# Bi-objective Orienteering: Towards a Dynamic Multi-objective Evolutionary Algorithm

Jakob Bossek[1], Christian Grimme[1(✉)], Stephan Meisel[1], Günter Rudolph[2], and Heike Trautmann[1]

[1] Department of Information Systems, University of Münster, Leonardo-Campus 3, 48149 Münster, Germany
{bossek,grimme,meisel,trautmann}@wi.uni-muenster.de

[2] Department of Computer Science, TU Dortmund University, Otto-Hahn-Str. 14, 44227 Dortmund, Germany
guenter.rudolph@tu-dortmund.de

**Abstract.** We tackle a bi-objective dynamic orienteering problem where customer requests arise as time passes by. The goal is to minimize the tour length traveled by a single delivery vehicle while simultaneously keeping the number of dismissed dynamic customers to a minimum. We propose a dynamic Evolutionary Multi-Objective Algorithm which is grounded on insights gained from a previous series of work on an a-posteriori version of the problem, where all request times are known in advance. In our experiments, we simulate different decision maker strategies and evaluate the development of the Pareto-front approximations on exemplary problem instances. It turns out, that despite severely reduced computational budget and no oracle-knowledge of request times the dynamic EMOA is capable of producing approximations which partially dominate the results of the a-posteriori EMOA and dynamic integer linear programming strategies.

**Keywords:** Multi-objective optimization · Metaheuristics · Vehicle routing · Combinatorial optimization · Dynamic optimization

## 1 Introduction

Bi-objective orienteering belongs to the class of vehicle routing problems. It differs from classical Traveling Salesperson Problems (TSP) in that the number of cities resp. customers is not fixed but rather a certain number of dynamic customer requests have to be handled on the way from the start to the end depot. Naturally, both the overall tour length as well as the number of unvisited customers are desired to be minimized and we would like to dynamically react to new customer requests so that previously optimized tours can be adjusted in an efficient and optimal way. The design of an appropriate optimization algorithm given this scenario is not trivial, especially as, additionally, decision makers' preferences regarding the importance of both objectives have to be taken


K. Deb et al. (Eds.): EMO 2019, LNCS 11411, pp. 516–528, 2019.
https://doi.org/10.1007/978-3-030-12598-1_41

into account which might vary in the course of the operation time of the whole tour. This paper introduces such a real-time expert system in terms of a specific dynamic evolutionary multi-objective algorithm (EMOA) integrating local search strategies via inexact TSP solvers. The algorithm was designed by relating to the detailed problem insights gained by previous studies which approached the problem in a retrospective, offline way leading to a Pareto-front approximation exploiting the full information about the dynamic problem characteristics.

Experimental studies provide a proof-of-concept analysis of the proposed approach. It will be shown that it has the potential of outperforming competitive integer linear programming (ILP) strategies in terms of solution quality. Moreover, the algorithm is capable of generating Pareto-front approximations which come very close to and even partially dominate the solutions which resulted from the offline approach. First results show that clustered instances are more challenging compared to random ones. As purely numerical performance assessment is not trivial due to a lack of an appropriate performance indicator capturing all requirements stated above, sophisticated visualizations illustrate algorithm characteristics.

The paper is organized as follows: Sect. 2 gives an overview on related work, followed by a detailed description of our proposed dynamic multi-objective evolutionary algorithm in Sect. 3. Experimental results are provided in Sect. 4 and summarized in Sect. 5, supplemented by an outlook on promising further research building on the consolidated findings.

## 2 Background and Related Work

### 2.1 Static Multi-objective Optimization Problems

Let $\mathbb{X}$ and $\Theta$ be nonempty sets and $f(x;\theta) = (f_1(x;\theta), \ldots, f_d(x;\theta))^\mathsf{T}$ a vector-valued mapping with $d \geq 2$ functions $f_i : \mathbb{X} \times \Theta \to \mathbb{R}$ for $i = 1, \ldots, d$, where $x$ is variable and $\theta \in \Theta$ a tuple of fixed parameters. If these functions are to be minimized simultaneously, they are called *objective functions* of the *multi-objective optimization problem* $\min\{f(x;\theta) : x \in X\}$ with *decision set* $X \subseteq \mathbb{X}$. The optimality of a multi-objective optimization problem (MOP) is defined by the concept of *dominance*.

Let $u, v \in F \subseteq \mathbb{R}^d$ where $F$ is equipped with the partial order $\preceq$ defined by $u \preceq v \Leftrightarrow \forall i = 1, \ldots d : u_i \leq v_i$. If $u \prec v \Leftrightarrow u \preceq v \wedge u \neq v$ then $v$ is said to be *dominated by* $u$. An element $u$ is termed *non-dominated* relative to $V \subseteq F$ if there is no $v \in V$ that dominates $u$. The set $\mathsf{ND}(V, \preceq) = \{u \in V \mid \nexists v \in V : v \prec u\}$ is called the *non-dominated set* relative to $V$.

If $F = f(X;\theta)$ is the *objective set* of some MOP with decision set $X \subseteq \mathbb{R}^n$ and objective function $f(\cdot)$ then the set $F^* = \mathsf{ND}(f(X;\theta), \preceq)$ is called the *Pareto-front* (PF). Elements $x \in X$ with $f(x) \in F^*$ are termed *Pareto-optimal* and the set $X^*$ of all Pareto-optimal points is called the *Pareto set* (PS).

### 2.2 The Dynamic Multi-objective Vehicle Routing Problem

The dynamic vehicle routing problem we consider in this work consists of one vehicle that visits customer locations over time. The set of customers $C \backslash \{1, N\} = C^m \cup C^o$ resolves into $C^m$, the subset of initially known customers and the set $C^o$ of additional locations, which become known randomly while the vehicle is en route. The vehicle starts its tour at a given location 1 (start depot) and ends at a different location $N$ (end depot). Locations that are known initially must be visited by the vehicle (including depots), whereas locations that become known in the course of time are optional. We refer to the set of optional customers that have arrived until time $t$ as $C^o_{\leq t}$.

Clearly, a static MOP (as defined above) has to be adapted as the Pareto-front and Pareto-set now depend on dynamic parameters $\theta$, i.e., in general we have $F^*_\theta$ and $X^*_\theta$. In a dynamic MOP the parameters are no longer constant but variable over time. As a consequence, a dynamic MOP (DMOP) at time step $t \geq 0$ is given by $\min\{f(x; \theta_t) : x \in X\}$ where $(\theta_t)_{t \geq 0}$ is a sequence of parameter tuples with time index $t \geq 0$. For each point in time $t \geq 0$ we could solve a static MOP with solution $F^*_{\theta_t}$ and $X^*_{\theta_t}$ and might regard the sequences of both sets as the final solution. For a general survey on dynamic MOO, see [1].

However, this solution concept has little practical relevance. Instead, we specify a closed time interval $\Delta_t$ and monitor (the quality of) the best solutions that can be achieved within the time interval. A similar solution concept can be found in [15]. This is repeated multiple times, where at the end of each so-called *era*, a decision maker (DM) is provided with the best solutions of that era. The quantitative assessment of the sequence of best solutions found within the time interval heavily depends on the application scenario.

Specifically, our VRP is dynamic in the sense that decisions about the vehicle's route (which of the customer locations known so far to visit, and how to sequence these locations) are made repeatedly over time by a decision maker. Although dynamic decision making has been an important research topic in the field of vehicle routing (see, e.g., [9,14]), and although static variants of bi-objective orienteering problems have been considered by a number of authors (e.g., [2,5,8,10]), the research on dynamic bi-objective orienteering problems still is in a very early stage. So far, only few authors work on dynamic multi-objective vehicle routing problems, most of them proposing solution approaches in terms of methodological frameworks that rely on evolutionary computation (e.g., [6,13]).

Over the past decade a number of authors have solved (single-objective) dynamic orienteering problems by combining integer linear programming with waiting strategies (see [11] for an overview). The idea is to maximize the number of visited customers over a given fixed time horizon by using linear programming for calculation of a routing plan at each decision time. Therefore, simple waiting strategies[1] are used, i.e., the vehicle remains idle at locations in these plans, hoping for a close-by customer request to occur in the near future.

This approach can be transformed into an a-posteriori benchmark solution for dynamic bi-objective optimization algorithms by selecting the best waiting strategy and by then solving the problem several times, each time with a different bound of the maximum tour length in the linear program. In Sect. 4 we use the waiting strategies and the linear program described in [11] as benchmark for the dynamic multi-objective evolutionary algorithm introduced in the following Sect. 3.

## 3 The Dynamic Multi-objective Evolutionary Algorithm

Our dynamic EMOA for the considered orienteering problem is based on the a-posteriori EMOA introduced in [10] with refined adjustments—in particular in initialization and mutation—to meet the requirements of the dynamic setting.

**Algorithm 1.** Dynamic EMOA

**Require:** Instance $I = (C^m, C^o)$, time resolution $\Delta_t$, nr. of time slots $n_t$
$t := 0$
tour := localSearch($C^m$) ▷ No dynamic customers, i.e., solve single-obj. problem
$t := t + \Delta_t$
$P = \mathsf{NIL}$
driven.tour := findDrivenTour(tour, $t$)
**for** $i$ in 1 to $n_t$ **do**
    $(P, F(P))$ := EMOA($I$, driven.tour, $t$, $P$)
    tour := decide($P, F(P)$)
    $t := t + \Delta_t$
    driven.tour := findDrivenTour(tour, $t$)

**Algorithm 2.** EMOA

**Require:** Instance $I = (C^m, C^o)$, driven.tour, time $t$, population of previous era $Q$, population size $\mu$
**for** $i$ in 1 to $\mu$ **do**
    $P_i$ := initIndividual($I$, driven.tour, $t$, $Q_i$) ▷ $Q_i$ is NIL on start
$F(P)$ := evaluateFitness($P$)
**while** stopping condition not met **do**
    $O$ := mutate($P$)
    $O$ := localSearch($O$)
    $(P, F(P))$ := Select($P \cup O$)
**return** $(P, F(P))$

**Algorithm 3.** initIndividual

**Require:** instance $I = (C^m, C^o)$, driven.tour, current time $t$, template individual $y$
**if** not $y$ is NIL $\wedge$ $y$ is feasible **then**
    **return** $y$
$C^o_{\leq t}$ := Dyn. customers arrived so far
$D := C^o_{\leq t} \setminus$ driven.tour
$x.b, x.p, x.t$ are vectors of length $N - 2$
$x.b_i := 1, x.p_i := 0 \,\forall\, i \in C^m \vee i \in$ driven.tour
$x.t$ := concat(driven.tour, randPerm($C \setminus$ driven.tour))
$x.p_i := 1/|D| \,\forall i \in D$
$u := \mathcal{R}(1, \ldots, |D|)$ ▷ Rnd. number
Set $x.b_i := 1$ for $u$ rnd. customers from $D$
**if** not $y$ is NIL **then**
    $x$ := transfer($x$, $y$)
**return** $x$

**Algorithm 4.** mutate

**Require:** Population $P$, swap prob. $p_{\text{swap}}$, nr. of swaps $\sigma_{\text{swap}}$
**for** $x \in P$ **do**
    flip $x.b_i$ with probability $x.p_i$
    $t_{\text{active}}$ := seq. of active customers in $x.t$
    **if** $r \sim R(0, 1) \leq p_{\text{swap}}$ **then**
        **for** 1 to $\sigma_{\text{swap}}$ **do**
            swap two random pos. in $t_{\text{active}}$
**return** $P$

[1] Two prominent strategies used also in this work for comparison reasons are Drive First (DF) and Distributed Waiting (DW). While in DF the vehicle only waits at its current customer location if both waiting time is available and the planned route only contains the end depot, the latter strategy distributes the amount of available waiting time equally among all customer locations of the current planned route.

We start with a high-level description of the dynamic EMOA framework accompanied by a example first and discuss the more complex solution encoding scheme and mutation later on. The dynamic EMOA (see Algorithm 1) is basically a wrapper around the static version introduced in [7] which uses NSGA-II [4] as the encapsulating meta-heuristic (see Algorithm 2). It is started at time $t = 0$. Note, that at this point in time only mandatory customers $C^m$ are available. Since no subset selection is necessary in this special case the problem is of single-objective nature and we simply apply local search[2] to approximate the optimal tour serving all mandatory customers (see Fig. 1 left) and the first *era* ends. Here, the DM is given only a single choice and there is nothing left to do. In subsequent eras $j = 1, \ldots, n_t$ however, already time $j \cdot \Delta_t$, $\Delta_t$ being the adjustable time resolution, has passed and hence more and more dynamic customers request for service. To be precise, in era $j$ dynamic customers with request times $r_i \in ((j-1) \cdot \Delta_t, j \cdot \Delta_t]$ arrive. In each such era the static EMOA is started feeding in the partial tour already driven by the vehicle (as time goes by, the vehicle already may have served some of the mandatory and/or dynamic customers). After termination, the resulting approximations are handed over to the DM who needs to choose exactly one solution (see line 8 in Algorithm 1 and Fig. 1 middle and right for example).

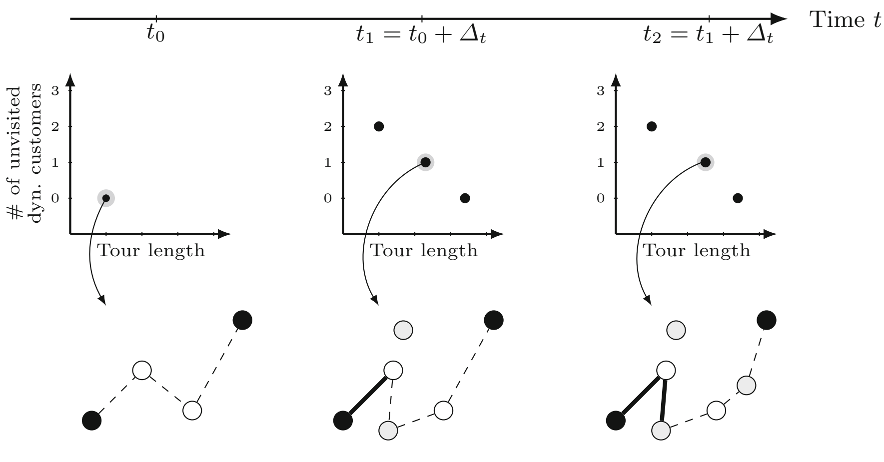


**Fig. 1.** Exemplary progress of the dynamic EMOA. The scatter plots show the Pareto-front approximations with selected solutions highlighted (●). Below the decision maker choices are depicted (depots ●, mandatory customers ○ and dynamic customers ○). A dashed path indicates the tour chosen by the decision maker while the thick solid prefix path highlights the partial tour already driven.

[2] We adopt EAX [12] as the local search procedure with focus on tour length minimization. Note, that we need to solve a shortest Hamiltonian path problem, but EAX is a TSP solver. Thus, before application of the local search procedure, the problem is transformed into a TSP by a sequence of modifications to the distance matrix (see [10] for details).

The encoding of candidate solutions needs to account for both the subset selection of customers and the minimization of the Hamiltonian path serving all selected customers. Thus, three essential vectors of length $N-2$ are maintained: (1) a permutation vector holds the sequence of customers, i.e., the actual tour, (2) vector $b = (b_2, \ldots, b_{N-1}) \in \{0,1\}^{N-2}$ indicates whether a customer $i \in \{2, \ldots, N-1\}$ is active ($b_i = 1$) or inactive ($b_i = 0$) and (3) vector $p \in [0,1]^{N-2}$ holds the flip probabilities for the mutation operator (see below). Building the initial population (see Algorithm 3) is a complex process since several dynamic aspects need to be considered: (1) we need to ensure, that both mandatory and already visited (potentially dynamic) customers are active and cannot be removed by mutation (line 6). Hence, $b_i = 1$ and $p_i = 0$ for those customers. (2) the partial tour already driven must not be changed and hence the first positions of the permutation vector correspond to this sequence (line 7). (3) We transfer knowledge from the final population of the previous era in order to not start from scratch. This is achieved by simply copying the individual if it is still feasible (line 2). Otherwise, we transfer as much information by keeping active customers active and maintaining the tour as far as possible (line 12). The initialization procedure guarantees feasibility of initial solutions.

Mutation is twofold to account for both objectives (see Algorithm 4). First, available customers are added or removed by flipping each bit $b_i$ independently with probability $p_i$. Next, with probability $p_{\text{swap}} \in (0,1]$ some random position exchanges in the permutation vector are performed limited to active customers not yet visited, i.e., which are not part of the already driven part of the tour. Note, that mutation is non-destructive and hence feasibility is maintained. Finally, mutated solutions are subject to local search at certain generations. Here, we apply EAX [12] with the last customer of the already driven tour as the start node and the end depot as the destination node omitting already visited customers. It is important to stress, that the local search operator is focused on tour length minimization only, since we consider this objective to be more difficult. Furthermore, take notice that EAX does not take request times into consideration. Hence, the length of the resulting tour is a lower bound on the true tour length. We take the solid foundations and results laid down in [3,10] as a justification for this approach.

## 4 Computational Experiments

**Experimental Setup:** In order to evaluate the dynamic EMOA introduced in Sect. 3, we perform proof-of-concept experiments. We select 5 instances with $N = 100$ customers (including depots) each: one instance with locations distributed uniformly at random in the Euclidean plane and 4 instances with $2, 3, 5$ and 10 clusters respectively form the instances introduced in [10]. The proportion of dynamic customers is chosen to be 75% for all instances, in order to specifically analyze the working principles of our approach.

We fix the time resolution $\Delta_t = 100$ and determine the number of eras as $\lceil \max_{i \in C}(r_i)/\Delta_t \rceil + 1$, where $r_i \geq 0$ is the request time of customer $i \in C$. The final parametrization of the dynamic EMOA is gathered in Table 1. These settings deserve further explanation: Preliminary experiments were performed testing different parameter settings. More precisely, we varied local search (on/off), transfer of knowledge of previous eras (on/off), the swap-mutation probability $p_{\text{swap}} \in \{0.2, 0.4, \ldots, 1\}$ and the way available dynamic customers are being distributed in the initial solutions of each era (uniform/binomial). Unsurprisingly, local search (see our a-posteriori study in [3]) and knowledge transfer are beneficial settings to not discard progress already being made. The latter two varied parameters, $p_{\text{swap}}$ and the distribution of dynamic customers in initial solutions, however, show strong interaction with local search. It turns out, that a high swap probability with binomial distribution leads to poor front coverage in areas with a high number of unvisited customers. This can be explained as follows: Local search pushes solutions to the left (focus on tour length minimization). Now assume, we are given a very good solution with respect to tour length and apply mutation with high swap probability. Assume further, that mutation deactivates some customers. Clearly, since the tour can only become even shorter, this step pushes the solution to the top left area of the Pareto-front approximation. Since the tour is already close to optimal, the subsequent swaps introduce edge crossings and have a destructive effect with overwhelming probability. Consequently, the mutated individual shifts to the right (larger tour length) and is likely to be dismissed by the following survival selection. In case of binomial distribution each available dynamic customer is activated with probability $1/2$. Hence, the number of activated dynamic customers is binomially distributed with expected value $N_t^d/2$ where $N_t^d$ is the number of dynamic available customers at time $t > 0$. The probability that the actual number deviates from the expectation is rather low and hence is concentrated heavily around it. Thus, this type of initialization in combination with activated local search and high swap probability tends to produce the above mentioned poor coverage. We bypass this problem by adopting a uniform distribution of dynamic customers, i.e., each number of active available customers is active with equal probability.

**Table 1.** Dynamic EMOA parameterization.

| Parameter | Setting |
|---|---|
| Generations per era | 65.000 |
| $\mu, \lambda$ | 100 |
| $p_{\text{swap}}$ | 0.6 |
| $\sigma_{\text{swap}}$ | $N/10 = 10$ |
| LS application in generations | initial, half-time, last |
| Cutoff time for LS | 1s |
| Transfer knowledge from last era | on |
| Distribution of dynamic customers | Uniform |

In this study, we simulate different decision maker strategies which are based on order ranking of the first objective (tour length). In a nutshell, the $n$ solutions of the EMOA are ordered in ascending order of tour length[3] and the DM decides for the $\lfloor \text{rank} \cdot n \rfloor$-ranked solution with rank $\in \{0.25, 0.5, 0.75\}$ in each era. Clearly, in real world scenarios, the DM can make different decisions in each era to adapt to different situations and we are aware of the limitations of our DM policies. However, for a first study and for an automated evaluation of the approach, we consider these fixed three strategies a good starting point.

We performed 10 independent runs on each instance. The implementation of our dynamic EMOA is available at a public repository[4].

**Results:** On the one hand, the following results contribute to the understanding of the working principle of the dynamic evolutionary approach. On the other hand, they show the applicability and provide a feeling for the potential of such an approach.

Figure 2 comprises two representative series of depictions of the intermediate Pareto-front approximations generated in each era of the algorithm run, for uniform (top) and clustered (bottom) topologies of customers. Each era bases on decisions made during previous process. For the decision making process three ranks were fixed. In each plot, the Pareto-fronts of the dynamic approach are colored per era from dark blue (first era) to light green (last era). For visual comparison, Pareto-front approximations of the a-posteriori EMOA recently proposed in [3] and of an $\varepsilon$-constrained-based ILP approach using the dynamic single-objective strategies [10] described in Sect. 2 are shown. Note, that – for comparison reasons – the results of all eras have been transformed to the a-posteriori solution space. Additionally, the sub-figures contain horizontal lines colored according to the eras. Those lines define a true upper bound of available unvisited customers for that era. It is clear that depending on the current era and previous actions of the DM, the upper bound decreases.

The first interesting finding is, that our approach is capable of outperforming the ILP-based a-posteriori strategy directly and the MOEA-based a-posteriori approach on the long run. Although the a-posteriori approaches possess complete information on the (virtually) dynamic service requests, the dynamic approach is able to generate comparable or even better solutions without foresight - especially for uniform topologies. For clustered topologies, the approach often outperforms the ILP-based strategy in its final era and sometimes even becomes comparable to the a-posteriori EMOA solutions. This is especially true, when the (higher) decision maker rank favors the second objective (number of unvisited customers).

The at a first glance surprising superiority over the a-posteriori approach is rooted in the fact, that the search space for the a-posteriori problem is much larger than the restricted dynamic scenario, in which previous decisions and a

[3] Note that in the bi-objective case this leads to a sorting in descendant order of the number of unvisited customers.

[4] Repository: https://github.com/jakobbossek/dynvrp/.

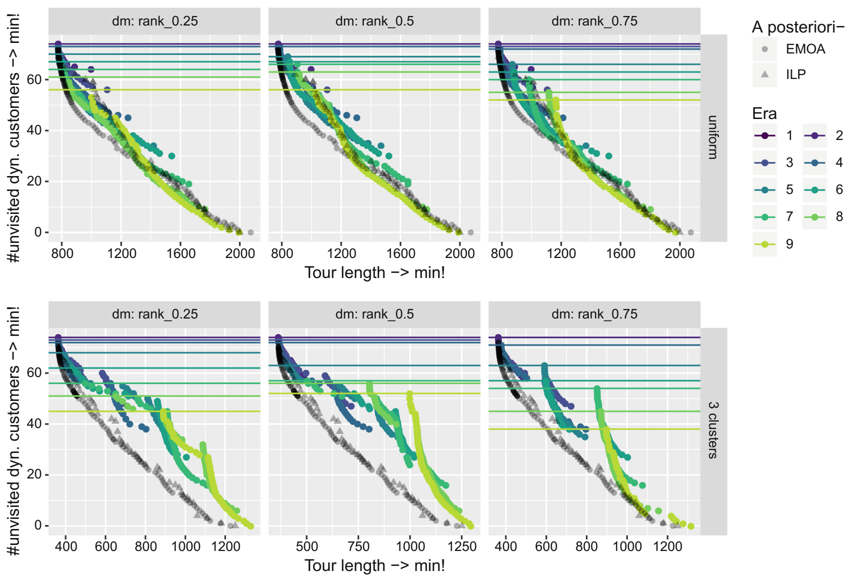


**Fig. 2.** Scatter plots of representative Pareto-front approximations for three different decision maker strategies on uniform (top) and clustered (bottom) topologies. Points are colored by era. Colored horizontal lines indicate a true upper bound for the number of unvisited available dynamic customers w.r.t. the era. For comparison front approximations based on complete a-posteriori knowledge obtained in [3] and [10] are shown. (Color figure online)

fixed partial tour reduce the search space dramatically. While in the a-posteriori case for selecting an optimal subset of visited customers, all customers are eligible, the dynamic approach can narrow the subset selection to still available customers w.r.t. the already fixed partial tour.

The analysis of the representative results in Fig. 2 for uniform and clustered topologies[5] shows that era results for uniform topologies are closer to the a-posteriori results than era results for clustered topologies. To provide a more detailed insight into this aspect, we show respective embeddings of found (intermediate) solution tours for both topologies in Fig. 3. For both settings, the DM selected solutions of era 1, era 4 and era 9 are plotted including the path to already visited customers (bold) and the plan for the remaining tour considering currently available customers. In the top row of Fig. 3, the tour starts with the mandatory customers and successively integrates new appearing customers into the tour. As customers are uniformly distributed in search space, later appearing customers can easily be integrated in the not yet fixed part of the tour.

In contrast to this, for clustered instances like in Fig. 3 (bottom), new customers appear over time in different clusters. Here, the mutation operator

[5] We find similar behavior for all investigated (but not shown) topologies for multiple repetitions.

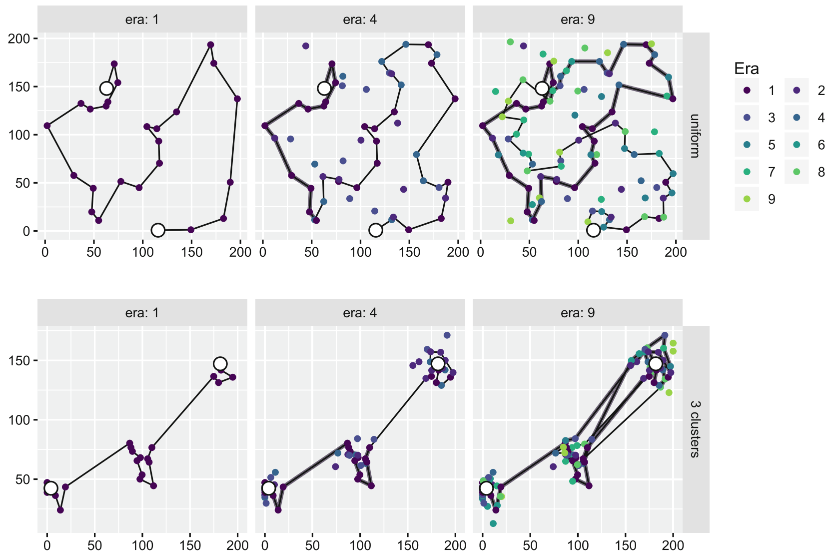


**Fig. 3.** Embedding of the actual tours the decision maker (rank 0.75) decides for at the end of eras 1, 4 and 9 respectively. The bold part of the tour is already fixed/visited and is hence not subject to change in subsequent eras.

(i.e. random activation of customers) and preferences of the DM potentially have major impact on the quality of the solution. On the one hand, mutation may include customers from a distant cluster. On the other hand, strong DM preference on maximizing the number of visited customers (set to 0.75 for example shown in Fig. 3) may force the algorithm to select newly available customers from a distant cluster. Both will lead to long traveled distances in the resulting tour and as such deteriorate the overall trade-off solution compared to the a-posteriori results. This suggests, that future work should deal with elaborated mutation mechanisms that try to avoid (or alternatively repair) multiple long distance travels between clusters.

In order to evaluate the process of decision making and to test our approach for stability w.r.t. multiple runs, we plot the intermediate decision results leading to the final realized tour in Fig. 4. According to our standard color scheme, we show picked solutions of the parametrized DM for all eras and over all runs. Additionally, the centroid of the final realizations is shown as black-framed dot. The solid black line connects the centroids of the intermediate decisions and shows the decision path. For the representative results in Fig. 4 we can conclude two aspects: (1) The dynamic approach is stable over multiple runs, i.e. the variance in produced solutions is low. (2) Compared to the a-posteriori approximated Pareto-front, the final decisions made under the dynamic evolutionary scheme quite perfectly reflect the parametrized ranking set up for the DM.

Note, that all qualitative results presented here also hold for the investigated topologies (different number of clusters) in the same way.

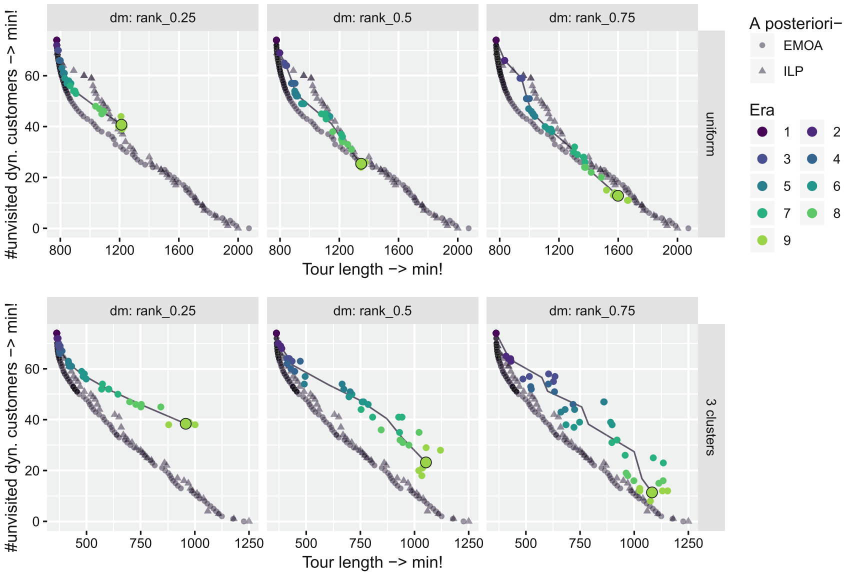


**Fig. 4.** Paths of decisions taken by different decision maker policies. Colored points represent the decisions made in the corresponding eras for all 5 independent runs. Paths run through the centroids of per era decisions. The centroid of final decisions is highlighted as black-framed dot at the end of the decision path. (Color figure online)

## 5 Conclusions and Outlook

Our previous studies on bi-objective orienteering from an offline perspective resulted in detailed insights into problem characteristics and challenges for respective multi-objective (evolutionary) algorithm design. This paper proposes an online-approach for multi-objective dynamic optimization, which is required in practice for adjusting a currently active vehicle tour to new customer requests. A crucial feature of the new real-time optimization system is the possibility of incorporating user preferences regarding both objectives, which can either be given as a fixed a-priori rule or interactively adjusted along the algorithm run whenever an adjustment decision of the current tour has to be made.

Initial proof-of-concept experiments indicate that ILP strategies are outperformed by our approach in terms of solution quality and efficiency. The latter point is especially important regarding scalability w.r.t. the instance size. With increasing instance size ILP strategies will become infeasible in terms of the real-time system requirement. Moreover, ILP methods are based on a-priori fixed waiting strategies in contrast to flexible preference incorporation. Additionally, in our settings, the dynamic approach comes close or even dominates certain parts of the Pareto-front approximation gained by the retrospective offline EMOA. We find, however, that dynamic optimization becomes more challenging on clustered instances due to higher probability of long distances travels between customers.

Next steps will include a comprehensive benchmark study on a large set of representative instances in terms of proportion of dynamic customers (different from here considered 75% optional customers), degree of clustering as well as instance sizes. Realistically, the current instance size is already quite large in terms of one vehicle serving 100 customers a day. From the hybridization point of view, the influence of local search has to be investigated for the online case. A straightforward extension will be allowing for more than one vehicle, which increases practical relevance but poses additional challenges onto dynamic EMOA design. For a systematic validation, a suitable performance indicator simultaneously incorporating the quality of the final Pareto-front approximation, the any-time performance along the EMOA run, robustness across multiple runs, and the degree of user preference fulfillment, has to be derived.

**Acknowledgments.** J. Bossek, C. Grimme, S. Meisel and H. Trautmann acknowledge support by the European Research Center for Information Systems (ERCIS).

## References

1. Azzouz, R., Bechikh, S., Ben Said, L.: Dynamic multi-objective optimization using evolutionary algorithms: a survey. In: Bechikh, S., Datta, R., Gupta, A. (eds.) Recent Advances in Evolutionary Multi-objective Optimization. ALO, vol. 20, pp. 31–70. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-42978-6_2
2. Berube, J.-F., Gendreau, M., Potvin, J.-Y.: An exact ∈-constraint method for bi-objective combinatorial optimization problems: application to the traveling salesman problem with profits. Eur. J. Oper. Res. **194**(1), 39–50 (2009)
3. Bossek, J., Grimme, C., Meisel, S., Rudolph, G., Trautmann, H.: Local search effects in bi-objective orienteering. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2018, pp. 585–592. ACM, New York (2018)
4. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
5. Filippi, C., Stevanato, E.: Approximation schemes for bi-objective combinatorial optimization and their application to the TSP with profits. Comput. Oper. Res. **40**(10), 2418–2428 (2013)
6. Ghannadpour, S.F., Noori, S., Tavakkoli-Moghaddam, R.: A multi-objective vehicle routing and scheduling problem with uncertainty in customers' request and priority. J. Comb. Optim. **28**, 414–446 (2014)
7. Grimme, C., Meisel, S., Trautmann, H., Rudolph, G., Wölck, M.: Multi-objective analysis of approaches to dynamic routing of a vehicle. In: ECIS 2015 Completed Research Papers. Paper 62. AIS Electronic Library (2015)
8. Jozefowiez, N., Glover, F., Laguna, M.: Multi-objective meta-heuristics for the traveling salesman problem with profits. J. Math. Model. Algorithms **7**(2), 177–195 (2008)
9. Meisel, S.: Anticipatory Optimization for Dynamic Decision Making. Operations Research/Computer Science Interfaces Series, vol. 51. Springer, New York (2011). https://doi.org/10.1007/978-1-4614-0505-4
10. Meisel, S., Grimme, C., Bossek, J., Wölck, M., Rudolph, G., Trautmann, H.: Evaluation of a multi-objective EA on benchmark instances for dynamic routing of a vehicle. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2015, pp. 425–432. ACM, New York (2015)

11. Meisel, S., Wölck, M.: Evaluating idle time policies for real-time routing of a service vehicle. In: ECIS 2015 Completed Research Papers. Paper 132. AIS Electronic Library (2015)
12. Nagata, Y., Kobayashi, S.: A powerful genetic algorithm using edge assembly crossover for the traveling salesman problem. INFORMS J. Comput. **25**(2), 346–363 (2013)
13. Nahum, O.E., Hadas, Y.: A framework for solving real-time multi-objective VRP. In: Żak, J., Hadas, Y., Rossi, R. (eds.) EWGT/EURO -2016. AISC, vol. 572, pp. 103–120. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-57105-8_5
14. Pillac, V., Gendreau, M., Guéret, C., Medaglia, A.L.: A review of dynamic vehicle routing problems. Eur. J. Oper. Res. **225**(1), 1–11 (2013)
15. Raquel, C., Yao, X.: Dynamic multi-objective optimization: a survey of the state-of-the-art. In: Yang, S., Yao, X. (eds.) Evolutionary Computation for Dynamic Optimization Problems, pp. 85–106. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38416-5_4