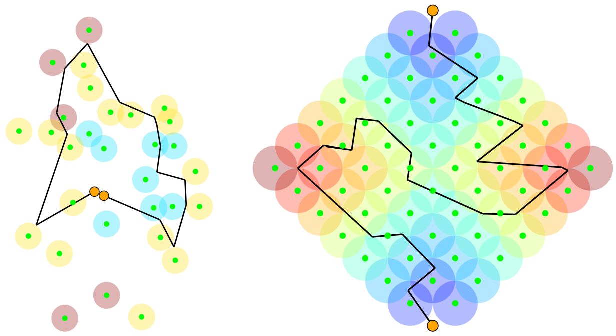
Hopfield Neural Network in Solution of the Close Enough Orienteering Problem

Jindřiška Deckerová, Jan Faigl

Czech Technical University in Prague, Technicka 2, 16627 Prague, Czech Republic,
{deckejin,faiglj}@fel.cvut.cz

In this paper, we report on the Hopfield Neural Network (HNN) for the Orienteering Problem (OP) that is generalized to solve instances of the Close Enough Orienteering Problem (CEOP). In the orienteering problems, we are searching for a limited budget tour to maximize collected rewards by visiting selected target locations. In the CEOP, it is allowed to collect the reward remotely within a non-zero communication range. Thus we can save travel costs by collecting rewards at suitable visiting locations of the disk-shaped neighborhoods of target locations. The proposed approach combines the HNN for the OP with the Second-Order Cone Programming (SOCP) that is employed to determine locally optimal locations of visits to the disk-shaped neighborhoods of the target locations. Regarding the reported evaluation results using standard benchmarks, the proposed SOCP-based approach provides solutions with the improved solution quality compared to the previous HNN-based method with discrete samples of the possible locations of visits.



(a) Set 1: $T_{max} = 46, \delta = 1, R = 200$ (b) Set 64: $T_{max} = 45, \delta = 1, R = 1320$

Figure 1: Example of the CEOP solutions, where the color of the disk-shaped neighborhoods indicates the value of reward (low rewards are in the blue and highly rewarding locations are in the red).

1 Introduction

The Orienteering Problem (OP) is a routing problem with profits inspired by the outdoor sport *Orienteering*. In a particular variant of *Orienteering*, the participants are given a set of locations, each associated with a reward. The participants aim to collect as many rewards as possible by visiting the defined locations within the given time budget. Similarly, the OP stands to maximize the sum of the collected rewards associated with the locations such that the tour cost does not exceed the given travel budget. The OP has been formally introduced in [1], although, the first approaches to solve the orienteering have been presented in [2], and since that, several approaches have been proposed [3–6].

The OP is a suitable problem formulation for data collection missions, where the utilized vehicles have a limited travel budget. Because visiting all locations is not feasible, the problem is to determine a subset of the most rewarding locations that can be visited within the given travel budget. Furthermore, we can exploit a non-zero communication range in a case where data can be retrieved from the locations remotely. Thus, we can save travel costs by collecting data distantly and utilize the travel budget to collect more rewards. Because the reward can be collected arbitrarily within a disk-shaped neighborhood of each target locations, such a variant of the OP is referred to as the Close Enough OP (CEOP) [7,8], albeit it can also be found as the OP with Neighborhoods (OPN) [9–11]. An example of the CEOP instances is depicted in Fig. 1.

In this paper, we propose a new approach to the solution of the CEOP based on the Hopfield Neural Network (HNN) for the OP presented in [12] that is combined with the convex optimization approach of the Second-Order Cone Programming (SOCP). The recurrent HNN is usually used in the data reconstruction tasks, although it has been adapted to various routing problems [13, 14]. The HNN for the CEOP is studied in [15], where an approach based on the discretization of the continuous neighborhoods is proposed, and the problem is solved as a variant of the Set OP [11, 16].

The drawback of the Set OP based approach is in the pre-determination of the points of visits to the neighborhoods of the target locations that are determined independently on the final tour. We propose a generalization of the HNN, where the locally optimal data collection points are determined using the SOCP. Based on the empirical evaluation using the existing benchmarks for the CEOP, the proposed approach exhibits improved solutions quality compared to the previous HNN approach based on the solution of the discrete Set OP. Besides, the proposed SOCP-based HNN is compared with the state-of-the-art Greedy Randomized Adaptive Search Procedure (GRASP) [8]. The proposed HNN-based method seems to provide solutions with the competitive solution quality to the GRASP, but it is more computationally demanding.

The rest of the paper is organized as follows. A brief overview of the existing approaches to the CEOP is presented in Section 2. The CEOP is formally defined in Section 3. The utilized baseline HNN for the OP is described in Section 4 and the proposed SOCP-based improvements are introduced in Section 5. The evaluation results are reported in Section 6. The concluding remarks are discussed in Section 7.

2 Related Work

The Orienteering Problem (OP) belongs to the class of routing problems with profits motivated by data collection missions. The OP can be considered a combination of the two NP-hard problems, the Traveling salesman problem in finding the most cost-efficient tour, and the Knapsack problem in determining the most rewarding subset of the target locations. Thus finding an optimal solution is computationally demanding, and therefore, heuristics have been proposed, such as the S-algorithm, D-algorithm, and route improvement heuristic in [2], the Center of gravity heuristic in [1], the Four-phase heuristic [3], the Five-step heuristic [4]. Besides, combinatorial meta-heuristics have been adopted to solve the OP, such as the Variable Neighborhood Search (VNS) [17] and the Greedy Randomized Adaptive Search (GRASP) [5]. Moreover, two sets of benchmark instances have been established based on the problems studied in [2,4]. In contrast to that, only a few approaches to the CEOP are reported in the literature: the Growing Self-Organizing Array (GSOA) [7], the VNS-based approach [11], and the GRASP-based method [8], that are overviewed in the rest of this section.

The GSOA [7, 18] is an unsupervised learning method based on the Self-Organizing Map (SOM) [6, 9, 10]. The GSOA stands for an array of nodes representing the requested route. It is iteratively adapted towards the target locations by determining the closest point of the route to a randomly selected target. The node is added to the array at the closest point position, and it is adapted with its neighboring nodes towards the target location. The GSOA addresses the continuous optimization (of determining locations of visits to the neighborhoods) during the insertion of the node into the array when a point from which the data can be obtained is determined using the communication range.

A similar approach to address the continuous neighborhoods as in the GSOA is utilized in the GRASP-based solution of the CEOP proposed in [8]. The GRASP consists of two steps: construction and local search. In the construction step, a route is constructed by iteratively adding new locations according to an insertion heuristic until the route is unchanged. Afterward, the route is improved in the local search step, where data collection points are iteratively refined to shorten the travel cost allowing insertion of new target locations into the final route. The VNS-based approach [11] addresses the continuous optimization of the CEOP by discretizing the disk-shaped neighborhoods into a finite set of locations and solving the problem as the Set OP [16].

All the methods are reported to provide solutions of high quality, although they have particular drawbacks. The GSOA is a fast constructive heuristic utilized in many routing problems; however, once it converges to a stable state, the solution does not further improve. On the other hand, the VNS provides better quality solutions than the GSOA, but it is more computationally demanding. So far, the GRASP provides the solutions to the CEOP of the best quality in the shorter computational time than the GSOA.

In the current work, we investigate the Hopfield Neural Network (HNN) for the OP [12] in the solution of the CEOP. The first attempt to solve the CEOP by the HNN is presented in [15], where the disk-shaped neighborhoods are discretized, and the problem is solved as a variant of the Set OP, albeit the problem is referred as the OPN in [15]. The network is represented as the three-dimensional matrix, where the third dimension represents discretized locations within the neighborhood. The objective of the network is to minimize a complex energy function that addresses the constraints of the CEOP. Once the network converges to a local minimum, a route is constructed from the network and further improved. Although the quality of the obtained solutions is relatively weak compared to the GSOA, VNS, and GRASP based heuristics, the expected advantage of the HNN is in the possible parallelization of the energy function computation and also in the potential to address sequence-dependent generalization of the routing problems. Therefore, we propose to address the solution quality of the HNN-based solutions of the CEOP by the deployment of the Second-Order Cone Programming (SOCP) in determining locally optimal distant locations of visits to the target locations. The HNN for the OP and the employed SOCP are thoroughly described in Section 4 and Section 5, respectively.

3 Problem Statement

The CEOP is a variant of the OP motivated by data collection missions, where the goal is to collect the most rewarding data from a set of target locations S, starting and ending at the predefined targets, while the traveled tour remains within the given travel budget $\mathrm{T}_{\max}$. The targets S are rewarded by a score r_i depending on the importance of the particular target, and data can be remotely collected within δ distance from the particular target location; thus, forming a disk-shaped neighborhood with the radius δ. The CEOP can be defined as follows.

Let $S = \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n\}$ be a set of n target locations, where $\boldsymbol{s}_i \in \mathbb{R}^2$, and δ be a communication radius associated with every location $\boldsymbol{s}_i \in S$, except the start and end locations that are denoted $\boldsymbol{s}_1$ and $\boldsymbol{s}_n$, respectively, since it is requested to visit precisely these locations. Each location is associated with the reward $r_i > 0$, except $\boldsymbol{s}_1$ and $\boldsymbol{s}_n$ that are assigned with zero reward $r_1 = r_n = 0$. The CEOP stands to determine a tour maximizing the sum of the collected rewards R, while the tour length does not exceed the given travel budget $\mathrm{T}_{\max}$. Since the tour length is limited, only a subset of k locations $S_k \subseteq S$ can be visited. The tour visiting the subset S_k of the targets can be described as a sequence of visits $\Sigma = (\sigma_1, \ldots, \sigma_k)$, where $1 \leq \sigma_i \leq n, \forall i \neq j : \sigma_i \neq \sigma_j$, and $\sigma_1 = 1, \sigma_k = n$, together with a set of waypoint locations $P = (\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k)$, $\boldsymbol{p}_i \in \mathbb{R}^2$, where each $\boldsymbol{p}_i$ is within the communication radius δ of the corresponding location $\boldsymbol{s}_{\sigma_i}$, i.e., $\|\boldsymbol{p}_i - \boldsymbol{s}_{\sigma_i}\| \leq \delta$. The CEOP is defined as the optimization Problem 1, where $\left\|\boldsymbol{p}_i - \boldsymbol{p}_j\right\|$ denotes the Euclidean distance between the locations $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$.

Problem 1 Close Enough Orienteering Problem (CEOP)**.**

$$\begin{array}{ll} \text{maximize}_{k,\Sigma,P} & R = \sum_{i=1}^{k} r_{\sigma_i} \\ \text{s.t.} & \sum_{i=1}^{k-1} \left\|\boldsymbol{p}_i - \boldsymbol{p}_{i+1}\right\| \leq \mathrm{T}_{\max} \\ & 2 \leq k \leq n \\ & \|\boldsymbol{p}_i - \boldsymbol{s}_{\sigma_i}\| \leq \delta \\ & \boldsymbol{p}_1 = \boldsymbol{s}_1\,, \boldsymbol{p}_k = \boldsymbol{s}_n \end{array}$$

4 Hopfield Neural Network for OP

The herein proposed solution to the CEOP is based on the Hopfield Neural Network (HNN) for the OP presented in [12] that is briefly described in this section to keep the paper self-contained. The HNN consists of three main parts: the problem representation, the energy function, and the improvement heuristics, such as 2-Opt [19], the insertion and deletion heuristics. The HNN is conceptualized as a state matrix $\Phi \in \mathbb{R}^{n,n+1}$, where each cell denotes the continuous activation level $\Phi_{i,j} \in [0,1]$. Each activation level $\Phi_{i,j}$ is updated by the sigmoid activation function

$$\Phi_{i,j} = \frac{1}{1 + e^{-\alpha}} \tag{1}$$

with

$$\alpha = \ln(\Phi_{i,j}) - \ln(1 - \Phi_{i,j}) - \frac{\partial E}{\partial \Phi_{i,j}} \Delta t \tag{2}$$

where E is the energy function and Δt is the time step.

Each activation level $\Phi_{i,j}$ is referred to as a *state*, and it denotes that the location $\boldsymbol{s}_i$ is visited at the j-th position of the tour. If the value of $\Phi_{i,j}$ is close to 1, then it is likely that the location $\boldsymbol{s}_i$ is selected to be visited at the j-th position. The start and end locations are requested to visit; hence the states $\Phi_{1,1}$ and $\Phi_{n,n}$ are forced to 1. The last column of $\Phi_{i,n+1}$ is used to calculate the reward of the tour. Thus, if the location $\boldsymbol{s}_i$ is to be included in the route, the last column is set to 1, $\Phi_{i,n+1} = 1$, otherwise $\Phi_{i,n+1} = 0$.

The essential part of the HNN is the complex energy function E for which the second derivative w.r.t. the current state is used as the weights of the network. The aim of the network is to minimize the energy function designed to meet the constraints of the OP. The energy function for the OP [12] is

$$E = \frac{a}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{\substack{h=1 \\ h \neq i}}^{n} \Phi_{i,j} \Phi_{h,j} \tag{3}$$

$$+ \frac{b}{2} \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \Phi_{i,j} - n \right)^2 \tag{4}$$

$$+ \frac{c}{2} \Gamma\left(\sum_{i=1}^{n} \sum_{j=1}^{n-1} \sum_{h=1}^{n} \|\boldsymbol{s}_i - \boldsymbol{s}_h\| \, \Phi_{i,j} \Phi_{h,j+1} - \mathrm{T}_{\max} \right) \tag{5}$$

$$+ d \left(2 - \Phi_{1,1} - \Phi_{n,n}\right) \tag{6}$$

$$+ e \sum_{i=1}^{n} \left(\Phi_{i,n+1} \left(1 - \sum_{j=1}^{n} \Phi_{i,j}\right) \right) \tag{7}$$

$$- f \sum_{i=1}^{n} r_i \Phi_{i,n+1}, \tag{8}$$

where a, b, c, d, e, f are parameters of the network and

$$\Gamma(x) = \begin{cases} x^2 & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}. \tag{9}$$

The energy function E consists of six terms; each term is associated with the multiplication parameter and ensures a given constraint of the OP is met. The first term (3) associated with the parameter a penalizes multiple activated states in the same column to ensure only one location can be visited at the time. The second term (4) associated with the parameter b ensures the maximal number of activated states is less or equal to n. If the number is less than n, the activated states are consecutively repeated. The third term (5) penalizes routes that their traveled length exceeds the travel budget $\mathrm{T}_{\max}$. The term (6) associated with the parameter d ensures that route starts and ends at the start location $\boldsymbol{s}_1$ and end location $\boldsymbol{s}_n$, respectively. The next term (7) associated with the parameter e sets $\Phi_{i,n+1} = 1$ if the location $\boldsymbol{s}_i$ is included in the route. Finally, the last term (8) is used to maximize the sum of collected rewards.

Having the network representation and the energy function, the states of Φ are iteratively updated according to the activation function (1). A brief overview of how the representation and energy function is utilized in the solution for the OP follows.

First, the state matrix is initialized to random values, and parameters are initialized to predefined values. Then, a random row of Φ is selected, and all states of the row are updated according to the activation function (1). The selection and update are repeated until a local minimum is reached. The local minima occur when for any state $\Phi_{i,j}$ the value of $\left| -\frac{\partial E}{\partial \Phi_{i,j}} \Delta t \right|$ is less than a predefined threshold ϑ three times in succession. When a local minimum is reached, a route is constructed from the state matrix by selecting a state with the largest value for each column. Afterward, the local improvement by the 2-Opt [19] is applied to the route. If the traveled route does not violate the travel budget $\mathrm{T}_{\max}$, the parameter f of the network is increased; otherwise, f is decreased to satisfy the length constraint. The locally improved route is further tweaked by utilizing the cheapest insertion heuristic as in Definition 1 and the deletion heuristic as in Definition 2, and thus more locations can be visited and more rewards can be collected.

Definition 1. *Location $\boldsymbol{s}_{insert}$ to be inserted into the route at position j.*

$$\boldsymbol{s}_{insert}, j = \underset{\boldsymbol{s}_i \in S \setminus S_k, \boldsymbol{s}_j \in S_k}{\operatorname{argmax}} \frac{\|\boldsymbol{s}_i - \boldsymbol{s}_j\| + \|\boldsymbol{s}_i - \boldsymbol{s}_{j+1}\| - \|\boldsymbol{s}_j - \boldsymbol{s}_{j+1}\|}{r_i}.$$

Definition 2. *Location $\boldsymbol{s}_{remove}$ to be removed from the route*

$$\boldsymbol{s}_{remove} = \underset{\boldsymbol{s}_i \in S_k}{\operatorname{argmax}} \frac{\|\boldsymbol{s}_i - \boldsymbol{s}_{i+1}\|}{r_i}.$$

The state matrix is then adjusted according to the improved route, and the process of finding a local minimum of the adjusted network is repeated for the predefined number of repetitions. Then, the parameters and the state matrix are reinitialized, and the process starts again. The whole process is repeated for the predefined number of iterations.

In [12], the authors state that the promise of the HNN is in combination with the traditional heuristics; hence we follow the combination of the HNN and heuristics in the solution of the CEOP. The proposed extensions of the HNN are described in the following Section 5.

5 Proposed Method

The essential modification of the HNN to solve the CEOP is the term (5) of the energy function associated with the parameter c. The original term penalizes routes that exceed the travel budget and considers the Euclidean distance between two target locations. Since we aim to solve the CEOP, the disk-shaped neighborhoods need to be considered in (5), and the proposed form of (5) is highlighted in the modified energy function:

$$E = \frac{a}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{\substack{h=1 \\ h \neq i}}^{n} \Phi_{i,j} \Phi_{h,j} + \frac{b}{2} \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \Phi_{i,j} - n \right)^2 \quad (10)$$

$$+ \frac{c}{2} \Gamma \left(\sum_{i=1}^{n} \sum_{j=1}^{n-1} \sum_{k=1}^{n} \sum_{h=1}^{n} D(\boldsymbol{s}_k, \boldsymbol{s}_i, \boldsymbol{s}_j) \Phi_{k,j-1} \Phi_{i,j} \Phi_{h,j+1} - \mathrm{T}_{\max} \right) \quad (11)$$

$$+ d\,(2 - \Phi_{1,1} - \Phi_{n,n}) + e \sum_{i=1}^{n} \left(\Phi_{i,n+1}\,(1 - \sum_{j=1}^{n} \Phi_{i,j}) \right) \quad (12)$$

$$- f \sum_{i=1}^{n} r_i\, \Phi_{i,n+1}. \quad (13)$$

The function $D(\boldsymbol{s}_k, \boldsymbol{s}_i, \boldsymbol{s}_j)$ denotes the estimate of the minimal distance between three locations $\boldsymbol{s}_k, \boldsymbol{s}_i, \boldsymbol{s}_j$ using a determined waypoint location $\boldsymbol{p}_i$ for the location $\boldsymbol{s}_i$ as depicted in Fig. 2. We utilize the exact locations $\boldsymbol{s}_j$ and $\boldsymbol{s}_k$ instead of the waypoint locations $\boldsymbol{p}_j$ and $\boldsymbol{p}_k$, since the HNN is robust, the effect of the term (11) in the energy function can be influenced by the parameter c. Furthermore, the estimate between three exact locations can be precomputed, and thus the computational demands are lower than using the waypoint locations.

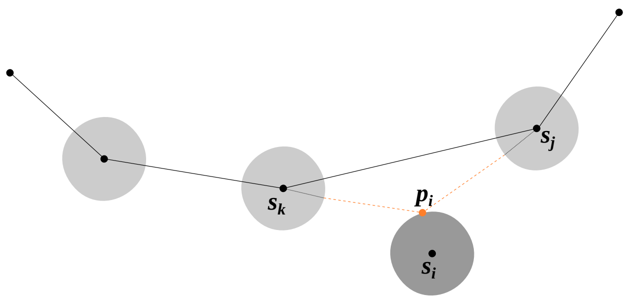


Figure 2: Estimation of the distance between three locations $\boldsymbol{s}_k, \boldsymbol{s}_i, \boldsymbol{s}_j$ using the waypoint location $\boldsymbol{p}_i$ of the location $\boldsymbol{s}_i$ (dark gray).

The value of the function D can be determined by various approaches, such as a geometric approach, or an approach based on the solution of the Second-Order Cone Programming (SOCP). In the geometric approach, we can consider already established waypoint locations instead of the target locations. Then, the value function becomes $D(\boldsymbol{p}_k, \boldsymbol{p}_i, \boldsymbol{p}_j)$, and $\boldsymbol{p}_i$ can be determined as the closest point to the line segment $(\boldsymbol{p}_k, \boldsymbol{p}_j)$. The geometric approach may seem conceptually suitable; however, it does not scale to more than three locations. Since we aim to utilize the studied HNN in the sequence-dependent problems with various lengths of the sequences in the future, we utilize a more general formulation of the SOCP. Thus the value of the function D together with the partial waypoint location is determined by the solution of the corresponding SOCP that is a convex optimization Problem 2 with affine constraints solved optimally by an optimization solver.

Problem 2 Second-Order Cone Programming (SOCP)**.**

$$\min \sum_{i=0}^{m-1} \boldsymbol{f}_i \quad (14)$$

s.t.

$$\boldsymbol{f}_i^2 \geq \boldsymbol{w}_i^T \cdot \boldsymbol{w}_i \qquad \forall i = 0, \ldots, m \quad (15)$$

$$\boldsymbol{w}_i = \boldsymbol{x}_{i+1} - \boldsymbol{x}_i \qquad \forall i = 0, \ldots, m-1 \quad (16)$$

$$\|\boldsymbol{x}_i - \boldsymbol{s}_i\| \leq \delta^2 \qquad \forall i = 0, \ldots, m \quad (17)$$

$$\boldsymbol{x}_1 = \boldsymbol{s}_1, \boldsymbol{x}_m = \boldsymbol{s}_m \quad (18)$$

$$\boldsymbol{x}_i \in \mathbb{R}^2 \qquad \forall i = 0, \ldots, m \quad (19)$$

Problem 2 stands to determine a set of m waypoint locations $\boldsymbol{x}_i$ such that the tour formed by the locations is of the minimal length. The objective function of Problem 2 is a linear function that stands to minimize the variable $\boldsymbol{f}$ (14), while the constraints represented by Equations (15) to (18) are met. The optimization problem consists of three variables $\boldsymbol{f} \in \mathbb{R}^m$, $\boldsymbol{w} \in \mathbb{R}^{m,2}$, $\boldsymbol{x} \in \mathbb{R}^{m,2}$ represented as vectors, where the variable $\boldsymbol{x}$ represents the set of waypoint locations $\boldsymbol{x}_i \in \mathbb{R}^2$. The variable $\boldsymbol{w}$ is an auxiliary variable that denotes the difference in coordinates between two consecutive waypoint variables (16), and it is used to calculate the length of the line segment between two waypoint locations (15). (17) ensures that each waypoint location $\boldsymbol{x}_i$ is within the given communication radius δ from the respective location $\boldsymbol{s}_i$. (18) is employed to ensure the start and end waypoint locations correspond to the start and end locations, respectively.

Besides, the solution of the SOCP is also utilized in the improvement of the route after the application of the 2-Opt heuristic [19]. The proposed HNN-SOCP solver to the CEOP is overviewed in Algorithm 1.

Algorithm 1: HNN-SOCP for the CEOP

Input: $S = \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n\}$ – a set of target locations
Input: I, R – the number of iterations and repetitions
Input: $\mathrm{T}_{\max}$ – the travel budget
Parameters : $a = 1$, $b = 1$, $c = 20$, $d = 1$, $e = 20$, $f = 15$, $\Delta\vartheta = 2$, $\Delta t = 0.0001$
Output: (Σ, P) – a sequence of visits Σ to the subset of the target locations S_k with the corresponding waypoint locations P

- $\Phi \leftarrow$ init_network(S)
- **foreach** i in I **do**
  - $\Phi \leftarrow$ reset_network()
  - $a, b, c, d, e, f, \Delta\vartheta, \Delta t \leftarrow$ reset_params()
  - **foreach** r in R **do**
    - **while** *local optima is not reached* **do**
      - $L \leftarrow$ get_random_level()
      - $\Phi_{*,L} = \frac{1}{1+e^{\ln(\Phi_{*,L}) - \ln(1-\Phi_{*,L}) - \frac{\partial E}{\partial \Phi_{*,L}}\Delta t}}$
    - $(\Sigma', P') \leftarrow$ construct_route(Φ)
    - $(\Sigma', P') \leftarrow$ 2-Opt(Σ', P')
    - $(\Sigma', P') \leftarrow$ SOCP-Opt(Σ', P')
    - **if** $\mathscr{L}(\Sigma', P') \leq \mathrm{T}_{\max}$ **then**
      - $f \leftarrow f - 1$
    - **else**
      - $f \leftarrow f + 1$
    - **while** *no. improvements is not reached* **do**
      - **if** $\mathscr{L}(\Sigma', P') < \mathrm{T}_{\max}$ **then**
        - $(\Sigma', P') \leftarrow$ insertion(Σ', P')
      - **else**
        - $(\Sigma', P') \leftarrow$ deletion(Σ', P')
    - **if** $\mathscr{L}(\Sigma', P') \leq T_{max}$ `and` $\mathscr{R}(\Sigma', P') > \mathscr{R}_{best}$ **then**
      - $(\Sigma, P) \leftarrow (\Sigma', P')$
      - $\mathscr{R}_{best} \leftarrow \mathscr{R}(\Sigma', P')$
    - $\Phi \leftarrow$ adjust_network(Σ', P')
- **return** (Σ, P)

6 Results

The proposed HNN with the SOCP-based modifications denoted as the HNN-SOCP has been empirically evaluated and compared with the Set OP based HNN [15] (further referred as the HNN) to observe how the continuous optimization affects the performance of the HNN-based solution of the CEOP. In addition, the GRASP method [8] is also considered in the herein reported evaluation study to show the performance of the HNN-based approaches in comparison to the currently best (to the best of the authors' knowledge) performing solver to the CEOP.

The approaches are evaluated on five datasets: Set 64, and Set 66 of Chao sets [4] and Set 1, Set 2, and Set 3 of Tsiligirides sets [2]. Chao sets contain 40 problem scenarios with budgets ranging from 5 to 130, and Tsiligrides sets contain 49 problem scenarios with budgets varying from 5 to 110. For each problem instances, the communication radius δ is selected from $\delta \in \{0.5, 1.0, 1.5, 2.0\}$. Due to the excessive number of problem instances, detailed results are reported only for five selected instances with the particular budget $\mathrm{T}_{\max}$. The selected instances are with budgets slightly above the median value of budget ranges for a particular scenario because low budgets are quite constraining, and high budgets usually allow covering all locations.

The evaluated HNN-SOCP is implemented in `C++`[1] using `CPLEX` solver [20] for the solution of the SOCP and the computational environment is a single-core of the Intel Core i5-4460 CPU running at 3.2 GHz. The results for the HNN are adopted from [15] and have been obtained by a different computational environment. Therefore, the computational times of the HNN [15] are adjusted by the factor 0.81 according to [21] to take into account different environments.

Since all the algorithms are randomized, the evaluation is performed among several trials, and results are reported using two performance indicators, %PDB and %PDM, and the maximal sum of collected rewards R obtained among the performed trials. The %PDB indicates the quality of solutions measured as the percentage deviation of the best solution $\mathscr{R}_{\text{best}}$ obtained among performed trials to the reference solution $\mathscr{R}_{\text{ref}}$, i.e., $\%\text{PDB} = \frac{\mathscr{R}_{\text{best}} - \mathscr{R}_{\text{ref}}}{\mathscr{R}_{\text{ref}}} \cdot 100\%$. The %PDM measures the robustness of the found solutions determined as the percentage deviation of the mean of the sum of collected rewards $\mathscr{R}_{\text{avg}}$ over the trials to the reference solution $\mathscr{R}_{\text{ref}}$, and it is computed as $\%\text{PDM} = \frac{\mathscr{R}_{\text{avg}} - \mathscr{R}_{\text{ref}}}{\mathscr{R}_{\text{ref}}} \cdot 100\%$. The reference value $\mathscr{R}_{\text{ref}}$ is selected as the best solution found among all methods and performed trials. Due to the high number of instances, the reported results in Table 1 are aggregated values, where $\overline{\%\text{PDB}}$ and $\overline{\%\text{PDM}}$ denote the average value of %PDB and %PDM for the particular scenario with all selected travel budgets.

The particular methods have been parameterized as fol-

[1] Source codes of the proposed HNN-SOCP are publicly available at https://github.com/comrob/ceop-hnn-socp.

Table 1: Aggregated performance indicators for the selected CEOP instances.

Problem		GRASP [8]		HNN [15]		**HNN-SOCP**	
		%PDB	%PDM	%PDB	%PDM	%PDB	%PDM
Set 64	$\delta = 0.5$	**0.00**	**2.53**	10.53	12.84	10.53	17.58
	$\delta = 1.0$	**0.00**	1.64	5.38	10.63	**0.00**	**1.03**
	$\delta = 1.5$	**0.00**	0.18	6.25	6.83	**0.00**	**0.00**
	$\delta = 2.0$	**0.00**	**0.00**	2.68	4.24	**0.00**	**0.00**
Set 66	$\delta = 0.5$	**0.00**	**2.51**	13.10	15.50	19.65	23.89
	$\delta = 1.0$	**0.00**	**2.82**	16.50	21.17	5.50	9.58
	$\delta = 1.5$	**0.00**	**2.52**	16.06	20.91	1.21	4.82
	$\delta = 2.0$	**0.00**	**0.40**	12.50	16.18	**0.00**	0.60
Set 1	$\delta = 0.5$	**0.00**	**2.07**	9.76	15.85	4.88	6.10
	$\delta = 1.0$	**0.00**	**1.09**	10.87	13.70	2.17	2.39
	$\delta = 1.5$	**0.00**	**2.21**	13.46	17.69	3.85	5.77
	$\delta = 2.0$	3.51	5.44	17.54	19.82	**0.00**	**0.00**
Set 2	$\delta = 0.5$	**0.00**	**1.11**	4.44	14.78	2.22	6.22
	$\delta = 1.0$	**0.00**	**0.22**	4.44	5.78	**0.00**	5.33
	$\delta = 1.5$	**0.00**	**0.22**	4.44	4.44	**0.00**	1.78
	$\delta = 2.0$	**0.00**	**0.00**	4.44	4.44	**0.00**	3.67
Set 3	$\delta = 0.5$	**0.00**	**1.64**	1.72	6.21	1.72	3.97
	$\delta = 1.0$	**0.00**	**0.63**	4.76	6.83	**0.00**	2.22
	$\delta = 1.5$	**0.00**	**1.34**	7.46	8.81	1.49	1.79
	$\delta = 2.0$	**0.00**	**1.53**	9.72	11.81	1.39	2.64

Table 2: Results for the selected CEOP instances.

Problem		T_{max}	GRASP* [8]		HNN [15]		**HNN-SOCP**	
			R	T [s]	*R*	T [s]	*R*	T [s]
Set 64	$\delta = 0.5$	45	**1140**	**0.062**	1020	157.486	1020	163.699
	$\delta = 1.0$	45	**1338**	**0.015**	1266	211.228	**1338**	181.535
	$\delta = 1.5$	45	**1344**	**0.005**	1260	260.347	**1344**	191.205
	$\delta = 2.0$	45	**1344**	**0.004**	1308	292.635	**1344**	188.050
Set 66	$\delta = 0.5$	60	**1145**	**0.047**	995	150.014	920	200.655
	$\delta = 1.0$	60	**1545**	**0.032**	1290	194.808	1460	217.480
	$\delta = 1.5$	60	**1650**	**0.016**	1385	219.205	1630	204.701
	$\delta = 2.0$	60	**1680**	**0.004**	1470	243.154	**1680**	203.833
Set 1	$\delta = 0.5$	46	**205**	**0.003**	185	8.330	195	11.323
	$\delta = 1.0$	46	**230**	**0.003**	205	11.519	225	9.221
	$\delta = 1.5$	46	**260**	**0.003**	225	15.039	250	8.484
	$\delta = 2.0$	46	275	**0.001**	235	17.531	**285**	8.475
Set 2	$\delta = 0.5$	38	**450**	**0.000**	430	4.640	440	5.566
	$\delta = 1.0$	38	**450**	**0.000**	430	5.532	**450**	3.701
	$\delta = 1.5$	38	**450**	**0.000**	430	12.482	**450**	3.643
	$\delta = 2.0$	38	**450**	**0.000**	430	45.091	**450**	3.662
Set 3	$\delta = 0.5$	50	**580**	**0.003**	570	11.885	570	14.069
	$\delta = 1.0$	50	**630**	**0.001**	600	14.156	**630**	13.086
	$\delta = 1.5$	50	**670**	**0.002**	620	17.079	660	12.400
	$\delta = 2.0$	50	**720**	**0.002**	650	18.419	710	12.144

*Solutions of Set 2 are reported with T = 0 [s] in [8], hence the same results are reported here. They are practically found in less than 1 ms.

lows. The GRASP has been run for 20 trials. The HNN is executed for 10 trials for 2 iterations and 10 repetitions with the parameters set to $a = 1$, $b = 1$, $c = 10$, $d = 20$, $e = 10$, $f = 15$, the local minimum threshold $\vartheta = 2$, the time step $\Delta t = 0.0001$, and the number of samples is $m = 10$. The proposed HNN-SOCP is executed for 10 iterations and 20 repetitions with a slightly adjusted parameters set to $a = 1$, $b = 1$, $c = 20$, $d = 1$, $e = 20$, $f = 15$, the local minimum threshold $\vartheta = 2$, and the time step $\Delta t = 0.0001$. The values of the function D are precomputed as a distance matrix, and the computation time is not included in the reported results since the matrix is precomputed only once for each scenario regardless of the travel budget.

The aggregated results are reported in Table 1 and results for the selected instances in Table 2. Overview of the results is presented in Fig. 3 and Fig. 4. The reported results indicate that the proposed HNN-SOPC provides competitive results in almost half of the presented problems, and in one case, the HNN-SOCP even outperforms the GRASP. However, the computational demands of the HNN-SOCP are significantly higher than of the GRASP, as depicted in Fig. 4. Overall, the proposed HNN-SOCP provides higher quality solutions for the problems with larger radii since the neighborhoods overlap, and we can collect rewards from multiple locations by one visit.

The SOCP-based modifications overall improve the performance of the HNN-based approach in comparison to the original HNN to the CEOP [15] based on the explicit discretization of the disk-shaped neighborhoods. However, the HNN-SOCP struggles to address problems with small communication radii. The computational times of both HNN-based approaches are similar, although the HNN-SOCP is run with five times more iterations and two times more repetitions, and thus more solutions are processed. It is because the HNN-SOCP has less complex problem representation, and the network converges faster.

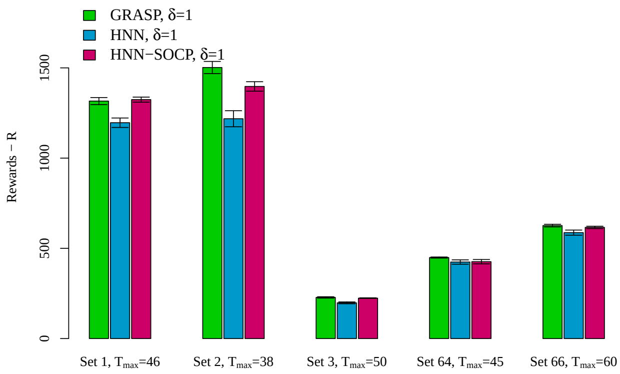


Figure 3: Average sum of collected rewards with standard deviations visualized as the error bars.

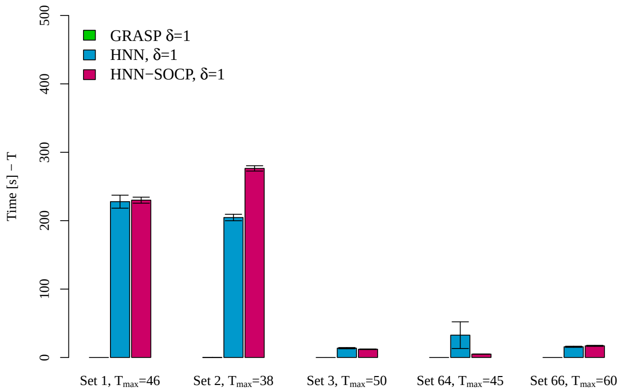


Figure 4: Average computational times of the respective algorithms with standard deviations visualized as the error bars.

7 Conclusion

In this paper, we report the study of solving the Close Enough Orienteering Problem (CEOP) by the Hopfield Neural Network (HNN). The original HNN for the OP has been combined with the convex optimization approach of the Second-Order Cone Programming (SOCP) to tackle the continuous neighborhoods of the CEOP. The proposed HNN-SOCP has been empirically evaluated and compared with the GRASP-based approach and the former HNN-based approach with the explicit discretization of the disk-shaped neighborhoods utilized as the baseline method. Although the HNN-SOCP does not outperform the GRASP-based solver, it significantly improves the existing HNN approaches to the CEOP in terms of the solution quality that becomes competitive to the GRASP.

In our future work, we aim to study the influence of the HNN parameters on the performance since the parameters are an essential part of the HNN. Besides, we also aim to utilize the HNN, particularly the possibility of the parallelization of the energy function, in the solution of the multi-vehicle and sequence-dependent routing problems.

Acknowledgments – This work was supported by the Czech Science Foundation (GAČR) under research project No. 19-20238S.

References

[1] B. L. Golden, L. Levy, and R. Vohra, “The orienteering problem,” *Naval Research Logistics (NRL)*, vol. 34, no. 3, pp. 307–318, 1987.

[2] T. Tsiligirides, “Heuristic methods applied to orienteering,” *Journal of the Operational Research Society*, vol. 35, no. 9, pp. 797–809, 1984.

[3] R. Ramesh and K. M. Brown, “An efficient four-phase heuristic for the generalized orienteering problem,” *Computers & Operations Research*, vol. 18, no. 2, pp. 151–165, 1991.

[4] I.-M. Chao, B. L. Golden, and E. A. Wasil, “A fast and effective heuristic for the orienteering problem,” *European journal of operational research*, vol. 88, no. 3, pp. 475–489, 1996.

[5] V. Campos, R. Martí, J. Sánchez-Oro, and A. Duarte, “Grasp with path relinking for the orienteering problem,” *Journal of the Operational Research Society*, vol. 65, no. 12, pp. 1800–1813, 2014.

[6] J. Faigl, “On self-organizing maps for orienteering problems,” in *2017 international joint conference on neural networks (IJCNN)*. IEEE, 2017, pp. 2611–2620.

[7] ——, “Data collection path planning with spatially correlated measurements using growing self-organizing array,” *Applied Soft Computing*, vol. 75, pp. 130–147, 2019.

[8] P. Štefaníková, P. Váňa, and J. Faigl, “Greedy randomized adaptive search procedure for close enough orienteering problem,” in *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, 2020, pp. 808–814.

[9] J. Faigl, R. Pěnička, and G. Best, “Self-organizing map-based solution for the orienteering problem with neighborhoods,” in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2016, pp. 001 315–001 321.

[10] G. Best, J. Faigl, and R. Fitch, “Online planning for multi-robot active perception with self-organising maps,” *Autonomous Robots*, vol. 42, no. 4, pp. 715–738, 2018.

[11] R. Pěnička, J. Faigl, and M. Saska, “Variable neighborhood search for the set orienteering problem and its application to other orienteering problem variants,” *European Journal of Operational Research*, vol. 276, no. 3, pp. 816–825, 2019.

[12] Q. Wang, X. Sun, B. L. Golden, and J. Jia, “Using artificial neural networks to solve the orienteering problem,” *Annals of Operations Research*, vol. 61, no. 1, pp. 111–120, 1995.

[13] J.-Y. Potvin, “State-of-the-art survey—the traveling salesman problem: A neural network perspective,” *ORSA Journal on Computing*, vol. 5, no. 4, pp. 328–348, 1993.

[14] S. Kashmiri, “The travelling salesman problem and the hopfield neural network,” in *IEEE Proceedings of the SOUTHEASTCON ’91*, 1991, pp. 940–943 vol.2.

[15] J. Deckerová, “Artificial neural networks in solution of the orienteering problems,” B.S. thesis, Czech Technical University in Prague, 2018.

[16] C. Archetti, F. Carrabs, and R. Cerulli, “The set orienteering problem,” *European Journal of Operational Research*, vol. 267, no. 1, pp. 264–272, 2018.

[17] Z. Sevkli and F. E. Sevilgen, “Variable neighborhood search for the orienteering problem,” in *International Symposium on Computer and Information Sciences*. Springer, 2006, pp. 134–143.

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

[19] G. A. Croes, “A method for solving traveling-salesman problems,” *Operations research*, vol. 6, no. 6, pp. 791–812, 1958.

[20] “CPLEX,” [cited on 2020-Jun-27]. [Online]. Available: https://www.ibm.com/products/ilog-cplex-optimization-studio

[21] “Passmark software, intel core i5-5200u @ 2.20ghz vs intel core i5-4460 @ 3.20ghz,” [cited on 2020-Jun-27]. [Online]. Available: https://www.cpubenchmark.net/compare/Intel-i5-5200U-vs-Intel-i5-4460/2440vs2230