Ann Oper Res
DOI 10.1007/s10479-014-1683-6

Distribution of waiting time for dynamic pickup and delivery problems

Stefan Vonolfen · Michael Affenzeller

Abstract Pickup and delivery problems have numerous applications in practice such as parcel delivery and passenger transportation. In the dynamic variant of the problem, not all information is available in advance but is revealed during the planning process. Thus, it is crucial to anticipate future events in order to generate high-quality solutions. Previous work has shown that the use of waiting strategies has the potential to save costs and maximize service quality. We adapt various waiting heuristics to the pickup and delivery problem with time windows. Previous research has shown, that specialized waiting heuristics utilizing anticipatory knowledge potentially outperform general heuristics. Direct policy search based on evolutionary computation and a simulation model is proposed as a methodology to automatically specialize waiting strategies to different problem characteristics. Based on the strengths of the previously introduced waiting strategies, we propose a novel waiting heuristic that can utilize historical request information based on an intensity measure which does not require an additional data preprocessing step. The performance of the waiting heuristics is evaluated on a single set of benchmark instances containing various instance classes that differ in terms of spatial and temporal properties. The diverse set of benchmark instances is used to analyze the influence of spatial and temporal instance properties as well as the degree of dynamism to the potential savings that can be achieved by anticipatory waiting and the incorporation of knowledge about future requests.

Keywords Dynamic pickup and delivery problem · Waiting strategies · Direct policy search · Simulation-based optimization

1 Introduction and motivation

Due to advances both in the field of telematics and in the field of operations research, dynamic models of vehicle routing problems (VRP) are getting much attention recently. More and

S. Vonolfen (✉) · M. Affenzeller
School of Informatics, Communication and Media, Heuristic and Evolutionary Algorithms Laboratory, University of Applied Sciences Upper Austria, Softwarepark 11, 4232 Hagenberg, Austria
e-mail: stefan.vonolfen@fh-hagenberg.at

Published online: 12 August 2014

more real-time information is available during the planning process and transport logistics companies are faced with increasingly competitive environments with global players. In many markets, customers are demanding a flexible and timely fulfillment of their requests. A challenging issue in dynamic vehicle routing is missing or uncertain knowledge about the future which can increase the problem complexity compared to static problems.

Previous work on dynamic VRPs has shown, that it is important to anticipate future events and consider them in the planning process, since not all information is known in advance and thus sub-optimal decisions are made due to incomplete knowledge. One way of considering future requests is the use of waiting strategies which have been already applied successfully to minimize distribution costs and increase the service quality.

The aim of this work is to revise, review and improve the methodology on distributing the waiting time for dynamic pickup and delivery problems (PDP). For that purpose, several waiting strategies from the literature are reviewed and adapted to the dynamic PDP with time windows. Following and concluding from a survey of the state-of-the-art, two methodological improvements are proposed. Firstly, a novel waiting heuristic is presented that builds on successful aspects of previously proposed waiting strategies. Secondly, a method for automatically parameterizing waiting strategies that are specialized to their problem environment is presented. On a set of standard benchmark instances, a computational study is performed analyzing the benefit of anticipatory waiting under different spatial and temporal conditions.

The main contribution of this article is threefold:

– The performance of various waiting strategies that have been proposed by different authors are compared on the same set of benchmark instances and their performance is analyzed with respect to diverse problem instance characteristics. Previously, the proposed approaches have not been compared directly. Furthermore it is analyzed under which circumstances it is beneficial to apply waiting strategies and to utilize historical data.
– Waiting strategies are automatically parametrized and adapted to problem instance characteristics by using a set of training instances which has often been done manually in the past. In this work an evolution strategy is used for this purpose. In a more general context, the potential of applying direct policy search and an algorithm portfolio in the field of vehicle routing is illustrated.
– Combining the strengths of previous approaches, a novel waiting strategy is proposed that can utilize historical data without an additional preprocessing step by means of an intensity measure. Its competitiveness on standard benchmark instances is shown.

The remainder of this paper is organized as following: In Sect. 2 we provide a literature survey and relate our findings to the existing work. In Sect. 3 the problem description is provided which is the basis for our study. In Sect. 4 we give an overview of existing waiting strategies, In Sect. 5 we propose a novel waiting strategy and present an approach for automatic parametrization in Sect. 6. In Sect. 7 we perform computational experiments with various waiting strategies on a set of standard benchmark instances. Finally, in Sect. 8 we summarize our findings and give an outlook on future research.

2 Literature overview

The VRP has been formulated by Dantzig and Ramser (1959) and since then many variants of have emerged and have been successfully applied in practice. For a taxonomic overview of different VRP variants the reader is referred to Eksioglu et al. (2009). A survey on recent advances and challenges in the field of vehicle routing is given by Golden et al. (2008).

Since the seminal work of Psaraftis (1988), many different variants of dynamic VRPs have been studied in the literature. Successful practical applications are numerous and include the distribution of heating oil, taxi cab services, on-demand transportation of elderly people or courier services. Surveys, reports and special issues about dynamic routing problems include the work of Ghiani et al. (2003), Gendreau and Potvin (2004), Zeimpekis et al. (2007) and Pillac et al. (2013).

Variants of pickup and delivery problems have wide practical applications. A recent survey on dynamic PDP is given by Berbeglia et al. (2010). There are several variants of dynamic PDP that differ in terms of problem characteristics and application areas.

A special variant is the dynamic stacker crane problem where the vehicles can only transport one item at a time. Applications include full truckload transportation problems (Tjokroamidjojo et al. 2006), the dispatching of automated guided vehicles (Mes et al. 2007) and dispatching of passenger transportation with vehicles that can only carry a single person (Gutenschwager et al. 2004).

In the general case PDP deal with vehicles that can serve multiple requests at once. This includes the transportation of objects such as parcels or letters. Different variants have been studied such as uncapacitated (Waisanen et al. 2008; Ghiani et al. 2003), capacitated (Sáez et al. 2008) and time window constrained (Mitrovic-Minic et al. 2004) cases. A special variant dealing with passenger transportation is the dynamic dial-a-ride problem (DARP) which also considers service quality for the passengers. A survey about static and dynamic variants of the DARP is given by Cordeau and Laporte (2007). The solution approaches include parallel metaheuristics (Attanasio et al. 2004), local search (Xiang et al. 2008) and repositories of solutions (Coslovich et al. 2006). A general survey about different PDP variants and algorithms is provided by Parragh et al. (2008).

As pointed out by Psaraftis (1988) and recently again by Pillac et al. (2013), a critical aspect of dynamic vehicle routing is incomplete and unknown information. The quality of a solution can only be evaluated a-posteriori after all information is known. Even if we would have an exact algorithm to optimize the situation at a given time, the appearance of new data might render the current solution suboptimal. It is thus crucial to anticipate future events and consider them in the planning process, since not all information is known in advance and thus sub-optimal decisions are made due to incomplete knowledge.

Various approaches have been proposed in the past to anticipate future requests in dynamic vehicle routing as outlined by Ichoua et al. (2007). These include double horizon, fruitful region and multiple scenario approaches as well as waiting strategies.

One way of anticipating future requests has been proposed by Mitrovic-Minic et al. (2004) where they introduce the concept of the double horizon. Basically they use different objectives for short-term requests and long-term requests. The short-term objective is to minimize the true objective (e.g. distance) while the long-term objective is to maximize slack times to better incorporate future requests. This approach relies on a modification of the evaluation function and does not consider any knowledge about demand distributions.

In contrast, approaches based on dynamic and stochastic models consider knowledge about demand distributions. According to Berbeglia et al. (2010), the literature is still very scarce in terms of dynamic and stochastic PDP, however there has been an increasing effort to exploit stochastic information in the planning process for dynamic vehicle routing problems. For instance, Van Hemert and La Poutré (2004) proposed anticipated moves towards fruitful regions based on probability distributions. If the occurrence probability is high in a certain region, the vehicle is relocated there if this does not impose any constraint violations.

A general framework for solving online stochastic combinatorial optimization problems with application to dynamic vehicle routing is presented by Hentenryck and Bent (2009).

By utilizing a sampling technique they produce plans that include both actual and forecast requests. Different sampled scenarios are solved independently and then combined together to a multiple scenario approach (MSA). A natural extension of the MSA is the incorporation of waiting and relocation of vehicles as shown by (Bent and Van Hentenryck 2007). In contrast to their approach we will focus on an a-posteriori distribution of waiting time along a planned route while they focused on inference from sampled customers.

In terms of waiting strategies, approaches based on dynamic and stochastic models as well as general strategies not considering anticipatory knowledge have been proposed. The various strategies have been evaluated on different variants of dynamic VRPs previously and have not been tested on a single set of benchmark instances so far. We have implemented the referenced waiting strategies and adapted them to the dynamic pickup and delivery problem with time windows (PDPTW) as described in Sect. 4 and compared their performance on a set of standard benchmark instances.

Mitrovic-Minic and Laporte (2004) have evaluated the benefit of waiting strategies on a PDPTW. They have coupled several waiting strategies with a tabu search heuristic and empirically shown the benefit of anticipatory waiting on benchmark instances that are based on real-life data. The objective was to optimize both the number of used vehicles and the expected detour. For that purpose they have proposed two heuristics that rely on dynamic partitioning of the route based on service zones and compared them to two simple strategies, namely the drive first and wait first. Similar to their work we perform our experiments on dynamic PDPTW and also aim at optimizing the fleet size and driven distance. However, they did not consider anticipatory knowledge about future demands.

Branke et al. (2005) were the first to formulate the waiting drivers problem and showed its NP-hardness. They have analytically proven that for the single vehicle case it is optimal not to wait. However, when the service region is not restrictive or there are multiple vehicles, waiting can be beneficial. For two vehicles they have analytically derived an optimal strategy under certain restrictions. Based on the analytical findings they have presented various heuristics to distribute the waiting time as well as an evolutionary algorithm approach. Their experiments have been carried out on a distance constrained VRP (DVRP) where a single additional customer arrives during the planning period. The objective is to maximize the probability to serve the additional customer. The empirical findings indicated that the heuristics derived from the optimal policy performed best. Compared to their work, we consider a more specialized VRP variant, namely the PDPTW with multiple arriving customers and aim at minimizing the distribution costs. We do not perform an analytical study but base our findings solely on computational experiments on a less restricted model. Also, no information about future customers was utilized in the strategies assuming a uniform distribution.

The first approach to utilize stochastic knowledge in a waiting strategy was proposed by Ichoua et al. (2006) assuming that future events arrive in a predictable way. They presented a threshold-based heuristic which is based on a partitioning of the distribution area into geographic zones. For each zone, a Poisson parameter is given that specifies the arrival rate of new customers. They evaluated the performance on a vehicle routing problem with time windows (VRPTW) and stochastic customers. The objective is to maximize the served customers as a primary objective and as a secondary objective to minimize the travel time and lateness penalty. They coupled the strategy with a parallel tabu search algorithm and show the advantages on two different sets of generated benchmark instances. For each set, they performed the parameter tuning manually to adapt the strategy to instance characteristics. Similar to their approach, we propose a threshold-based heuristic that can incorporate anticipatory knowledge. In contrast, our waiting heuristic is not based on probability distributions but on an intensity measure and thus does not require a partitioning of the service area and also requires no preprocessing step to utilize historical data. The parametrization is not done manually but using an evolution strategy on a training set of benchmark instances.

3 Problem description

In this article, we will focus on a special variant of vehicle routing problems - the dynamic PDPTW. Applications of the dynamic PDPTW are manifold and include full-truckload problems, less-than-truckload problems and passenger transportation. For instance, this problem arises at courier companies that operate same-day pickup and delivery of parcels and letters.

The formulation is based on the static PDPTW model defined by Savelsbergh (1995). A fleet of homogeneous capacitated vehicles has to serve a set transportation requests during a planning period (i.e. a day of operation). Each request is characterized by a pickup and a delivery location and the size of the load to be transported. For each location, a time window is given in which the service has to occur. A request has to be fulfilled by exactly one service of a single vehicle, this means that split deliveries are not allowed. In the dynamic formulation, not all pickup and delivery requests are known in advance but are revealed during the planning period.

Each service request s is revealed at a certain time r_s and contains the pickup location l_s^p and the delivery location l_s^d. Each location l can be serviced in the time window defined by the opening time o_l and closing time c_l.

The dynamic PDPTW consists of determining routes for each vehicle, i.e. a ordered sequence of all pickup and delivery locations. The routes have to satisfy certain constraints:

- each vehicle serves one route
- routes always start and end at the depot
- each request is served exactly once by a single vehicle
- a pickup is always made before its corresponding delivery
- all pickup and delivery time windows of the locations are satisfied
- the capacity restrictions of the vehicle are not exceeded

The objective is to minimize the required fleet size as well as the total driven distance. The costs are calculated as a weighted sum of the utilized vehicles and the driven distance. The costs for using a vehicle are set to $c_v = 3,000$, the costs for the traveled distance is set to $c_d = 1$. These values were set according to previous practical studies such as Vonolfen et al. (2013a). We assume that enough vehicles are available to serve all requests as also observed in case studies such as Mitrovic-Minic et al. (2001). The main objective is thus to minimize the fleet size, the secondary objective to minimize the driven distance over a given planning horizon.

As pointed out by Mitrovic-Minic and Laporte (2004) not only the routing but also the scheduling aspect is important when dealing with dynamic PDPTW. While routing determines the ordered sequence of visited locations, the scheduling determines arrival and departure times for each location. Waiting strategies deal with distributing the slack time along a given route. We assume that the wait time is distributed a-posteriori after the routes have been created. A vehicle can wait after each completed service, i.e. pickup or delivery operation, along the route.

4 Waiting strategies

In the following, we will describe various heuristics that deal with distributing the slack time over a route for dynamic PDPTW. Several waiting strategies have been proposed previously for different variants of vehicle routing problems and we have adapted them to the PDPTW. For details, the reader is referred to the referenced publications where the strategies have been originally defined.

We roughly categorize the strategies into general and specialized waiting strategies. General waiting strategies do not incorporate any knowledge about the problem structure such as spatial distribution of customers or areas where appearance of new requests is likely. In contrast, specialized waiting strategies incorporate knowledge about demand distributions and are parametrized according to the problem properties. Building on the strengths and success factors of previous approaches, we will motivate the need for and propose a novel waiting strategy that is based on an intensity measure.

4.1 General waiting strategies

4.1.1 Simple heuristics

The most straightforward waiting heuristics are the **DriveFirst** and **WaitFirst** strategies which reflect the two most extreme situations. The former strategy is never to wait and always leave immediately for the next location while the latter is always to wait at the current location while feasible. These trivial strategies provide a bound for comparison with more sophisticated approaches. In terms of the PDPTW, the maximum waiting time at a location is determined by the slack in the remaining route.

4.1.2 Heuristics proposed by Mitrovic-Minic and Laporte (2004)

Two more sophisticated strategies were defined and investigated for the PDPTW by Mitrovic-Minic and Laporte (2004). In the **Dynamic** waiting strategy, the route is partitioned and each partition contains a number of consecutive locations that are close to each other. A partitioning algorithm is used to decompose the route to a series of service zones. The partitioning is not static but changes over time. Within a service zone, the *DriveFirst* strategy is used and between zones the *WaitFirst* strategy is applied. The **AdvancedDynamic** strategy is an extension of the former. It propagates the total waiting time along the route by setting the after-zone waiting time proportional to the total time span of the zone.

4.1.3 Heuristics proposed by Branke et al. (2005)

Several waiting heuristics for the DVRP were proposed by Branke et al. (2005). The **Depot** waiting strategy waits at the depot as long as feasible before starting a new tour. The **MaxDistance** strategy waits at the location with maximum distance from the depot as long as possible. The **Location** strategy distributes the slack time equally over all locations of a tour while the **Distance** strategy distributes it proportionally to the driven distance. The **Variable** waiting strategy was derived from an analytically proven optimal strategy. The vehicles do not wait until the time to return to the depot is equal to the slack time. After that, the remaining wait time is distributed proportional to the remaining distance.

The adaption to the PDPTW is straightforward. Instead of distributing the time buffer of returning back to the depot, the slack in the remaining route in terms of the time windows is considered.

4.2 Specialized waiting strategies

4.2.1 Incorporation of stochastic knowledge

The importance of incorporating knowledge about future events in waiting heuristics was pointed out by Ichoua et al. (2006). They proposed a threshold-based heuristic that exploits probabilistic knowledge which will be denoted as the **Stochastic** waiting strategy in the following. The strategy was proposed for a VRP with soft time windows and stochastic customers where the main objective is the maximization of the served customers and the secondary objective the minimization of travel time and lateness.

The waiting heuristic is based on a spatial and temporal separation of the service area into zones. For each zone, which is defined by a geographic area j at a time period m, an arrival rate is specified in terms of a Poisson parameter λ_{jm}. Based on this information, the vehicle waits at the current location as long as it is feasible and not longer than a given maximum wait time Δ if the following conditions are satisfied:

- The distance to the next location is greater than a given threshold α
- The number of already waiting vehicles in the geographic zone is smaller than a given number V
- The probability for a request to occur during the waiting time in the vehicle neighborhood is greater than a given threshold s. A parameter β has been added which scales the size of the neighborhood. In the original definition, the neighborhood was set to the dimensions of a geographic zone and the authors pointed out that extension.

The adaption to the PDPTW mainly concerns the fact that in the original problem formulation soft time windows were applied. In the original formulation a parameter η was incorporated to allow a tolerance to lateness. This parameter has been removed, since in the used problem formulation no time window violations are allowed. Summarizing, the strategy has five parameters that are adjusted according to the problem properties: $\Delta, \alpha, s, V, \beta$.

5 A novel intensity-based heuristic

After analyzing the strengths of existing approaches, three important factors have been identified that influence the decision how long a vehicle should wait at a given location. The aim is to distribute the waiting time in a fine-grained and proactive way. The first factor is based on the observation that, if the next location is far away, it might be beneficial to wait. This has been considered by the partitioning approach used in the *Dynamic* strategy and the threshold for the distance used in the *Stochastic* strategy. The second factor is the relative intensity of the current location and to distribute the available slack proportionally. A proportional distribution has been successfully applied using the *Dynamic* and *Variable* strategies. The third factor is the available slack at a given time. The *Variable* waiting strategy proposes not to wait at the beginning of the route. This indicates, that the available waiting time must be carefully distributed over the route.

Based on the strengths of previous waiting strategies, a new waiting policy is proposed that is based on an intensity measure which in the following will be denoted as the **Intensity** waiting strategy. The motivation to use an intensity measure instead of a stochastic model is motivated by the fact that stochastic information needs to be of a certain quality and might require intensive pre-processing steps in practice as pointed out by Ferrucci et al. (2012). The use of an intensity measure aims at removing these restrictions.

Instead of using a stochastic model, the intensity is calculated based on historical request data which has been for example collected during daily operations. In the case of this study, a set of training instances is used as historical data. The data consists of a set S of historical service requests where each service request $s \in S$ occurs in a certain planning horizon h. Each service request s is revealed at a certain time r_s and contains the pickup location l_s^p and the delivery location l_s^d.

The definition of the intensity measure is based on the transition time between two locations l_1 and l_2 at the current time t. The transition time is the time that passes between the time the vehicle leaves location l_1 and the time the vehicle starts servicing location l_2. It consists of the distance $(Dist)$ between l_1 and l_2 and a possible waiting time before the time window at location l_2 opens. Formally it can be defined as following:

$$TransitionTime(l_1, l_2, t) = Dist(l_1, l_2) + \max\{0, o_{l_2} - (t + Dist(l_1, l_2))\}$$

The intensity of a location l at the current time t is defined as the average transition time for requests in the historical request set S that would have been not revealed yet. The transition times are normalized according to the total planning horizon h. Locations with a lower average transition time have a higher intensity as defined by the following formula:

$$Intensity(l, t) = 1 - \frac{\sum_{\{s \in S : r_s > t\}} \frac{TransitionTime(l, l_s^p, t)}{h}}{|\{s \in S : r_s > t\}|}$$

By combining these considerations into a single policy, the decision how long to wait at a current location R_i in a given route R or to move to the next location R_{i+1} after finishing the service at location R_i at time t_i is based on these three factors:

- The transition time between R_i and the next location R_{i+1} in route R. The transition time is a sum of the distance between R_i and R_{i+1} and the waiting time before the time window opens at R_{i+1}.
- The intensity of location R_i as opposed to location R_{i+1}.
- The slack in the route R at position R_i which is the maximum time the vehicle can wait at position R_i until the route would become infeasible.

To evaluate the benefit of waiting at a given location in the route, the three factors are combined into a single value. The formula for calculating the value of waiting in the given route R at a location R_i at time t_i is listed in Equation 1. The value of waiting is calculated at time t_i before moving to the next location R_{i+1} at which the service would be completed at time t_{i+1}. The function $Slack(R_i, t_i)$ calculates the maximum time at location R_i in a route R at time t_i that can be waited until a time window in the remaining section of the route would be violated. The transition time and slack are normalized according to the remaining time within the total planning horizon h after the location R_i has been served at time t_i.

$$\begin{aligned} v(R_i, t_i) = \alpha' * \frac{TransitionTime(R_i, R_{i+1}, t_i)}{(h - t_i)} \\ + \beta' * \frac{Intensity(R_i, t_i)}{Intensity(R_i, t_i) + Intensity(R_{i+1}, t_{i+1})} \\ + \gamma' * \frac{Slack(R_i, t_i)}{(h - t_i)} \end{aligned} \tag{1}$$

Based on the value of waiting at a certain location the waiting policy which returns the amount of time the vehicle should wait at the current position R_i of route R at time t_i is listed in Equation 2. The parameter ϵ' defines a threshold the value of waiting must exceed, otherwise the vehicle does not wait at the current location. The waiting time in the route is scaled according to the amount v exceeds the threshold. All features are normalized in the interval [0, 1], thus the value v is in the range $[0, v_{max}]$. The maximal value of v is $v_{max} = \max\{0, \alpha'\} + \max\{0, \beta'\} + \max\{0, \gamma'\}$. For locations that have a high value of waiting, a large proportion of the available slack in the route is used.

$$WaitingTime(R_i, t_i) = \max\{0, Slack(R_i, t_i) * (\frac{v(R_i, t_i) - v_{max} * \epsilon'}{v_{max} - v_{max} * \epsilon'}\} \tag{2}$$

In total, the *Intensity* policy has four parameters of the waiting strategy that have to be set according to the problem characteristics: α', β', γ' and ϵ'. The parameters α', β' and γ' define the weighting of the different factors and the parameter ϵ' defines the threshold that has to be exceeded so the vehicle waits at the current location.

6 Simulation-based evolution of waiting strategies

Both specialized waiting strategies impose parameters that have to be set according to the problem characteristics. For example, in some scenarios the region intensity, in other scenarios the transition time might be more important. In Ichoua et al. (2006) the parameters have been set manually by means of preliminary experiments and the need for specialized parameter sets has been pointed out. The observation of non-robustness of specialized strategies in terms of problem characteristics can be explained according to the *no free lunch* theorem (Wolpert and Macready 1997).

These considerations motivate the need for specializing the waiting strategies for different problem characteristics by setting the parameters accordingly. We apply direct policy search based on a simulation model and evolutionary computation for automating this task. Training instances are used to parametrize both the *Stochastic* and *Intensity* strategy accordingly for each instance class containing a set of similar problem instances. The evolved parameter settings are combined to a portfolio of waiting policies containing a different parametrization for each class. The combination of different specialized policies to an algorithm portfolio aims at overcoming the observed non-robustness of the different parameter settings.

Intentions to automate the generation and adaption of heuristics are not new and have been investigated from different perspectives in the past but have not received wide attention in the field of vehicle routing yet. As Pappa et al. (2013) point out, it can be generally observed, that research efforts in different fields including operations research, optimization, and machine learning ultimately evolved from algorithm / parameter selection and control to the automated generation of algorithms. The combination of simulation-based optimization and evolutionary computation allows performing a search in the space of possible policies by trial and error interactions with a simulation model (Whiteson 2012). The policies are evolved offline in a training phase based on a simulation model and are evaluated a-posteriori on a set of unseen test instances which have similar characteristics as the training instances.

Simulation optimization is a powerful tool to evolve and evaluate a number of possible approaches in complex problem environments (Fu (2002)). Especially the combination of simulation with metaheuristics has proven to be fruitful for various application domains (Tekin and Sabuncuoglu (2004)). In this context, HeuristicLab has been applied in several application domains such as production planning (Can et al. (2008), Pitzer et al. (2011)),

Table 1 Parameter settings of the σ-self-adaptive ES

Parents (μ)	2
Children (λ)	16
Maximum Generations	50
Replacement	Comma
Learning parameters (τ / τ_0)	0.4 / 0.4
Mutation	Normal ($\mu = 0, \sigma = \sigma_i$)
Recombination	Heuristic

facility layout planning (Beham et al. (2009)), inventory routing (Vonolfen et al. (2013a)) and dynamic dial-a-ride problems (Beham et al. (2009)). Especially the evolution of policies in dynamic and volatile environments has been shown to be a powerful application of simulation-based optimization.

Combining a simulation model and reinforcement learning, direct policy search allows a compact representation of policies whose structure is either fixed and derived from domain knowledge or is flexible and subject to optimization itself (Moriarty et al. 1999). The search space as well as the components for the generation and adaption of the routing policies is defined by a human expert. As Burke et al. (2013) points out, previous studies have shown that human experts are able to provide good building blocks that can be assembled and adapted in superior ways by algorithmic strategies. The search space thus can be defined either as a set of parameters for a policy with a fixed structure or as a set of formulas with a fixed grammar for flexible policies. Evolutionary algorithms are applied for searching in the space of policies, since they are global search techniques that are known to work well in large and multimodal search spaces (Affenzeller et al. 2009). In the context of this paper, a fixed parameter set is used as a policy representation.

The *Stochastic* strategy has five parameters, namely Δ, α, s, V and β. In terms of the *Intensity* strategy, there are four parameters of the waiting strategy that can be set: α', β', γ' and ϵ'. For optimization, the parameters are encoded as a real-valued vector that has five elements in the case of the *Stochastic* and four elements in the case of the *Intensity* strategy. Each element corresponds to one parameter of the waiting strategy. These parameters are optimized using a self-adaptive evolution strategy (ES) as proposed by Beyer and Schwefel (2002). Additionally, we use heuristic recombination.

The parametrization of the ES is tailored to the computationally complex task of simulation based evolution. Thus, we use a comparatively small population size and the ratio between parents and generated children is high which leads to a greedy search process. The inclusion of self-adaptive parameters and also recombination has turned out to improve the achieved quality in our preliminary experiments. The parameters are listed in Table 1. Each generation, 16 children are generated from the 2 parent individuals by heuristic recombination and a normally distributed mutation. The 2 best children replace the parents for the next generation. The mutation strength is adapted according to the learning parameters. This process is repeated for 50 generations.

To evaluate a certain parametrized waiting strategy, the historical requests are simulated and the strategy with these certain parameters is applied. The past request set is used as training data to set the parameters accordingly. This training set could contain transportation requests that have been recorded over several days. Each day would be a part in the historical data.

Each subset of our training set P can be used as a problem instance to evaluate the performance of the parametrized waiting strategy. To evaluate a parametrization, we use a subset $V \subseteq P$ and perform a separate simulation and optimization run on each instance

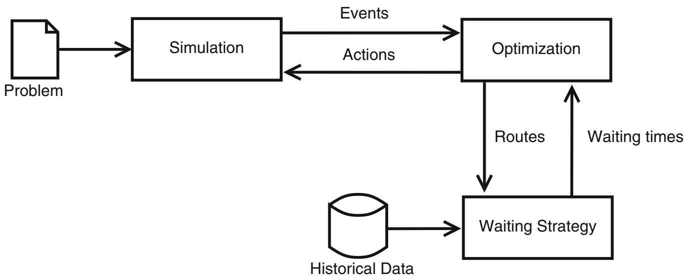

Fig. 1 System architecture of the dynamic vehicle routing system

$v \in V$. The evaluation value of the parametrization is the weighted sum of the driven distance and the utilized vehicles during the simulation of all instances. In that way we can evaluate, how the parametrized waiting strategy performs on historical data.

To increase the possibility that a general parametrization is evolved that will also work well on unknown data, we only use parts of the training set in each generation of the algorithm for fitness evaluation. This validation set V is chosen randomly in each generation, where $|V| = \lfloor |P|/2 \rfloor$. During the algorithm run we store the vector that performs best on the whole training set P and use this vector as the final parametrization of our waiting strategy. This approach allows us to adapt the waiting strategy parameters to a certain problem environment automatically.

7 Computational experiments

To examine different waiting strategies, we created a simulation and optimization system for dynamic vehicle routing. Our simulation and optimization model has been implemented in the open source optimization software HeuristicLab (Wagner (2009)) which is a flexible and extensible software platform for heuristic optimization and has been used in diverse areas such as combinatorial optimization and data-based modeling (Affenzeller et al. (2009)). The test runs have been executed on a desktop PCs with a four-core Intel Core2 CPU running at 2.66 gigahertz accessing 4 GB of memory and running a 64 bit Windows 7 Professional operating system. The dynamic vehicle routing system used for our computational experiments is open source and includes an implementation of all examined waiting strategies. The implementation alongside with the benchmark instances and samples can be downloaded from the HeuristicLab website[1]. The system architecture is illustrated in Figure 1.

The simulation model allows the testing of algorithms and waiting strategies on a large number of cases. Modeling and simulation is a powerful tool for analysis and evaluation of real-world industrial and logistic systems, especially in dynamic and stochastic environments (Longo (2011)). In our case, a simulation model of the real system is necessary while testing different algorithms and waiting strategies because it would be hard to evaluate them in a live system. A problem instance can be loaded which specifies the transportation requests that arrive during the planning process. It is a placeholder for the live system where new information arrives during the planning process. The simulation component responsible for simulating vehicle activities and triggering events. Whenever an event is triggered, the opti-

[1] http://dev.heuristiclab.com/AdditionalMaterial

Table 2 Number of training and test instances for each problem class

Class	Training	Test
C1	5	4
C2	4	4
R1	6	6
R2	6	5
RC1	4	4
RC2	4	4
Total	29	27

mization component is notified and can react accordingly by setting actions that are performed by the vehicles and utilizes a strategy to distribute the waiting time

7.1 Benchmark instances

For evaluating the performance of the waiting strategies we use a fixed set of benchmark instances. As pointed out by Pillac et al. (2013), there is still a lack of widely used standard benchmark instances for dynamic vehicle routing. However, some benchmark sets for dynamic pickup and delivery problems exist in the literature.

The test setup consists of dynamic PDPTW instances that have been proposed by Pankratz (2005) which are derived from the static pickup and delivery instances provided by Li and Lim (2001) which in turn are based on the classical Solomon benchmark instances for the vehicle routing problem with time windows proposed by Solomon (1987). Each of the instances contains 50 dynamically appearing requests.

The 56 different instances are split into different classes which differ in terms of geographical properties and time window properties. In terms of geographical properties, the test set contains clustered, random and mixed locations of transportation requests. The instances also have different time window properties. Some instances have large time windows, others have small time windows. Each instance class represents a combination of these properties, which results in the naming scheme. For example, the instance class C1 contains clustered (C is clustered, R is random, RC is mixed) instances with tight time windows (1 is tight, 2 is large).

The individual classes contain 8–12 problem instances that have similar properties. From each class, half of the instances are used as the test set and the other 4–6 instances as the training set. The training set is used during the direct policy search and is regarded as historical data to derive knowledge about future demands in the form of stochastic or intensity information. The test set is used to compare the different policies on previously unseen instances. The number of instances for each class is listed in Table 2.

In the training phase, no static customers are considered which means that all requests appear dynamically during the planning process. In total, there are 29 training instances in 6 different classes. In the test phase, the different waiting policies are evaluated on the remaining 27 instances with different degree of dynamism ranging from 0 % static customers to 90 % static customers in steps of 10 %. This results in a total of 270 test instances with varying degree of dynamism. Ten independent test runs are performed for each instance. The static customers are chosen randomly according to a uniform distribution for each run.

7.2 Utilization of anticipatory knowledge

The *Stochastic* and the *Intensity* strategy incorporate anticipatory knowledge. Ferrucci et al. (2012) have shown the importance of the information quality on the potential benefit of utilizing knowledge and proposed the structural diversity measure. They state that the

Table 3 Structural diversity (sd) for different problem classes

Class	sd
C1	0.07
C2	0.13
R1	0.03
R2	0.08
RC1	0.06
RC2	0.11
Average	0.08

structural diversity is a crucial criterion to classify the quality of past request information. The identified experimentally, that the structural diversity (sd) has to be at least $sd \geq 0.06$ so that the stochastic knowledge is beneficial. We evaluated the quality of information for each problem class. The results are summarized in Table 3.

On average, the structural diversity is higher than the critical level. However, there is a large variation with respect to the problem classes. Problem instances with clustered customers have a higher ($sd = 0.10$) structural diversity than instances with random ($sd = 0.05$) or mixed customers ($sd = 0.08$). Instances with tight time windows have a lower ($sd = 0.05$) structural diversity than instances with large time windows ($sd = 0.10$) As we will show later, the instance properties effect the potential savings when applying waiting strategies.

For the *Stochastic* waiting strategy a data preprocessing step is required to generate the stochastic knowledge from the past request information. For that purpose we have preprocessed the problem instances of the training set by separating the requests into 225 geographic slices ($1 \leq j \leq 225$) and 15 time slices ($1 \leq m \leq 15$). This has been done for each class of problem instances. This leads to a total of 3375 zones for each problem class. For each zone, an arrival rate is specified in terms of a Poisson parameter λ_{jm} which represents the intensity in a certain geographic region at a certain time. The parameter λ_{jm} was calculated by averaging the number of arriving requests in that zone over all training instances.

To evaluate if the Poisson distribution is a feasible assumption for our instances, we performed a Chi-squared goodness of fit test. For that purpose, we have considered the whole set of training instance set and added the value of the λ parameters to achieve a large enough sample size. Based on the total intensity, we have calculated the expected number of arriving requests for each region. This expected value was then compared to the actual requests. For the training set, the average p value is 0.86. In total, for 97.02 % of all regions the null-hypothesis that the arrival rate is Poisson distributed is accepted with a significance level of 0.05. For the test set the average p value is 0.55 and the hypothesis is accepted for 82.11 % of all regions. This indicates that the probabilistic knowledge still fits the unseen test instances to a large degree. However, the goodness of fit decreases for previously unseen test instances which results from deriving the Poisson parameters from historical data.

7.3 Parameter optimization

By applying the methodology described in Sect. 6, we have optimized the parameters for both the *Stochastic* and the *Intensity* waiting strategy using the training instances. As detailed in Sect. 4, the *Stochastic* strategy has five parameters while the *Intensity* strategy has four parameters to be set. The parameters are adapted to the different instance characteristics of the training instances. For each of the problem classes of the benchmark set that have been initially defined by Solomon (1987), we use the training set to evolve a separate parameter set as described in Sect. 6.

Table 4 Evolved parameters for the *Stochastic* strategy

Class	α	Δ	s	β	V
C1	1.000000	0.843837	0.206368	0.535048	0.000000
C2	0.136841	0.609929	0.575356	0.380122	0.336438
R1	0.000000	0.537976	0.604499	0.092322	0.753874
R2	1.000000	1.000000	0.999918	0.756439	0.948408
RC1	0.693591	0.459593	0.678374	1.000000	0.370727
RC2	0.000000	1.000000	0.793226	0.216212	0.429234

Table 5 Evolved parameters for the *Intensity* strategy

Class	α'	β'	γ'	ϵ'
C1	0.883440	0.519018	0.000000	0.000000
C2	0.000000	0.567995	0.275205	0.000000
R1	0.486998	0.080907	0.747825	0.104138
R2	0.000000	0.000000	0.673014	0.023979
RC1	0.942528	0.949774	0.165874	0.250547
RC2	0.053280	0.097974	0.800274	0.000000

Because of the high computational complexity, we have used an insertion heuristic for route calculation while tuning the parameters of the waiting strategies. The task of setting the parameters requires a lot of computation time, since each parameter setting has to be evaluated by optimizing the training set using the parameterized waiting strategy. Thus we decided to use a less runtime-intensive insertion heuristic for route calculation. However, after the parameter tuning on the training set has been carried out, the adapted waiting strategy can be used in conjunction with more powerful algorithms as we will show later on.

The algorithm used for route calculation is a push forward insertion heuristic (PFIH) which has been originally proposed by Solomon (1987) for the VRPTW. It has been adapted to the PDPTW by Li and Lim (2001). It basically inserts pairs of locations into routes. A pair of locations consists of a pickup and a delivery location. First a route is initialized with a pair based on the distance to depot and time windows. Then iteratively a pair that causes the minimum insertion costs is inserted into the route until no pair can be feasibly inserted. Then a new tour is started. This procedure is repeated until all pairs of customers are routed.

The evolved parameters for the *Stochastic* strategy are listed in Table 4, the parameters for the *Intensity* strategy in Table 5. The results illustrate that a quite diverse set of parameter settings is required depending on the problem characteristics. This corresponds to the findings of Ichoua et al. (2006) who used two different parametrization for the two instance sets. The parameter sets represent the algorithm portfolio where each parameter setting is applied to the respective instance class.

7.4 Performance comparison

After parametrization the specialized strategies, we have evaluated their performance on the training set which is shown in Table 6. As stated earlier, for the training phase a push forward insertion heuristic is applied for route calculation. It is characterized by a comparatively low computational complexity which is required since the policy search requires much computational resources. To validate the parametrization process, we have also included the results of the general waiting strategies. The strategies are compared to not waiting (*DriveFirst*)

Table 6 Performance of the parametrized waiting strategies on the training set using the push-forward insertion heuristic. The total costs are compared to not waiting (DriveFirst). The footnote indicates, where the individual strategies have been presented

Category	Strategy	Distance (μ)	Vehicles (μ)	Costs (μ)	Costs (rel.) (%)
No wait	DriveFirst[a]	2324.86	22.63	70213.75	100.00
General	WaitFirst[a]	2280.64	22.97	71191.75	101.39
	Dynamic[b]	2294.57	22.84	70805.68	100.84
	Adv.Dynamic[b]	2257.48	22.44	69590.81	99.11
	Depot[c]	2343.91	22.96	71232.79	101.45
	Location[c]	2257.18	22.79	70612.74	100.57
	MaxDistance[c]	2296.36	22.50	69785.25	99.39
	Distance[c]	2221.29	22.07	68421.29	97.45
	Variable[c]	2170.14	21.60	66970.14	95.38
Specialized	Stochastic[d]	2205.86	21.33	66205.86	94.29
	Intensity[e]	2165.72	20.56	63832.39	90.91

[a] Trivial strategy
[b] Mitrovic-Minic and Laporte (2004)
[c] Branke et al. (2005)
[d] Ichoua et al. (2006)
[e] Presented in this paper

in terms of total costs which are calculated according to the objective function presented in Sect. 3. Both specialized strategies perform better on the training set than the general ones. The *Stochastic* strategy can save 5.71% while the *Intensity* strategy produces route plans with 9.09% less costs compared to not waiting (*DriveFirst*). This indicates a successful parametrization process.

To compare the performance of all waiting strategies on the test set, we have applied a tabu search algorithm. The unified tabu search has been shown to be very successful for different kinds of routing problems (Cordeau et al. (2002)). A tabu search heuristic that works on various variants of problem formulations was developed by Cordeau et al. (2001) and has also been adapted to pickup and delivery problems by Cordeau and Laporte (2003). For details the reader is referred to their work.

The search process is based on a shift neighborhood where customers are moved from one tour to another using the best possible insertion position. The constraints are adaptively relaxed and tightened to be able to move through infeasible regions of the search space which makes the search process very powerful. In terms of parametrization, we use a tabu tenure of $5log_{10}n$, where n is the number of customers. Every time a new request occurs, it is inserted at the best possible position of the current executing plan and 100 iterations are performed to calculate a new plan. The waiting strategy is applied once a-posteriori on the calculated routes after the procedure has finished. Thus, it does not impose a significant runtime overhead. The average runtime for the PFIH is 25 seconds for each request while executing the tabu search algorithm takes 3 minutes and 42 seconds on average which is 8.88 times more.

The computational results on the overall test set are listed in Table 7. The two best performing heuristics on the overall test set are the *Intensity* strategy and the *Variable* strategy while the *Intensity* strategy performed 0.70 % better. The average savings of the *Intensity* strategy on the overall test set were 4.59% compared to not waiting. In general, the performance of both specialized strategies decreased on the test set in comparison to the training set while the *Intensity* strategy was more robust to the previously unseen test instances.

To test the results for statistical relevance, a multi-sample statistical analysis was performed using the open-source software R (R Core Team 2013) where each sample represents the

Table 7 Performance comparison of different waiting strategies on the overall test set using the tabu search heuristic. The total costs are compared to not waiting (DriveFirst)

Category	Strategy	Distance	Vehicles	Costs	Rel. Costs (%)
No wait	DriveFirst	1980.94	17.89	55659.83	100.00
General	WaitFirst	1959.53	18.37	57059.53	102.51
	Dynamic	1940.37	17.98	55879.25	100.39
	Adv.Dynamic	1931.04	17.76	55223.26	99.22
	Depot	2005.70	18.31	56947.92	102.31
	Location	1925.70	17.85	55464.59	99.65
	MaxDistance	1937.82	17.92	55683.38	100.04
	Distance	1909.11	17.53	54493.56	97.90
	Variable	1898.53	17.19	53481.87	96.09
Specialized	Stochastic	1933.76	17.65	54895.98	98.63
	Intensity	1910.73	17.07	53107.40	95.41

average of the 10 test runs for each of the test instances leading to a sample size of 270 for each waiting policy. After the basic effect of the applied waiting policy on the achieved quality was confirmed using a Friedman test with $p = 2.2e - -16$, a post-hoc analysis was performed with pairwise comparison of the policies (see Table 8).

It can be stated that the *Intensity* and *Variable* policies perform significantly better than not applying a waiting policy (*DriveFirst*) while the *Depot* policy performs significantly worse. For all other policies, no significant difference could be identified as opposed to not waiting. In the context of this multiple comparison, it cannot be stated that a single policy performs better than all other policies. However, the *Intensity* and *Variable* strategies perform significantly better than all other policies. In addition to the full pairwise analysis of all policies, a single comparison between the two best policies without considering the other policies results in the *Intensity* policy performing better than the *Variable* with a significance of $p = 0.01201$.

Apart from these overall results, the benchmark instances are grouped into classes with different degree of dynamism as well as temporal and spatial characteristics allowing the analysis of these influence factors on the performance of the waiting strategies.

7.4.1 Influence of the degree of dynamism

The test set contains instances with different degree of dynamism ranging from 10 to 100 % dynamic customers. To evaluate the optimization potential of incorporating anticipatory knowledge, the results of the tabu search algorithm are compared to previously published results for the static instances without dynamic requests.

In Table 9, the results of the tabu search without anticipatory waiting are compared to the best static solutions presented by Li and Lim (2001) and Ropke and Pisinger (2006). At 10 % dod, the average distance increases by 34.82 % and the average fleet size by 57.67 % compared to the static solution. At 50 % dod, the increase is 100.08 % for the distance and 161.89 % for the fleet size while at 100 % dod the gap to the static solution increases to 145.12 % in terms of distance and 261.56 % in terms of vehicles. The findings are consistent with the results presented by Pankratz (2005) who investigated the influence of the degree of dynamism (dod) on the solution quality and pointed out the importance of incorporating anticipatory knowledge.

An interesting aspect is now how the addition of waiting strategies can decrease the gap to the static solution. For that purpose, the performance of the four best waiting policies

Table 8 Post-hoc pairwise comparison of the policies on the 270 test instances with a a significance level of $p < 0.05$. The Wilcoxon signed rank test was applied and the p values were adjusted using Hochberg's procedure. Significant results are highlighted in bold

	ADW	Intensity	Depot	Drive-First	Distance	DW	Location	MaxDistance	Stochastic	Variable
Intensity	**2E-16**	–	–	–	–	–	–	–	–	–
Depot	**4E-12**	**2E-16**	–	–	–	–	–	–	–	–
DriveFirst	0.6464	**2E-16**	**7E-11**	–	–	–	–	–	–	–
Distance	**1E-03**	**2E-08**	**6E-13**	0.0589	–	–	–	–	–	–
DW	**2E-11**	**2E-16**	**8E-06**	0.9826	**2E-12**	–	–	–	–	–
Location	0.9826	**7E-16**	**7E-07**	0.9826	**2E-14**	0.5382	–	–	–	–
MaxDistance	**7E-03**	**2E-16**	**1E-06**	0.9826	**2E-16**	0.9826	0.1574	–	–	–
Stochastic	0.9826	**4E-14**	**2E-16**	0.7211	0.9826	**3E-05**	0.136	**4E-03**	–	–
Variable	**2E-16**	0.2902	**2E-16**	**1E-11**	**1E-08**	**2E-16**	**2E-16**	**2E-16**	**8E-07**	–
WaitFirst	**1E-09**	**2E-16**	0.9826	0.0516	**2E-16**	**3E-06**	**2E-16**	**5E-15**	**2E-09**	**2E-16**

Table 9 Competitive analysis of the applying the tabu search (TS) without a waiting strategy for instances with different degree of dynamism (dod). The used vehicles and driven distance are compared with the best static solutions found by LiLim and Ropke

Class	LiLim (2001)		Ropke (2006)		TS (dod = 10 %)		TS (dod = 20 %)	
	Distance	Vehicles	Distance	Vehicles	Distance	Vehicles	Distance	Vehicles
C1	828.04	10.00	871.23	9.75	1062.61	13.58	1280.69	16.03
C2	589.84	3.00	589.69	3.00	943.92	6.18	1163.75	8.45
R1	1098.71	9.83	1093.54	9.83	1348.36	14.00	1479.96	16.00
R2	893.40	2.40	888.85	2.40	1289.57	6.12	1504.93	8.22
RC1	1360.32	11.25	1359.98	11.25	1689.12	15.50	1838.03	17.65
RC2	1185.62	3.25	1094.01	3.25	1657.43	7.05	1862.79	8.40
Average	996.83	6.70	987.60	6.67	1331.49	10.51	1517.98	12.56
Class	TS (dod = 30 %)		TS (dod = 40 %)		TS (dod = 50 %)		TS (dod = 60 %)	
	Distance	Vehicles	Distance	Vehicles	Distance	Vehicles	Distance	Vehicles
C1	1416.66	17.48	1509.42	18.68	1641.21	20.10	1718.10	21.35
C2	1341.22	9.63	1521.98	11.55	1658.44	12.65	1865.96	14.03
R1	1696.95	19.33	1740.38	19.68	1877.82	21.70	1968.87	23.43
R2	1703.84	9.94	1848.84	11.24	1984.42	12.98	2157.32	15.42
RC1	2126.63	21.48	2147.76	21.50	2268.54	23.10	2389.41	24.95
RC2	2074.10	10.40	2238.22	11.28	2472.66	13.23	2635.93	13.98
Average	1723.53	14.87	1828.00	15.79	1976.02	17.46	2112.50	19.07
Class	TS (dod = 70 %)		TS (dod = 80 %)		TS (dod = 90 %)		TS (dod = 100 %)	
	Distance	Vehicles	Distance	Vehicles	Distance	Vehicles	Distance	Vehicles
C1	1775.40	22.68	1799.28	23.13	1839.08	24.53	1782.82	24.50
C2	1918.86	14.73	2051.07	15.85	2103.25	16.48	2079.73	16.75
R1	2048.81	24.82	2115.92	26.20	2211.92	28.03	2278.67	29.67
R2	2310.41	16.42	2401.22	17.32	2522.11	19.26	2570.67	20.76
RC1	2523.31	26.58	2676.62	28.85	2800.11	31.10	2888.93	32.75
RC2	2798.85	15.78	2894.73	16.78	2959.01	17.48	2957.60	18.25
Average	2218.91	20.37	2310.68	21.56	2395.85	23.07	2420.80	24.10

depending on the degree of dynamism is illustrated in Fig. 2. The *Intensity* policy performs best in all degree of dynamism except the lowest one with 10% dynamic customers where the *Variable* policy performs 0.19% better. The largest potential savings of 5.82% compared to not waiting (*DriveFirst*) could be achieved with 60% static customers.

A convex behavior of the optimization potential can be observed for the *Intensity* strategy in relation to the degree of dynamism. A possible explanation regarding the influence of dynamic requests is the fact that the lower the degree of dynamism is, the more requests are known in advance and thus the waiting time can be distributed in a more fine-grained way. At the same time, with increasing degree of dynamism, the gap to the static solution and thus the optimization potential of applying anticipatory waiting decreases.

7.4.2 Influence of spatial and temporal characteristics

The test set consisted of instances with different spatial and temporal properties. Concretely, in terms of spatial properties, the performance on instances with randomly/mixed (Table 10) and clustered (Table 11) geographically distributed customers while in terms of temporal properties, the performance on instances with tight (Table 12) and large (Table 13) time windows was evaluated.

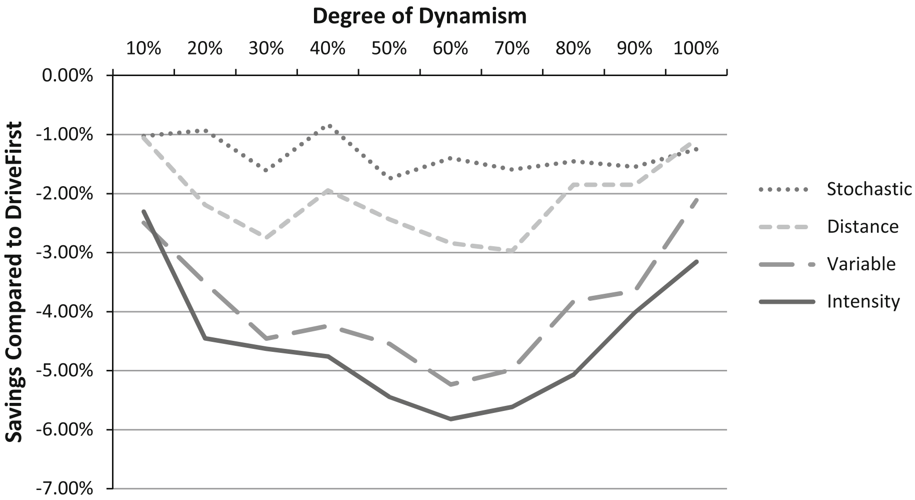

Fig. 2 Savings of the four best waiting policies compared to not waiting (DriveFirst) on the test set containing instances with varying degree of dynamism. Each category (10–100 % dynamic customers) contains 27 test instances

Table 10 Performance comparison of different waiting strategies on geographically random and mixed instances

Category	Strategy	Distance	Vehicles	Costs	Rel. Costs (%)
No wait	DriveFirst	2129.73	18.49	57613.94	100.00
General	WaitFirst	2155.08	19.69	61223.50	106.27
	Dynamic	2117.98	18.98	59065.87	102.52
	Adv.Dynamic	2103.89	18.65	58068.10	100.79
	Depot	2167.43	19.14	59574.80	103.40
	Location	2114.79	19.05	59272.68	102.88
	MaxDistance	2121.56	19.09	59389.98	103.08
	Distance	2096.02	18.71	58232.34	101.07
	Variable	2080.29	18.23	56773.45	98.54
Specialized	Stochastic	2109.63	18.47	57514.89	99.83
	Intensity	2097.27	18.13	56501.48	98.07

The temporal and spatial characteristics have a large influence on the potential savings that can be achieved with anticipatory waiting compared to not applying a waiting heuristic (*DriveFirst*). For the clustered instances, 11.71 % savings were achieved compared to 1.93 % for for the random/mixed instances. Also larger time windows are more beneficial than small time windows. The savings for instances with small time windows were 3.13 % while for large time windows they were 7.14 %.

Overall, the biggest savings potential for anticipatory waiting was identified for instances with clustered customers and time windows as listed in Table 14 where the *Intensity* strtategy performed 11.76% better than the *DriveFirst* strategy.

When comparing the two best waiting strategies (*Intensity* and *Variable*), the temporal and spatial characteristics of the problem instances have a large influence. On instances with clustered customers, the evolved *Intensity* policies perform 1.35 % better, on instances with large time windows they perform 1.11 % better, and on instances with clustered customers as well as large time windows they perform 3.44 % better than the *Variable* policy. On instances

Table 11 Performance comparison of different waiting strategies on geographically clustered instances

Category	Strategy	Distance	Vehicles	Costs	Rel. Costs (%)
No wait	DriveFirst	1627.57	16.46	51018.82	100.00
General	WaitFirst	1495.09	15.23	47170.09	92.46
	Dynamic	1518.53	15.60	48311.03	94.69
	Adv.Dynamic	1520.52	15.65	48466.77	95.00
	Depot	1621.58	16.36	50709.08	99.39
	Location	1476.62	14.98	46420.37	90.99
	MaxDistance	1501.44	15.13	46880.19	91.89
	Distance	1465.20	14.72	45613.95	89.41
	Variable	1466.85	14.73	45664.35	89.50
Specialized	Stochastic	1516.06	15.72	48676.06	95.41
	Intensity	1467.71	14.53	45046.46	88.29

Table 12 Performance comparison of different waiting strategies on instances with tight time windows

Category	Strategy	Distance	Vehicles	Costs	Rel. Costs (%)
No wait	DriveFirst	1918.63	22.19	68473.63	100.00
General	WaitFirst	1974.66	23.29	71844.66	104.92
	Dynamic	1935.39	22.58	69668.96	101.75
	Adv.Dynamic	1917.60	22.31	68836.89	100.53
	Depot	1965.70	23.00	70961.42	103.63
	Location	1924.77	22.42	69186.91	101.04
	MaxDistance	1927.40	22.52	69489.54	101.48
	Distance	1908.63	22.16	68386.49	99.87
	Variable	1883.39	21.59	66644.82	97.33
Specialized	Stochastic	1929.94	22.37	69039.94	100.83
	Intensity	1883.23	21.48	66327.52	96.87

Table 13 Performance comparison of different waiting strategies on instances with large time windows

Category	Strategy	Distance	Vehicles	Costs	Rel. Costs (%)
No wait	DriveFirst	2048.04	13.27	41860.35	100.00
General	WaitFirst	1943.23	13.06	41137.08	98.27
	Dynamic	1945.73	13.03	41028.81	98.01
	Adv.Dynamic	1945.51	12.87	40562.44	96.90
	Depot	2048.76	13.27	41856.46	99.99
	Location	1926.71	12.92	40686.71	97.20
	MaxDistance	1949.05	12.96	40815.21	97.50
	Distance	1909.63	12.54	39531.94	94.44
	Variable	1914.84	12.46	39306.38	93.90
Specialized	Stochastic	1937.87	12.58	39664.02	94.75
	Intensity	1940.35	12.31	38870.35	92.86

with mixed randomly and clustered geographically distributed customers as well as small time windows the picture is less clear while the *Intensity* strategy always offers comparable or better performance as the *Variable* strategy.

Table 14 Performance comparison of different waiting strategies on clustered instances with large time windows

Category	Strategy	Distance	Vehicles	Costs	Rel. Costs (%)
No wait	DriveFirst	1668.49	12.70	39753.49	100.00
General	WaitFirst	1484.83	11.83	36967.33	92.99
	Dynamic	1495.05	11.73	36677.55	92.26
	Adv.Dynamic	1508.18	11.97	37425.68	94.14
	Depot	1656.48	12.41	38886.48	97.82
	Location	1489.26	12.07	37699.26	94.83
	MaxDistance	1508.69	11.92	37268.69	93.75
	Distance	1473.13	11.64	36378.13	91.51
	Variable	1469.04	11.62	36329.04	91.39
Specialized	Stochastic	1447.88	11.30	35355.38	88.94
	Intensity	1470.47	11.20	35077.97	88.24

8 Conclusions and outlook

In this work, we have adapted and evaluated diverse general and specialized waiting heuristics for the dynamic pickup and delivery problem with time windows. We have implemented them in a simulation and optimization environment and compared their performance on a single set of test problem instances.

Combining the strengths of previous approaches, we have proposed a novel waiting strategy that can utilizes past request information without an additional preprocessing step. Furthermore, we have proposed an approach for parametrization of the specialized waiting strategies and for adapting them to different problem characteristics based on direct policy search and simulation-based optimization.

From the computational experiments the following main conclusions can be drawn:

- Specialized waiting strategies can be automatically parametrized on training instances using direct policy search in combination with simpler route construction heuristics to tackle the computational complexity. The evolved parameters also yield benefits when coupled with a more sophisticated routing algorithm.
- The newly proposed *Intensity* waiting heuristic offers robust performance over the whole benchmark set using different parameter sets for each problem class. It has shown more robust to previously unseen instances than the *Stochastic* strategy. On the overall test set, the *Variable* strategy, provides comparable performance. However, the instance properties have a large influence on the benefit of anticipatory waiting. Specialized policies clearly can exploit additional optimization potential compared to general heuristics in cases where it is beneficial to apply anticipatory waiting. On clustered instances, waiting strategies generally yield the largest benefits. This is especially the case in combination with large time windows.
- In general, the incorporation of knowledge is not beneficial in the case of random/mixed instances and small time windows. This corresponds to the findings of Ferrucci et al. (2012) who defined the value of information in terms of a structural diversity measure. As detailed in Sect. 7.2, the structural diversity is considerably lower for random/mixed instance as compared to clustered instances and also for instances with tight time window compared to instances with large time windows. In the case of random/mixed customers and small time windows the value of information lies under the required level of 0.06 to be beneficial.

– Also the degree of dynamism has an impact on the potential savings. The savings potential is largest for instances were around half of the customers appear dynamically with decreasing potential both for instances with higher and lower degree of dynamism.

Future work could focus on discovering more complex and powerful strategies by applying genetic programming as a search algorithm in the policy space. This allows the evolution of policies represented as trees without any pre-defined structure. Previous research in a similar context (Vonolfen et al. 2013b) has shown that rules with a flexible structure can outperform linear representations as used in this work. Additionally, it would be interesting to incorporate self-adaptive behavior in the waiting strategies. One example would be that the individual vehicles are coordinated which prevents them to wait all in the same area in an agent-based approach.

Another research direction is automatically selecting the best policy based on the instance characteristics from the algorithm portfolio which has been done manually in this work by using pre-defined instance classes. As Silverthorn (2012) notes, modern portfolio approaches apply algorithm selection based on machine-learning. Combining the presented methodology with an advanced algorithm portfolio, the selection of an appropriate waiting strategy could be done automatically.

Futhermore, the interpretation of the evolved parameters remains a difficult issue since they seem to span over the whole value range of the parameters depending on the scenario. A possibility to gain deeper insights is linking the findings regarding the meta-level landscape to the fitness landscape of the underlying optimization problem in terms of common features such as plateaus, ruggedness, and distribution of local optima as highlighted by Burke et al. (2013). However, advanced fitness landscape analysis methods require the exploration of a multitude of solution candidates leading to a high computational complexity since each evaluation requires a simulation run. One way to mitigate this issue is the utilization of parallel infrastructures. For that purpose we could use Hive, the cloud-ready parallelization environment provided by HeuristicLab Scheibenpflug et al. (2012).

Acknowledgments The authors would like to thank the attendees of the ORP3 workshop held during July 2012 in Linz as well as the reviewers for their helpful comments, discussions and feedback. The work described in this article was done within the Regio 13 program sponsored by the European Regional Development Fund and by Upper Austrian public funds.

LAND OBERÖSTERREICH

Regio 13 Impulse für OÖ

Regionale Wettbewerbsfähigkeit OÖ 2007–2013

References

Affenzeller, M., Winkler, S., Wagner, S., & Beham, A. (2009). *Genetic algorithms and genetic programming: Modern concepts and practical applications (Numerical Insights)* (1st ed.). UK: Chapman & Hall.

Attanasio, A., Cordeau, J. F., Ghiani, G., & Laporte, G. (2004). Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. *Parallel Computing*, *30*(3), 377–387.

Beham, A., Kofler, M., Wagner, S., Affenzeller, M. (2009). Coupling simulation with heuristiclab to solve facility layout problems. In: Simulation Conference (WSC), Proceedings of the 2009 Winter, pp. 2205–2217. doi:10.1109/WSC.2009.5429238.

Bent, R., Van Hentenryck, P. (2007). Waiting and relocation strategies in online stochastic vehicle routing. In: IJCAI, pp. 1816–1821.

Berbeglia, G., Cordeau, J. F., & Laporte, G. (2010). Dynamic pickup and delivery problems. *European Journal of Operational Research*, *202*(1), 8–15. doi:10.1016/j.ejor.2009.04.024.

Beyer, H. G., & Schwefel, H. P. (2002). Evolution strategies: A comprehensive introduction. *Natural Computing*, *1*(1), 3–52.

Branke, J., Middendorf, M., Noeth, G., Dessouky, M. (2005). Waiting strategies for dynamic vehicle routing. Transportation Science 39:298–312. doi:10.1287/trsc.1040.0095.

Burke, E. K., Gendreau, M., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., et al. (2013). Hyper-heuristics: A survey of the state of the art. *Journal of the Operational Research Society*, *64*(12), 1695–1724.

Can, B., Beham, A., Heavey, C. (2008). A comparative study of genetic algorithm components in simulation-based optimisation. In: Proceedings of the 40th Conference on Winter Simulation, Winter Simulation Conference, WSC '08, pp 1829–1837, URL http://dl.acm.org/citation.cfm?id=1516744.1517063

Cordeau, J., & Laporte, G. (2003). A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research Part B: Methodological*, *37*(6), 579–594.

Cordeau, J., Laporte, G., & Mercier, A. (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational research society*, *52*(8), 928–936.

Cordeau, J., Gendreau, M., Laporte, G., Potvin, J., & Semet, F. (2002). A guide to vehicle routing heuristics. *Journal of the Operational Research Society, 53*(5), 512–522.

Cordeau, J. F., & Laporte, G. (2007). The dial-a-ride problem: Models and algorithms. *Annals of Operations Research*, *153*(1), 29–46.

Coslovich, L., Pesenti, R., & Ukovich, W. (2006). A two-phase insertion technique of unexpected customers for a dynamic dial-a-ride problem. *European Journal of Operational Research*, *175*(3), 1605–1615.

Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, *6*(1), 80–91. doi:10.2307/2627477.

Eksioglu, B., Vural, A. V., & Reisman, A. (2009). The vehicle routing problem: A taxonomic review. *Computers and Industrial Engineering*, *57*(4), 1472–1483. doi:10.1016/j.cie.2009.05.009.

Ferrucci, F., Bock, S., & Gendreau, M. (2012). A pro-active real-time control approach for dynamic vehicle routing problems dealing with the delivery of urgent goods. *European Journal of Operational Research, 225*(1), 130–141.

Fu, M. C. (2002). Feature article: Optimization for simulation: Theory vs. practice. *INFORMS Journal on Computing*, *14*(3), 192–215. doi:10.1287/ijoc.14.3.192.113.

Gendreau, M., & Potvin, J. (2004). Issues in real-time fleet management. *Transportation Science*, *38*(4), 397–398.

Ghiani, G., Guerriero, F., Laporte, G., & Musmanno, R. (2003). Real-time vehicle routing: Solution concepts, algorithms and parallel computing strategies. *European Journal of Operational Research*, *151*(1), 1–11. doi:10.1016/S0377-2217(02)00915-3.

Golden, B., Raghavan, S., & Wasil, E. (2008). *The vehicle routing problem: Latest advances and new challenges. Operations Research/Computer Science Interfaces*. Dordrecht: Springer.

Gutenschwager, K., Niklaus, C., & Voß, S. (2004). Dispatching of an electric monorail system: Applying metaheuristics to an online pickup and delivery problem. *Transportation science*, *38*(4), 434–446.

Hentenryck, P. V., & Bent, R. (2009). *Online stochastic combinatorial optimization*. Cambridge: The MIT Press.

Ichoua, S., Gendreau, M., & Potvin, J. Y. (2006). Exploiting knowledge about future demands for real-time vehicle dispatching. *Transportation Science*, *40*, 211–225. doi:10.1287/trsc.1050.0114.

Ichoua, S., Gendreau, M., Potvin, JY. (2007). Planned route optimization for real-time vehicle routing. In: Dynamic fleet management, Springer, pp 1–18.

Li, H., Lim, A. (2001). A metaheuristic for the pickup and delivery problem with time windows. In: Tools with artificial intelligence, Proceedings of the 13th International Conference on, pp 160–167, Doi:10.1109/ICTAI.2001.974461.

Longo, F. (2011). Advances of modeling and simulation in supply chain and industry. *Simulation*, *87*(8), 651–656.

Mes, M., van der Heijden, M., & van Harten, A. (2007). Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems. *European Journal of Operational Research*, *181*(1), 59–75.

Mitrovic-Minic, S., & Laporte, G. (2004). Waiting strategies for the dynamic pickup and delivery problem with time windows. *Transportation Research Part B: Methodological*, *38*(7), 635–655. doi:10.1016/j.trb.2003.09.002.

Mitrovic-Minic, S., Adviser-Krishnamurti, R., & Adviser-Laporte, G. (2001). *The dynamic pickup and delivery problem with time windows*. Burnaby: Simon Fraser University.

Mitrovic-Minic, S., Krishnamurti, R., & Laporte, G. (2004). Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows. *Transportation Research Part B: Methodological*, *38*(8), 669–685. doi:10.1016/j.trb.2003.09.001.

Moriarty, D. E., Schultz, A. C., & Grefenstette, J. J. (1999). Evolutionary algorithms for reinforcement learning. *Journal of Artificial Intelligence Research*, *11*, 241–276.

Pankratz, G. (2005). Dynamic vehicle routing by means of a genetic algorithm. *International Journal of Physical Distribution and Logistics Management*, *35*(5), 362–383. doi:10.1108/09600030510607346.

Pappa, GL., Ochoa, G., Hyde, MR., Freitas, AA., Woodward, J., Swan, J. (2013). Contrasting meta-learning and hyper-heuristic research: The role of evolutionary algorithms. Genetic Programming and Evolvable Machines, 1–33.

Parragh, S. N., Doerner, K. F., & Hartl, R. F. (2008). A survey on pickup and delivery problems. *Journal für Betriebswirtschaft*, *58*(1), 21–51.

Pillac, V., Gendreau, M., Guéret, C., & Medaglia, A. L. (2013). A review of dynamic vehicle routing problems. *European Journal of Operational Research*, *225*(1), 1–11.

Pitzer, E., Beham, A., Affenzeller, M., Heiss, H., Vorderwinkler, M. (2011). Production fine planning using a solution archive of priority rules. In: Logistics and industrial informatics (LINDI), 2011 3rd IEEE International Symposium on, pp 111–116, Doi:10.1109/LINDI.2011.6031130.

Psaraftis, H. (1988). Dynamic vehicle routing problems. In: Vehicle routing: Methods and studies, Elsevier Science Publishers, pp 223–249.

R Core Team. (2013). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria, URL http://www.R-project.org

Ropke, S., & Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation science*, *40*(4), 455–472.

Sáez, D., Cortés, C. E., & Núñez, A. (2008). Hybrid adaptive predictive control for the multi-vehicle dynamic pick-up and delivery problem based on genetic algorithms and fuzzy clustering. *Computers and Operations Research*, *35*(11), 3412–3438.

Savelsbergh, M. (1995). The general pickup and delivery problem. *Transportation Science*, *29*(1), 17–29.

Scheibenpflug, A., Wagner, S., Kronberger, G., Affenzeller, M. (2012). Heuristiclab hive-an open source environment for parallel and distributed execution of heuristic optimization algorithms. In: 1st Australian Conference on the Applications of Systems Engineering ACASE'12, p 63.

Silverthorn, BC. (2012). A probabilistic architecture for algorithm portfolios. PhD thesis, The University of Texas at Austin, may.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, *35*, 254–265. doi:10.1287/opre.35.2.254.

Tekin, E., & Sabuncuoglu, I. (2004). Simulation optimization: A comprehensive review on theory and applications. *IIE Transactions*, *36*(11), 1067–1081.

Tjokroamidjojo, D., Kutanoglu, E., & Taylor, G. D. (2006). Quantifying the value of advance load information in truckload trucking. *Transportation Research Part E: Logistics and Transportation Review*, *42*(4), 340–357.

Van Hemert, JI., La Poutré, JA. (2004). Dynamic routing problems with fruitful regions: Models and evolutionary computation. In: Parallel problem solving from nature-PPSN VIII, Springer, pp 692–701.

Vonolfen, S., Affenzeller, M., Beham, A., Lengauer, E., & Wagner, S. (2013a). Simulation-based evolution of resupply and routing policies in rich vendor-managed inventory scenarios. *Central European Journal of Operations Research*, *21*(2), 379–400.

Vonolfen, S., Beham, A., Kommenda, M., Affenzeller, M. (2013b). Structural synthesis of dispatching rules for dynamic dial-a-ride problems. In: Proceedings of the 14th international conference on Computer Aided Systems Theory, Springer.

Wagner, S. (2009). Heuristic optimization software systems - Modeling of heuristic optimization algorithms in the HeuristicLab software environment. PhD thesis, Johannes Kepler University, Linz, Austria.

Waisanen, H. A., Shah, D., & Dahlehm, M. A. (2008). A dynamic pickup and delivery problem in mobile networks under information constraints. *Automatic Control, IEEE Transactions on*, *53*(6), 1419–1433.

Whiteson, S. (2012). Evolutionary computation for reinforcement learning. In: Reinforcement learning, Springer, pp 325–355.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *Evolutionary Computation, IEEE Transactions on*, *1*(1), 67–82.

Xiang, Z., Chu, C., & Chen, H. (2008). The study of a dynamic dial-a-ride problem under time-dependent and stochastic environments. *European Journal of Operational Research*, *185*(2), 534–551.

Zeimpekis, V., Tarantilis, C., Giaglis, G., & Minis, I. (2007). *Dynamic fleet management. Operations research/computer science interfaces*. New York: Springer.