Greedy Randomized Adaptive Search Procedure for Close Enough Orienteering Problem

Petra Štefaníková Petr Váňa Jan Faigl

ABSTRACT

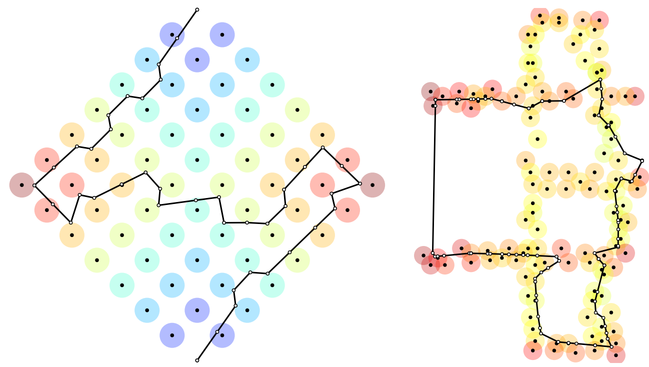
In this paper, we address the Close Enough Orienteering Problem (CEOP) that is motivated to find the most rewarding route visiting disk-shaped regions under the given travel budget. The CEOP differs from the regular OP in the continuous optimization of finding the most suitable waypoint locations to collect the reward associated with each region of interest in addition to the selection of the subset of the regions and sequence of their visits as in the OP. We propose to employ the Greedy Randomized Adaptive Search Procedure (GRASP) combinatorial metaheuristic to solve the addressed CEOP, in particular, the GRASP with Segment Remove. The continuous optimization is addressed by the newly introduced heuristic search that is applied in the construction phase and also in the local search phase of the GRASP. The proposed approach has been empirically evaluated using existing benchmarks, and based on the reported comparison with existing algorithms, the proposed GRASP-based approach provides solutions with the competitive quality while its computational requirements are low.

KEYWORDS

GRASP, Greedy Randomized Adaptive Search Procedure, CEOP, Close Enough Orienteering Problem, OP, Orienteering Problem

ACM Reference Format:

Petra Štefaníková, Petr Váňa, and Jan Faigl. 2020. Greedy Randomized Adaptive Search Procedure for Close Enough Orienteering Problem. In *The 35th ACM/SIGAPP Symposium on Applied Computing (SAC '20), March 30-April 3, 2020, Brno, Czech Republic*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3341105.3374010



(a) Set 64 (b) Set 130

Figure 1: Examples of the found solutions of the CEOP instances. The colored disks represent the disk-shaped neighborhoods of the locations of interest, and the color denotes the reward, where high rewards are in the red, and low rewards are in the blue.

1 INTRODUCTION

The Orienteering Problem (OP) belongs to the class of routing problems with profits [10] where we are searching for the tour from a given initial location to visit the most rewarding locations and terminates at the given final location such that the total tour length does not exceed the maximal travel budget [12]. In the OP, each location has associated reward, and the problem stands to determine the subset of the most rewarding locations that can be visited within the travel budget. The OP can be thus considered as a combination of the Knapsack Problem and the Traveling Salesman Problem (TSP) in finding the sequence of visits to the selected locations to ensure the tour length is shorter than the travel budget.

Routing problems with profits are motivated by problems where the travel budget does not allow visitation of all possible locations such as tourist trip design problem [24]. The herein addressed generalization of the regular OP is motivated by data collection task with remote sensing that has been introduced by the authors of [1]. In data collection tasks, the travel cost can be saved by avoiding precise visitation of the location by exploiting the fact that the reward (data or range measurements) can be collected from the specified distance ϱ from the particular location of interest, see examples of the found solutions depicted in Fig. 1. Although it is called the OP with Neighborhoods, in the presented approach, we prefer to call the problem the *Close Enough Orienteering Problem* (CEOP) to emphasize the neighborhood is of the disk shape because the reward can be collected from any point that is up to ϱ distant apart from the location.

Several heuristic algorithms have been proposed for the OP [13, 18, 22, 23] together with relatively well-established benchmark instances proposed by Tsiligirides [20] and Chao et al. [3]; however, there are only two approaches for the CEOP reported in the literature so far. The first approach is based on unsupervised learning [1], firstly improved in [9] and later improved in [6] using the learning of the Growing Self-Organizing Array (GSOA) [5]. The second approach has been proposed in [17], and it is based on the Variable

The authors are with the Computational Robotics Laboratory of the Artificial Intelligence Center (AIC) at the Dept. of Computer Science, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic.

The presented work has been supported by the Czech Science Foundation (GAČR) under research project No. 19-20238S.

SAC '20, March 30-April 3, 2020, Brno, Czech Republic

ACM ISBN 978-1-4503-6866-7/20/03…$15.00
https://doi.org/10.1145/3341105.3374010

Neighborhood Search (VNS) metaheuristic [14] initially deployed in the regular OP in [19]. Even though [17] addresses the Dubins OP where the curvature of the path is limited, the problem is formally identical with the OP with Neighborhoods, just instead of the waypoint location, the continuous optimization searches for the optimal heading of the vehicle at the location. Besides, the same VNS-based approach has been directly employed in solution of the Close Enough Dubins Orienteering Problem [8] and OPN [16].

The GSOA is a growing array of nodes that iteratively adapts the nodes towards the locations of interest [5] by the repeated determination of the closest point of the path represented by the array to the randomly selected location of interest. A new node is added to the array at the location of the closest point, and it is then adapted towards the location while its neighboring nodes in the array are also adapted, but with the power that decreases with the distance of the node to the newly added node. However, the main idea of the GSOA-based method to the CEOP is that the continuous optimization is directly addressed during the sequencing part [9], while the VNS-based approach [17] is based on the sampling of the continuous domain into a finite discrete set. Although both heuristics (based on the GSOA and VNS) provide approximate solutions of the CEOP, they both have particular drawbacks. The GSOA is a fast constructive heuristic, where the solution does not improve once the unsupervised learning converges to a stable state of the array of nodes [6]. On the other hand, the VNS-based solution is capable of finding better solutions than the GSOA. However, it is significantly more computationally demanding, e.g., about more than three orders of magnitude than the GSOA, as reported in [7].

In this paper, we propose to address the drawbacks of the previous methods to the CEOP by using the *Greedy Randomized Adaptive Search Procedure* (GRASP) combinatorial metaheuristic [11] that is enhanced by the heuristic search of the optimal waypoint location utilized in the GSOA. In particular, we consider the GRASP with Segment Remove (GRASP-SR) [15] variant of the GRASP, which is reported to provide better results than other approaches for the OP.

The GRASP consists of two phases, the constructive phase, and the local search phase. Thus, the waypoint location is determined using the closest point of the corresponding disk of the newly inserted location to the route. The proposed approach is similar to how a new winner node is determined in the GSOA for routing problems with disk-shaped neighborhoods [5]. Besides, the idea is employed in the local search phase to shorten the tour by adjusting waypoint locations and thus enable the insertion of additional locations while still keep the tour length within the travel budget.

The reported results indicate that the proposed GRASP-based approach for the CEOP provides better results than the GSOA in instances with a low travel budget, while the computational requirements are lower than the GSOA and significantly lower than the VNS-based approach. The main contributions of the presented work are considered in the novel variant of the GRASP for the CEOP, empirical evaluation using standard benchmarks, and comparison with the state-of-the-art solutions based on the GSOA and VNS.

The rest of the paper is organized as follows. The problem is formally defined in the following section. The proposed method is introduced in Section 3 and the evaluation results are reported in Section 4. The concluding remarks and future work are summarized in Section 5.

2 PROBLEM STATEMENT

Let $V = \{v_1, \ldots, v_n\}$ be a set of n locations, and $\|x, y\|$ denotes Euclidean distance between the two locations x, y. Each location v_i has a reward $r_i > 0$ except for the initial location v_1 and final location v_n for which the reward is zero, i.e., $r_1 = r_n = 0$. The *Close Enough Orienteering Problem* (CEOP) stands to find the most rewarding tour from v_1 to v_n that collects rewards associated with the locations V such that the total tour length does not exceed the given travel budget $T_{\max}$. The reward of a location is collected from a point with the distance to the location that is shorter or equal than the given sensing range ϱ.

The CEOP stands to determine the subset of k locations from V to be visited together with the sequence to their visits and the particular waypoint locations at which the rewards are collected; hence, the final tour can be defined as a permutation Σ of the locations' indexes $\Sigma = \langle\sigma_1, \ldots, \sigma_k\rangle$, where $2 \le k \le n$ and $\sigma_1 = 1$ and $\sigma_k = n$ because the initial and final locations are prescribed. Having the sequence of visits Σ, the path visiting the selected locations is defined by the waypoint locations $P = \langle p_1, \ldots, p_k\rangle$ such that the corresponding waypoint location p_i is within ϱ distance from v_{σ_i}. The final tour described by P (further called path) always includes the initial location $p_1 = v_1$ and final location $p_k = v_n$, and its length is denoted $\mathcal{L}(P, \Sigma)$ that can be expressed as

$$\mathcal{L}(P, \Sigma) = \sum_{i=2}^{k} \|p_{i-1}, p_i\| . \tag{1}$$

The CEOP can be formulated as the optimization Problem 2.1 to determine the number of locations k to be visited and the sequence of their visits Σ together with the particular points of the visits P such the total sum of the collected rewards $R(\Sigma)$ is maximized and the path length $\mathcal{L}(P, \Sigma) \le T_{\max}$. Notice a feasible solution exists iff $\|v_1, v_n\| \le T_{\max}$. The CEOP is at least NP-hard because the CEOP becomes the ordinary OP for zero sensing range, that is known to be NP-hard [23].

Problem 2.1 Close Enough Orienteering Problem (CEOP).

$$\begin{aligned}
\max_{k, \Sigma, P} \quad & R(\Sigma) = \sum_{i=1}^{k} r_{\sigma_i} \\
\text{s. t.} \quad & \mathcal{L}(P, \Sigma) \le T_{\max} \\
& 2 \le k \le n \\
& \Sigma = \langle\sigma_1, \ldots, \sigma_k\rangle, \quad 1 \le \sigma_i \le n,\ \sigma_i \ne \sigma_j \text{ for } i \ne j \\
& \sigma_1 = 1\ ,\ \sigma_k = n \\
& P = \langle p_1, \ldots, p_k\rangle, \quad p_1 = v_1, \quad p_k = v_n \\
& \|p_i, v_{\sigma_i}\| \le \varrho \text{ for } i = 2, \ldots, k-1
\end{aligned}$$

In the rest of the paper, we consider the locations $v \in V$ and points $p \in P$ are from a plane, i.e., $v \in \mathbb{R}^2$ and $p \in \mathbb{R}^2$. Notice that during the solution of the CEOP by the proposed GRASP-based method, the number of locations to be visited k is varying depending on the current best solution found so far. Therefore, $|P|$ and $|\Sigma|$ denotes the current number of the locations in the path P, i.e., $k = |\Sigma| = |P|$.

3 PROPOSED GRASP-BASED SOLUTION TO THE CEOP

The proposed heuristic approach is based on the existing GRASP with Segment Remove (GRASP-SR) [15] for the regular OP that has been extended to solve the herein addressed CEOP (Problem 2.1). The GRASP is a constructive heuristic with two main phases: the Construction Phase (CP), and the Local Search Phase (LSP). The GRASP-SR algorithm starts with the CP, where the initial path consists of the initial location $p_1 = v_1$ and the final location $p_k = v_n$ that are both prescribed. Then, the locations from V are iteratively tried to be inserted into the current path P by the procedure addLocation until P remains unchanged. After that, a local search procedure localSearch is applied to improve the initially constructed path.

Algorithm 1: GRASP-SR for the CEOP

Input: V – n locations to be visited, each $v_i \in V$ with the reward r_i, where v_1 and v_n are the specified initial and final locations, respectively.
Output: P, Σ – Final path from v_1 to v_n with the sequence of visits Σ to the subset of V.

$P \leftarrow \langle v_1, v_n \rangle$ `// Initial path`
$\Sigma \leftarrow \langle 1, n \rangle$
repeat
  $P, \Sigma \leftarrow$ addLocation(P, Σ) `// Constr. phase`
until P is unchanged
$P, \Sigma \leftarrow$ localSearch(P, Σ) `// Local search phase`
return P, Σ

An overview of the whole GRASP-SR is depicted in Algorithm 1 and it follows the original GRASP algorithm for the OP; however, the main proposed enhancements are in the insertion procedure, where the particular waypoint location p_i is determined for each location $v_{\sigma_i} \in V$ being inserted into the tour. In the rest of this section, the GRASP-SR algorithm is described in detail, and the proposed enhancements are presented in Section 3.1.

In the CP, a new location (and the corresponding waypoint location) is iteratively tried to be inserted into the path by the procedure addLocation that is depicted in Algorithm 2. A location v_i is tried to be inserted into the current path by the `insertion` operator defined in Eq. (2).

Definition 3.1 `insertion`(P, Σ, i)**.**

$$
\begin{aligned}
P' &= \left\langle p_1, \ldots, p_{j-1}, v_i, p_j, \ldots, p_{|P|} \right\rangle , \\
\Sigma' &= \left\langle \sigma_1, \ldots, \sigma_{j-1}, i, \sigma_j, \ldots, \sigma_{|\Sigma|} \right\rangle .
\end{aligned} \tag{2}
$$

The `insertion` operator greedily includes v_i into the current path P and v_i is added to the path at the position j, for which the prolongation is minimal, see Eq. (3)

$$
j = \operatorname*{arg\,min}_{j'=2}^{|P|} (\mathcal{L}(P') - \mathcal{L}(P)) . \tag{3}
$$

The new path is then examined to have its length within the budget T_{max}. If T_{max} is not exceeded, the path (P', Σ') is inserted into the set of potential paths for the next iteration that is called *the candidate list* denoted CL. Otherwise, segments are removed from the new path to satisfy T_{max}.

Algorithm 2: addLocation$(P, \Sigma, b = -1)$

Input: P, Σ – the current path.
Input: b – blocked index; if not specified $b = -1$ is used.
Output: $\overline{P}, \overline{\Sigma}$ – the updated path.

$CL \leftarrow \emptyset$ `// A candidate list of solutions`
for $i \in \{1, \ldots, |V|\}$ **do**
  if $(i \notin \Sigma) \wedge (i \neq b)$ **then**
    $P', \Sigma' \leftarrow$ `insertion`(P, Σ, i) `// Using Eq. (2)`
    if $\mathcal{L}(P', \Sigma') \leq T_{max}$ **then**
      $CL \leftarrow CL \cup \{(P', \Sigma')\}$
    else `// Segment Remove phase`
      $\beta \leftarrow 2$
      for $\alpha \in \{2, \ldots, |P'| - 1\}$ **do**
        if $\beta < \alpha$ **then**
          $\beta \leftarrow \alpha$
        $P'' \leftarrow P' \setminus \{p_\alpha, \ldots, p_\beta\}$
        $\Sigma'' \leftarrow \Sigma' \setminus \{\sigma_\alpha, \ldots, \sigma_\beta\}$
        while $(\beta + 1 < |P'|) \wedge (\mathcal{L}(P'', \Sigma'') > T_{max})$ **do**
          $\beta \leftarrow \beta + 1$
          $P'' \leftarrow P' \setminus \{p_\alpha, \ldots, p_\beta\}$
          $\Sigma'' \leftarrow \Sigma' \setminus \{\sigma_\alpha, \ldots, \sigma_\beta\}$
        if $\mathcal{L}(P'', \Sigma'') \leq T_{max}$ **then**
          if $(R(\Sigma'') > R(\Sigma)) \vee$
          $\vee (R(\Sigma'') = R(\Sigma) \wedge \mathcal{L}(P'', \Sigma'') < \mathcal{L}(P, \Sigma))$ **then**
            $CL \leftarrow CL \cup \{(P'', \Sigma'')\}$
  if $CL \neq \emptyset$ **then**
    $CL' \leftarrow$ `restrict`(CL)
    $\overline{P}, \overline{\Sigma} \leftarrow$ random(CL')

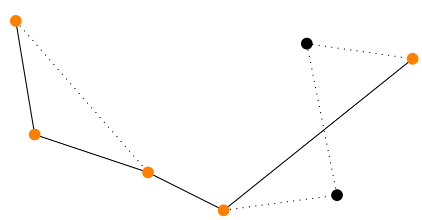


Figure 2: An example of the insertion of the location that prolongs the path above the travel budget T_{max}. Therefore, the two black locations are removed to satisfy the budget. The original path before the insertion and removal is visualized as a sequence of dotted segments.

In the case of segments remove, the method sequentially goes through the new path (P', Σ'), it removes locations from index α to β, and thus creates a new path (P'', Σ''). The path P'' is then examined to be within the travel budget T_{max}, and once P'' meets T_{max}, the path is added to CL; and the algorithm continues to remove other segments from P'. In this way, the algorithm attempts to remove all potential segments in linear time complexity that can be bounded by $O(|\Sigma'|)$. An example of the insertion and removal of two locations is depicted in Fig. 2 and an example of iterations of the Segment Remove is shown in Fig. 3.

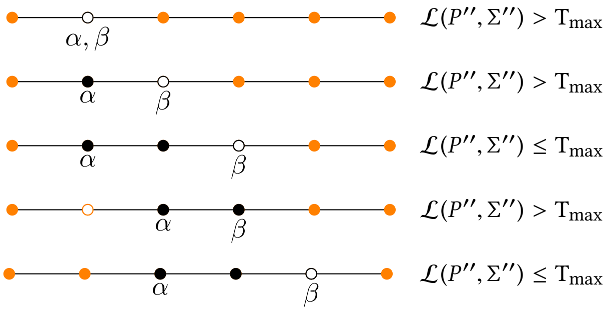


Figure 3: The phase Segment Remove when the path length exceeds the travel budget $\mathrm{T}_{\max}$ after insertion of the new location. When the travel budget is not met, the position index β shifts to the next location in the path, and the path length is examined again. Once $\mathrm{T}_{\max}$ is satisfied, the position index α shifts to the next location.

After the examination of all locations from V in the ADDLOCATION procedure, the final candidate list CL' is created by the `restrict` operator defined in Eq. (4) for the particular iteration of the CP. CL' then contains paths with associated reward greater or equal to c_{best} of the reward R_{best} of the best-known solution found so far. Following [2] and further based on the empirical evaluations, the value of c_{best} is set to 20 %.

Definition 3.2 `restrict`(CL).

$$\begin{aligned} R_{\text{best}} &= \max\left\{R(\Sigma) \mid (P, \Sigma) \in CL\right\} , \\ CL' &= \left\{(P, \Sigma) \mid (P, \Sigma) \in CL, R(\Sigma) \geq c_{\text{best}}\, R_{\text{best}}\right\} . \end{aligned} \tag{4}$$

After an insertion of p_i for the new location v_{σ_i}, the path (P, Σ) is chosen randomly from the restricted candidate list CL'. The construction phase terminates when CL is empty, which means that no better solution has been found.

Finally, after the initial construction of the solution (P, Σ), the solution is improved in the LOCALSEARCH procedure. The operator `remove`, defined in Eq. (5), is utilized to iteratively remove the location v_{σ_j} at the position $j \in \{2, \ldots, |P| - 1\}$. For each path with the removed location, the GRASP-SR uses the `2-Opt` optimization heuristic [4] to eliminate possible crossing segments by a local exchange of the particular sequence part.

Definition 3.3 `remove`(P, Σ, j).

$$\begin{aligned} P' &= \left\langle p_1, \ldots, p_{j-1}, p_{j+1}, \ldots, p_{|P|-1} \right\rangle , \\ \Sigma' &= \left\langle \sigma_1, \ldots, \sigma_{j-1}, \sigma_{j+1}, \ldots, \sigma_{|\Sigma|-1} \right\rangle . \end{aligned} \tag{5}$$

Once a location is removed from the path, a new attempt to insert not yet visited location is performed by ADDLOCATION procedure. If such an attempt on the path P' improves the collected rewards or it has the same rewards but $\mathcal{L}(P', \Sigma') < \mathcal{L}(P, \Sigma)$, the path (P', Σ') replaces (P, Σ). The local search phase is summarized in Algorithm 3.

Algorithm 3: LOCALSEARCH(P, Σ)

Input: P, Σ – the current path.
Output: $\overline{P}, \overline{\Sigma}$ – the updated path.

$(\overline{P}, \overline{\Sigma}) \leftarrow (P, \Sigma)$
repeat
  for $i \in \{2, |P| - 1\}$ **do**
    $P', \Sigma' \leftarrow$ `remove`$(\overline{P}, \overline{\Sigma}, i)$ `// Using Eq. (5)`
    $P', \Sigma' \leftarrow$ `2-opt`(P', Σ') `// See [4]`
    $P' \leftarrow$ `optimization`(P') `// Using Eq. (6)`
    $b \leftarrow \overline{\sigma}_i$ `// Index of blocked location from` $\overline{\Sigma}$
    repeat
      $P', \Sigma' \leftarrow$ ADDLOCATION(P', Σ', b)
    until P' is unchanged
    if $(R(\Sigma') > R(\overline{\Sigma})) \lor$
    $\lor\ (R(\Sigma') = R(\overline{\Sigma}) \land \mathcal{L}(P', \Sigma') < \mathcal{L}(\overline{P}, \overline{\Sigma}))$ **then**
      $\overline{P} \leftarrow P'$
      $\overline{\Sigma} \leftarrow \Sigma'$
until $\overline{P}$ is unchanged

3.1 Finding Optimal Position of Location Visit

There are two places in the original GRASP-SR algorithm where the particular waypoint location p_i can be determined, i.e., p_i is the location from which the reward of $v_{\sigma_i} \in V$ can be collected. The first is in the insertion of the location by the ADDLOCATION procedure (Algorithm 2). The second place is the optimization of the waypoint locations P of the whole determined path in the LOCALSEARCH procedure (Algorithm 3). In both cases, we consider a relatively straightforward heuristic (originally utilized in the GSOA [5]) that works as follows.

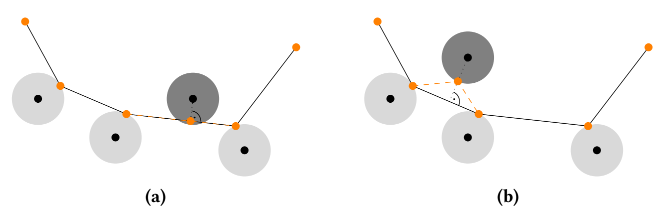


Figure 4: An example of waypoint location determination during the insertion of the location that corresponds to the dark gray disk.

For an insertion of the location v_{σ_i} into an existing path between two waypoint locations p_{j-1} and p_j, the corresponding waypoint locations p_i of v_{σ_i} is determined as the closest point to the segment (p_{j-1}, p_j) that is inside the disk with the radius ρ centered at v_{σ_i}. An example of the determined locations is shown in Fig. 4.

The heuristic determination of the position p_i for the insertion of v_{σ_i} is considered in the procedure ADDLOCATION in the `insertion` operator (Eq. (2)) to keep the prolongation as of Eq. (3) minimal.

In addition, the identical heuristic is utilized to shorten the length of the path $\mathcal{L}(P, \Sigma)$ in the local search phase by the `optimization` function (Line 6, Algorithm 3) that iterates several times over the path P' and tries to adjust the position of each location in the path. Based on empirical results, three iterations are observed to be a suitable trade-off between the solution quality and computational requirements because the optimization converges very fast. Three iterations are also used for the results. Thus, each visited p_j for $j \in \{2, \ldots, |P'| - 1\}$ is updated to p'_j using the heuristic for the segment defined by the neighboring waypoint locations p_{j-1} and p_{j+1} (see Fig. 4) if Eq. (6) holds.

$$\left\|p_{j-1}, p_j\right\| + \left\|p_j, p_{j+1}\right\| > \left\|p_{j-1}, p'_j\right\| + \left\|p'_j, p_{j+1}\right\| \qquad (6)$$

Although the proposed enhancement of the GRASP-SR to address the CEOP is relatively straightforward, the results indicate it provides competitive solutions to other existing methods for the CEOP. The empirical results are reported in the following section.

4 RESULTS

The proposed GRASP-based approach to the CEOP has been empirically evaluated using existing benchmark instances. In particular, the instances called Set 1, Set 2, and Set 3, proposed by Tsiligirides [21], two large sets Set 64 and Set 66 [3] with the 64 and 66 locations, respectively. Finally, we also include Set 130 [6], which further increases the number of locations to 130. For each particular set, the instance of the CEOP is defined by the travel budget T_{max} selected from a particular set of values and sensing range ϱ that is selected from the set $\varrho \in \{0.0, 0.5, 1.0, 1.5, 2.0\}$, where for $\varrho = 0.0$, the instance corresponds to a regular OP.

Two existing approaches to the CEOP are considered in the herein reported evaluation results. The first approach GSOA [6] uses unsupervised learning, and the second approach is a combinatorial heuristic based on the VNS [17] that is considered with eight samples per each disk neighborhood and with the termination condition of the 1000 iterations or 200 iterations without improvement.

The modified GRASP-SR to the CEOP (see Section 3) is considered in two variants. The first variant is denoted **GRASP** with the optimization of the waypoint location in the construction phase implemented in the addLocation procedure in the `insertion` operator Eq. (2). The second variant further includes the optimization of the found path in the local search procedure, according to Eq. (6), and it is denoted **GRASP**$_{\text{opt}}$.

All the evaluated algorithms have been implemented in C++ [25][1], and the reported results have been obtained within the same computational environment with the Intel Core i5-4460 CPU running at up to 3.2 GHz, but a single core of the processor has been utilized. The average real computational times t_{cpu} are reported in milliseconds and can be directly compared. Since all the algorithms are randomized, each particular instance is solved 20 times, and the solution quality is reported as the maximal sum of the collected rewards R among the performed trials.

Due to a relatively high number of instances (regarding T_{max} and ϱ), we report on aggregated results computed as the average value of the relative gap of the sum of the collected rewards R to the best-found solution R^* of the particular instance $G = R - R^*$, that is denoted $\overline{G}$ in Table 1. Depending on the sensing range ϱ, a particular value of T_{max} allows visitation of all the locations. Thus the reward would be the maximal reward possible that can bias the average value of the gap. Therefore, such instances for which at least two solution methods provide paths collecting all the rewards are not included in the reported aggregated results. Particular detailed results for the Set 64, Set 66, and Set 130 with the selected travel budgets T_{max} and sensing ranges ϱ are reported in Table 2 and Table 3, where the best-found solutions are highlighted in bold. An overview of the solution quality and computational requirements for the instances from Set 64 are visualized in Fig. 5.

Table 1: Aggregated results for CEOP Benchmark Instances

Instances	ϱ	GSOA $\overline{G}$ [%]	VNS $\overline{G}$ [%]	**GRASP** $\overline{G}$ [%]	**GRASP**$_{\text{opt}}$ $\overline{G}$ [%]
Sets 64, 66 [3]	0.5	5.55	1.87	1.83	**0.04**
	1.0	8.84	1.93	2.41	**0.08**
	1.5	11.88	4.33	2.80	**0.07**
	2.0	14.39	2.77	2.97	**0.50**
Sets 1, 2, 3 [21]	0.5	1.03	**0.19**	1.21	0.30
	1.0	0.33	**0.00**	0.72	0.66
	1.5	1.09	1.52	1.63	**0.84**
	2.0	0.75	1.46	0.67	**0.16**
Set 130 [6]	0.0	6.74	**0.77**	1.80	1.79
	1.0	8.29	**0.28**	5.42	2.04
	2.0	5.36	**0.01**	4.17	1.05

Based on the reported results, the proposed GRASP-based method for the CEOP provides solutions with the competitive solution quality to the solutions provided by both evaluated existing methods, the GSOA and VNS. In several cases, especially for relatively large instances Set 64 and Set 66, the proposed GRASP-based method provides the best results among the evaluated trials. The optimization in the local search phase significantly improves the solution quality at the cost of a bit increased computational requirements. However, the GRASP seems to be less or similarly demanding as the GSOA, while the solutions are better or the same. In few cases, VNS provides better results than the GRASP at the cost of significantly higher computational requirements because of the explicit discretization of the disk-shaped neighborhood, which is avoided by the heuristic determination of the waypoint location in the GRASP and GSOA.

Although the proposed method is based on the relatively simple and straightforward heuristic for continuous optimization of the addressed CEOP, the reported results support the proposed GRASP-based method is a suitable method with competitive computational requirements to the unsupervised learning of the GSOA. The quality of the found solutions is competitive to the computationally demanding VNS but noticeably better than the GSOA, specifically for the instances with relatively small travel budget T_{max}, where it is essential how the subset of locations is selected together with the high-quality solution of the sequencing part, where the GRASP outperforms the unsupervised learning.

[1]Source codes of the proposed GRASP algorithm are publicly available at https://github.com/comrob/ceop-grasp.

Table 2: Results for the CEOP instances of Set 64 and Set 66

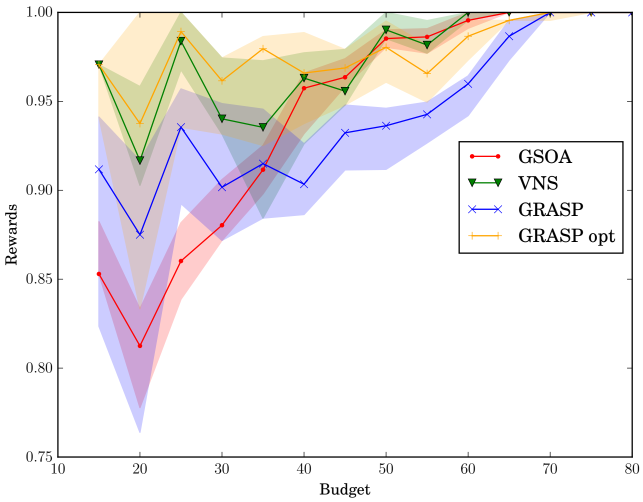
CEOP Instances	T_{max}	GSOA		VNS		**GRASP**		**GRASP$_{opt}$**	
		R	t_{cpu} [ms]	R	t_{cpu} [ms]	R	t_{cpu} [ms]	R	t_{cpu} [ms]
Set 64 ($\varrho = 0.5$)	15	186	22.2	198	2 848.3	198	4.2	**204**	8.5
	25	504	59.0	**558**	15 952.5	552	16.4	**558**	38.0
	35	822	84.1	858	30 078.5	846	39.2	**882**	72.0
Set 64 ($\varrho = 1.0$)	15	288	28.9	**300**	6 352.3	**300**	5.3	**300**	8.6
	25	672	69.2	**732**	24 500.6	720	13.8	726	38.2
	35	1116	108.5	**1152**	41 783.4	1122	20.1	1140	54.0
Set 64 ($\varrho = 1.5$)	15	372	36.6	324	7 446.4	402	6.1	**414**	10.1
	25	876	95.2	834	31 128.5	906	12.7	**912**	65.0
	35	**1344**	136.8	1284	47 363.6	1302	11.7	1308	57.0
Set 64 ($\varrho = 2.0$)	15	480	47.9	486	9 705.6	**534**	6.7	522	10.0
	25	1056	115.0	1032	38 251.8	1056	10.6	**1068**	36.9
	35	**1344**	134.3	**1344**	35 763.7	**1344**	9.1	**1344**	12.4
Set 66 ($\varrho = 0.5$)	20	260	36.5	**265**	8 933.1	260	4.5	**265**	6.7
	40	610	61.8	**730**	17 879.8	715	28.6	**730**	51.8
	80	1515	131.8	1525	64 843.9	1510	39.7	**1535**	69.5
Set 66 ($\varrho = 1.0$)	20	335	41.5	365	10 453.6	350	8.0	**380**	13.1
	40	875	70.2	980	26 533.5	965	30.1	**985**	60.4
	80	**1680**	124.0	**1680**	42 932.1	**1680**	9.8	**1680**	14.6
Set 66 ($\varrho = 1.5$)	20	415	49.1	435	12 270.6	455	8.7	**465**	45.6
	40	950	80.8	1055	31 819.7	1065	24.9	**1090**	111.3
	80	**1680**	126.8	**1680**	46 820.7	**1680**	4.7	**1680**	6.5
Set 66 ($\varrho = 2.0$)	20	520	54.1	**570**	17 281.0	555	10.6	**570**	77.5
	40	1195	97.6	**1315**	38 772.8	1255	17.8	1275	110.0
	80	**1680**	129.7	**1680**	49 093.5	**1680**	5.2	**1680**	6.8

Table 3: Results for the CEOP instances of Set 130

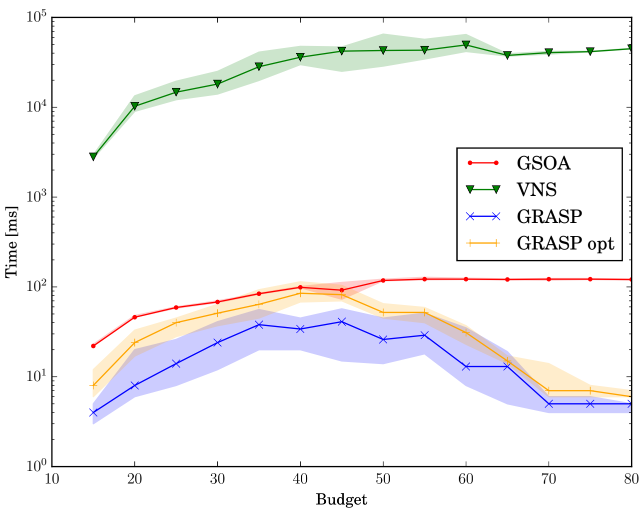
CEOP Instances	T_{max}	GSOA		VNS		**GRASP**		**GRASP$_{opt}$**	
		R	t_{cpu} [ms]	R	t_{cpu} [ms]	R	t_{cpu} [ms]	R	t_{cpu} [ms]
Set 130 ($\varrho = 0$)	50	**375**	30.4	**375**	753.4	**375**	11.7	**375**	11.4
	150	1210	145.3	**1369**	15 355.7	1296	270.9	1297	265.9
	250	2075	263.9	2218	24 273.8	2250	1 097.1	**2264**	1 073.2
Set 130 ($\varrho = 1$)	50	**462**	37.0	**462**	11 067.4	460	12.1	**462**	23.6
	150	1534	195.0	**1846**	175 414.1	1632	286.7	1777	473.8
	250	2802	367.4	2967	326 766.0	2876	470.1	**3003**	852.4
Set 130 ($\varrho = 2$)	50	543	41.5	**548**	10 500.0	521	15.3	545	29.5
	150	1774	217.8	**2115**	207 108.8	1875	203.9	2087	475.9
	250	3258	437.6	3296	398421.6	3233	274.1	**3297**	433.4

5 CONCLUSION

A novel Greedy Randomized Adaptive Search Procedure (GRASP) based approach to the Close Enough Orienteering Problem (CEOP) has been proposed and empirically evaluated using existing benchmark instances. The proposed GRASP method is based on the enhancements of the existing GRASP with Segment Remove (GRASP-SR) using a relatively simple and straightforward heuristic to determine the optimal position of the waypoint locations to collect reward within a non-zero sensing range, initially utilized in unsupervised learning of the GSOA. Although the proposed enhancements are straightforward and easy to implement, the developed GRASP-based algorithm for the CEOP outperforms the existing approaches to the CEOP in terms of the solution quality (in comparison to the GSOA) and computational requirements (in comparison to the VNS). Thus, the proposed approach is a vital method to address combinatorial routing problems that also include continuous optimization. In the addressed CEOP, the continuous optimization is based on exploiting the non-zero sensing range and determination of the suitable waypoint locations from which the rewards are collected. Thus more rewards can be collected because of the saved travel cost.



(a) Sum of the collected rewards – R



(b) Required computational time – t_{cpu}

Figure 5: Sum of the collected rewards R and the required computational time t_{cpu} of the evaluated algorithms in the instance of Set 64 with the sensing range $\varrho = 0.5$. The sum of the collected rewards is normalized to the highest value of the collected reward $R_{\max}$ for each particular instance of the CEOP found among all the evaluated methods. The solid curve in the plot represents the median, and the semi-transparent area represents 80% non-parametric confidence interval of the computed data.

The reported results support further research on employment of the GRASP in similar problems such as the Close Enough Traveling Salesman Problem (CETSP) but also other routing problems such as the curvature-constrained Dubins TSP and OP, where both existing approaches to the CEOP (the GSOA and VNS) have been already deployed. However, we also plan to exploit low computational requirements of the developed GRASP method to improve the heuristic determination of the waypoint locations by a more sophisticated local optimization to improve the quality of the determined solutions further.

REFERENCES

[1] Best, G., Faigl, J., and Fitch, R. Online planning for multi-robot active perception with self-organising maps. *Autonomous Robots 42*, 4 (2018), 715–738.

[2] Campos, V., Martí, R., Sánchez-Oro, J., and Duarte, A. Grasp with path relinking for the orienteering problem. *Journal of the Operational Research Society 65*, 12 (2014), 1800–1813.

[3] Chao, I.-M., Golden, B. L., and Wasil, E. A. A fast and effective heuristic for the orienteering problem. *European Journal of Operational Research 88*, 3 (1996), 475–489.

[4] Croes, G. A. A method for solving traveling-salesman problems. *Operations Research 6*, 6 (1958), 791–812.

[5] Faigl, J. GSOA: growing self-organizing array - unsupervised learning for the close-enough traveling salesman problem and other routing problems. *Neurocomputing 312* (2018), 120–134.

[6] Faigl, J. Data collection path planning with spatially correlated measurements using growing self-organizing array. *Applied Soft Computing 75* (2019), 130–147.

[7] Faigl, J. Unsupervised learning-based solution of the close enough dubins orienteering problem. *Neural Computing and Applications* (2019).

[8] Faigl, J., and Pěnička, R. On close enough orienteering problem with dubins vehicle. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2017), pp. 5646–5652.

[9] Faigl, J., Pěnička, R., and Best, G. Self-organizing map-based solution for the orienteering problem with neighborhoods. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (2016), pp. 1315–1321.

[10] Feillet, D., Dejax, P., and Gendreau, M. Traveling salesman problems with profits. *Transportation Science 39*, 2 (2005), 188–205.

[11] Feo, T. A., and Resende, M. G. C. Greedy randomized adaptive search procedures. *Journal of Global Optimization 6*, 2 (1995), 109–133.

[12] Golden, B. L., Levy, L., and Vohra, R. The orienteering problem. *Naval Research Logistics (NRL) 34*, 3 (1987), 307–318.

[13] Gunawan, A., Lau, H. C., and Vansteenwegen, P. Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research 255*, 2 (2016), 315–332.

[14] Hansen, P., and Mladenović, N. Variable neighborhood search: Principles and applications. *European Journal of Operational Research 130*, 3 (2001), 449–467.

[15] Keshtkaran, M., and Ziarati, K. A novel grasp solution approach for the orienteering problem. *Journal of Heuristics 22*, 5 (2016), 699–726.

[16] Pěnička, R., Faigl, J., and Saska, M. Variable neighborhood search for the set orienteering problem and its application to other orienteering problem variants. *European Journal of Operational Research 276*, 3 (2019), 816–825.

[17] Pěnička, R., Faigl, J., Váňa, P., and Saska, M. Dubins orienteering problem. *IEEE Robotics and Automation Letters 2*, 2 (2017), 1210–1217.

[18] Ramesh, R., and Brown, K. M. An efficient four-phase heuristic for the generalized orienteering problem. *Computers & Operations Research 18*, 2 (1991), 151–165.

[19] Sevkli, Z., and Sevilgen, F. E. Variable neighborhood search for the orienteering problem. In *21th International Symposium on Computer and Information Sciences (ISCIS)* (2006), pp. 134–143.

[20] Tsiligirides, T. Heuristic methods applied to orienteering. *The Journal of the Operational Research Society 35*, 9 (1984), 797–809.

[21] Tsiligirides, T. Heuristic methods applied to orienteering. *Journal of the Operational Research Society 35*, 9 (1984), 797–809.

[22] Vansteenwegen, P., Souffriau, W., Berghe, G. V., and Oudheusden, D. V. A guided local search metaheuristic for the team orienteering problem. *European Journal of Operational Research 196*, 1 (2009), 118–127.

[23] Vansteenwegen, P., Souffriau, W., and Oudheusden, D. V. The orienteering problem: A survey. *European Journal of Operational Research 209*, 1 (2011), 1–10.

[24] Vansteenwegen, P., and Van Oudheusden, D. The mobile tourist guide: An or opportunity. *OR Insight 20*, 3 (Jul 2007), 21–27.

[25] Štefaníková, P., Váňa, P., and Faigl, J. Greedy randomized adaptive search procedure for close enough orienteering problem. https://purl.org/comrob/sw, [cited on 2019-Dec-9].