scalable to larger systems.

A more in depth numerical analysis can be found in [Mes \(2008\)](#). From these experiments we conclude that in the open market, the profits of the individual vehicle are 2–10 times higher than the average profit of the other players, depending on the job arrival intensity and time-window length. If fewer jobs are being auctioned, the external market suffers from the individual vehicle selecting the most profitable jobs, and hence the relative profit of the individual vehicle increases. Longer time-windows result in higher relative profits for the individual vehicle because it provides more scheduling flexibility and increases the need to value the opportunities of a schedule. For the closed market, we observe a reduction in costs for empty moves and tardiness of 10–30%. Here, high reductions are achieved with a high job arrival intensity, long time-windows, and highly unbalanced networks. For the background and numerical details we refer to [Mes \(2008\)](#).

Further research consists of an extensive simulation study of the effects of different market settings on the performance of opportunity based bid pricing and scheduling techniques for agent-based transportation planning. Furthermore, some interesting model extensions will be examined, such as stochastic travel times and learning with feedback. Also, faster methods will be designed for real-time decision making in larger problem instances. Regarding the latter, we think of functional approximations of the value functions.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

- Bertsimas D, van Ryzin G (1991) A stochastic and dynamic vehicle routing problem in the euclidian plane. *Oper Res* 39(4):601–615
- Bertsimas D, van Ryzin G (1993) Stochastic and dynamic vehicle routing in the euclidean plane with multiple capacitated vehicles. *Oper Res* 41(1):60–76
- Brandt F, Weiß G (2002) Antisocial agents and Vickrey auctions, In: Tambe J, Meyer M (eds) *Intelligent agents VIII*, vol 2333 of lecture notes in computer science. Springer, New York, pp 335–347
- Branke J, Middendorf M, Noeth G, Dessouky M (2005) Waiting strategies for dynamic vehicle routing. *Transp Sci* 39(3):298–312
- Carvalho T, Powell W (2000) A multiplier adjustment method for dynamic resource allocation problems. *Transp Sci* 34(2):150–164
- Figliozzi M (2004) Performance and analysis of spot truck-load procurement markets using sequential auctions. PhD thesis, School of Engineering, University of Maryland
- Figliozzi M, Mahmassani H, Jaillet P (2006) Quantifying opportunity costs in sequential transportation auctions for truckload acquisition. *Transp Res Rec* 1964:247–252
- Ghiani G, Guerriero F, Laporte G, Musmanno R (2003) Real time vehicle routing: solution concepts, algorithms and parallel computing strategies. *Eur J Oper Res* 151(1):1–11
- Godfrey G, Powell W (2002) An adaptive dynamic programming algorithm for dynamic fleet management, I: single period travel times. *Transp Sci* 36(1):21–39
- Gumbel E (1958) *Statistics of extremes*. Columbia University Press, New York
- Ichoua S, Gendreau M, Potvin J (2006) Exploiting knowledge about future demands for real-time vehicle dispatching. *Transp Sci* 40(2):211–225
- Larsen A, Madsen O, Solomon M (2004) The a priori dynamic traveling salesman problem with time windows. *Transp Sci* 38(4):459–472
- Law A, Kelton W (2000) *Simulation modeling and analysis*, vol 3. McGraw-Hill, Singapore
- Mes M (2008) *Sequential auctions for full truckload allocation*. PhD thesis, School of Management and Governance, University of Twente, The Netherlands
- Mes M, van der Heijden M, van Harten A (2007) Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems. *Eur J Oper Res* 181(1):59–75
- Mes M, van der Heijden M, van Hillegerberg J (2008) Design choices for agent-based control of AGVS in the dough making process. *Decis Support Syst* 44(4):983–999
- Mitrović-Minić S, Laporte G (2004) Waiting strategies for the dynamic pickup and delivery problem with time windows. *Transp Res B* 38(7):635–655
- Powell W, Sheffi Y, Nickerson K, Atherton S (1988) Maximizing profits for north american van lines truckload division a new framework for pricing and operations. *Interfaces* 18(1):21–41
- Reiss R, Thomas M (1997) *Statistical analysis of extreme values*. Birkhauser Verlag, Basel
- Sandholm T (2002) Algorithm for optimal winner determination in combinatorial auctions. *Artif Intell* 135(1-2):1–54
- Song J, Regan A (2002) Combinatorial auctions for transportation service procurement: the carrier perspective. *Transp Res Rec* 1833:40–46

- Swihart M, Papastavrou J (1999) A stochastic and dynamic model for the single-vehicle pick-up and delivery problem. *Eur J Oper Res* 114(3):447–464
- Thomas B (2007) Waiting strategies for anticipating service requests from known customer locations. *Transp Sci* 41(3):319–331
- Thomas B, White C (2004) Anticipatory route selection. *Transp Sci* 38(4):473–487
- Vickrey W (1961) Counterspeculation, auctions, and competitive sealed tenders. *J Finance* 16(1):8–37
- Wellman M, Walsh W, Wurman P, MacKie-Mason J (2001) Auction protocols for decentralized scheduling. *Games Econ Behav* 35(1):271–303
- Wooldridge M (1999) Intelligent agents. In: Weiss G (ed) *Multiagent systems*. The MIT Press, Cambridge, pp 27–77