2015 IEEE Symposium Series on Computational Intelligence

Comparing a Weiszfeld's-based Procedure and (1+1)-ES for Solving the Planar Single-Facility Location–Routing Problem

Pepijn van Heiningen
LIACS, Leiden University
Niels Bohrweg 1, 2333 CA
Leiden, The Netherlands
pepijnvanheiningen@hetnet.nl

Edgar Reehuis
DENC Netherlands B.V.
Slochterenlaan 12, 1405 AM
Bussum, The Netherlands
ereehuis@denc.nl

Thomas Bäck
Natural Computing Group
LIACS, Leiden University
Niels Bohrweg 1, 2333 CA
Leiden, The Netherlands
t.h.w.baeck@liacs.leidenuniv.nl

***Abstract*—We compare two iterative methods for solving the Planar Single-Facility Location–Routing Problem (PSFLRP), involving optimizing the continuous-valued location of a single depot by iteratively solving instances of the Vehicle Routing Problem (VRP). An Ant Colony Optimization (ACO) algorithm is used for solving the routing problem, a procedure using Weiszfeld's algorithm and a simple Evolution Strategy (ES) are applied to the overlying locational problem. Weiszfeld's algorithm is used to iteratively find the geometric median of the end-points, i.e., each subtour's first and last stop from the depot. This approach is compared to a classical (1+1)-ES employing the 1/5-th success rule. The two methods are evaluated on common instances of the PSFLRP, showing that for obtaining comparable total lengths, the Weiszfeld's-based procedure requires less VRP evaluations (by the ACO algorithm) than the ES.**

I. Introduction

In logistics, two important problems arise: Finding the location of a facility, and finding the optimal routes for trucks that deliver cargo from a *facility* to a number of customers. In Computer Science, these problems are known as the *Facility Location Problem* (FLP) and the *Vehicle Routing Problem* (VRP).

In the FLP (for an overview, see [8]), the goal is to find the optimal placement of facilities, such that the transportation cost from the facility to a set of clients is minimized. Location models can be split up in *discrete* and *continuous* models. In the first case, the facilities consist of a discrete set of locations, from which a subset has to be selected. In the second case, the facilities can be located anywhere on a plane, increasing freedom but also complexity. In the FLP, it is often assumed that a full truckload is delivered to each client; per trip only one customer is serviced.

In a real-world scenario, freight trucks usually carry goods for multiple customers. Goods are picked up at a depot and have to be dropped off at the customers. The problem of finding optimal routes for the trucks is a well-known problem; the Vehicle Routing Problem [6]. Many different variants of the VRP have been studied, such as the *Vehicle Routing Problem with Time Windows*, where each customer has to be visited in a certain time window, or the case where multiple depots exist, and each customer first has to be assigned to a depot. In this paper, we only look at a basic variant that deals with vehicles with a maximum capacity: the *Capacitated Vehicle Routing Problem* (CVRP).

While usually considered separately, the *Planar Location–Routing Problem* combines the FLP and VRP into one: Given a set of customers and the distances between them, find the optimal location of the depot, where trucks are allowed to visit multiple clients on a trip. Even though this problem combines short-term and long-term objectives (the routing of trucks and the location of the depot, respectively), the location of the depot can influence the routes of the trucks, creating possibilities for reducing the total covered distance. As logistic costs can consume a large portion of the budget of companies, even a slight decrease can lead to considerable savings. An overview of the recent research on Location–Routing problems is given by Prodhon and Prins [13].

The rest of this paper is structured as follows: In Section II we will present the problem in a more formal way. The two compared approaches are explained in Section III and we describe the experiments and obtained results in Section IV.

II. Problem Definition

The *Planar Single-Facility Location–Routing Problem* (PSFLRP) is a combination of two sub-problems, namely the *Vehicle Routing Problem* and the *Planar Facility Location Problem*. First we will explain these two problems and then show how they are combined in the PSFLRP.

A. Facility Location Problem

In the classical unconstrained Facility Location Problem (FLP), two sets are given: A set of customers C and a set of potential facilities F. The goal is to find a subset of open facilities $F_{\text{open}} \subseteq F$, such that the total cost is minimized when each customer is assigned the 'nearest' open facility. Opening a facility i has a cost of f_i, and a cost c_{ij} is associated with assigning a customer $j \in C$ to that facility. The objective is to minimize the total cost T:

$$T = \sum_{i \in F_{\text{open}}} f_i + \sum_{i \in F_{\text{open}}} \min_{j \in C} c_{ij}. \quad (1)$$

978-1-4799-7560-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SSCI.2015.243

IEEE computer society

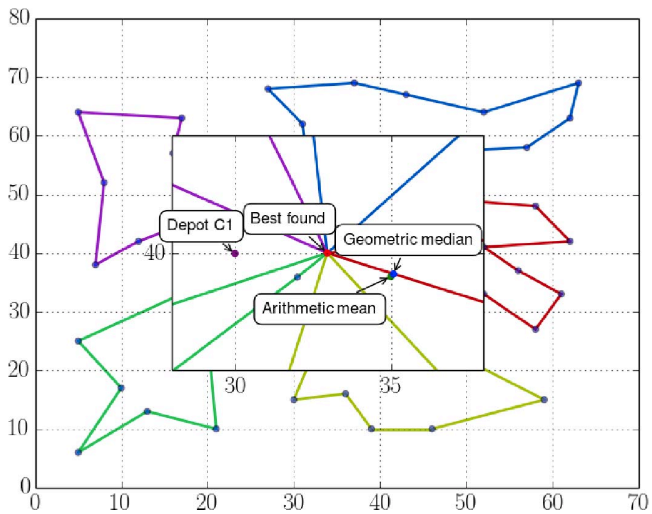

Fig. 1: *Depot Locations in C1.* The *geometric median* is the point that minimizes the sum of distances to a set of points, the *arithmetic mean* is the per-axis average of these points. The best location found by the algorithms compared in this paper is shown as well.

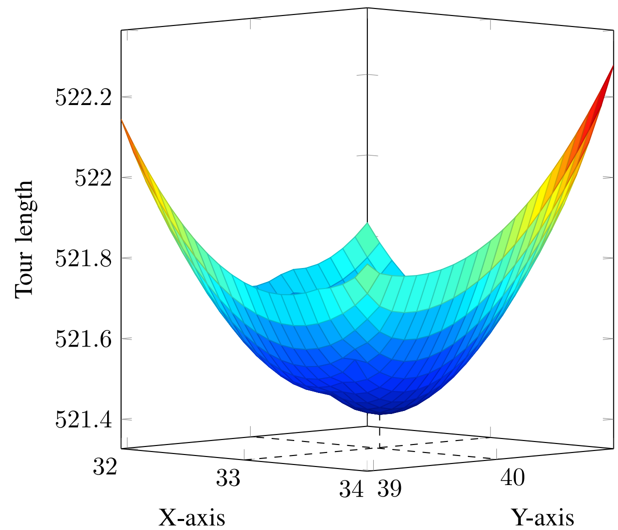

Fig. 2: *Influence of Depot Location.* Plot of the total route length change for varying depot locations in C1. Note that the resulting figure is not a regular shape.

In the *Planar* (or continuous) *FLP*, the location of facilities is not fixed. Instead of selecting from a discrete set of facility locations, a facility can be located anywhere on a continuous plane. This problem is more difficult, as F is no longer finite.

A special, simple instance of the Planar FLP is *Weber's problem* [21], which deals with finding exactly one point that minimizes the sum of routing costs to a set of points. Weber's problem is one of the most studied problems in continuous location theory. It allows for varying transportation costs by assigning weights to each client. In the planar case, the cost of assigning a customer to a facility can be represented by the Euclidean distance between the customer and the facility, reducing our problem to finding the *geometric median*, if all clients are weighted equally.

The geometric median problem is defined as: Given a set of n points $X = \{x_0, x_1, \ldots, x_n\}$, find the point GM that minimizes the sum of distances to these points, i.e.,

$$\text{GM} = \underset{y \in \mathbb{R}^2}{\operatorname{argmin}} \sum_{i=1}^{n} \|x_i - y\|_2. \tag{2}$$

Note that this is not necessarily the same as the *arithmetic mean* of the set of points, which minimizes the sum of squared distances to each point, see Figure 1 for a graphical overview of one of the tested problems. For determining the geometric median Weiszfeld's algorithm is used, see Section III-B.

B. Vehicle Routing Problem

The CVRP can be represented as a complete, weighted, directed graph $G = (V, A, d)$. The vertices V are comprised of a depot v_0 and n customers, $V = \{v_1, \ldots, v_n\}$. The arcs A represent the routes between vertices and are assigned weights d_{ij}, which are regarded as the distances between two vertices. Furthermore, each customer $\{v_i \in V | i \neq 0\}$ has a demand of q_i, a number that quantifies the volume of goods that have to be delivered. The quantity of goods at the facility v_0 is set to 0, and each customer has to be visited exactly once by one vehicle. A *tour* is a solution to the VRP, consisting of multiple *subtours*, a delivery run made by a single truck that starts and ends at the facility.

We consider a fleet of homogeneous trucks with a given maximum capacity. Each truck starts at the depot and visits customers in order to deliver goods. The total demand of all customers that a truck visits in one subtour may not exceed the maximum capacity. A feasible solution Z (where each client is visited exactly once) is constructed, such that $z_{ij} = 1$ if vertex v_j is visited after v_i in the solution representation, and $z_{ij} = 0$ otherwise. The aim is to find the route that minimizes the total travelling distance D of the trucks,

$$D = \sum_{i=0}^{n} \sum_{j=0}^{n} z_{ij} \cdot d_{ij}. \tag{3}$$

C. Planar Location–Routing Problem

In the FLP, it is assumed that a truck returns to the depot after it has visited a customer. In real-world scenarios, trucks do not always deliver on a single-customer trip basis, but carry goods for multiple customers. This is known as *Location–Routing*: The study of solving locational problems such that routing considerations are taken into account [15]. Trucks are allowed to stop at different customers before returning to the depot, as they are allowed in the VRP, and are still constrained by a maximum load that cannot be exceeded during a run.

In this paper, we consider the PSFLRP, where only a single facility is considered. The Euclidean distance is used to express the distances d_{ij} between vertices. There are some differences to the Planar FLP: No costs for opening a facility are considered, leaving only transportation costs. This allows us to use some of the benchmark problem sets from the VRP, without having to generate artificial costs for opening a facility. Furthermore, facilities are assumed to be uncapacitated. Incorporating a facility cost in our algorithm would be relatively simple, as we can add the cost of a facility to the cost of the routes.

In Figure 2, we see the influence of the depot location on the total route length. Because our VRP-solver is non-deterministic, the figure shows the minimum of 10 runs computed over a grid, centered close to the presumed optimal location of the depot. The resulting landscape is irregular, due to the fact that the order in which clients are visited can change depending on the location of the depot.

III. Approach

Both compared algorithms consist of two phases, solving the routing problem and subsequently improving the solution by moving the depot location (solving the location problem). These phases are iteratively executed for a maximum number of iterations. The routing problem is solved with an *Ant Colony Optimization* algorithm. Two approaches for the overlying location problem are compared: A method that uses Weiszfeld's algorithm (taken from Salhi and Nagy [15]) for determining the geometric median and a classical (1+1) *Evolution Strategy* (ES). This comparison allows us to investigate whether evolutionary algorithms can compete with heuristical methods such as Weiszfeld's algorithm. (1+1)-ES is chosen because of its simplicity and relatively fast convergence speed.

A. Ant Colony Optimization

In order to solve the VRP, we use an Ant Colony Optimization (ACO) algorithm. Bullnheimer et al. [2] were the first to use ACO for solving VRPs. ACO is derived from the way ants navigate the terrain to find potential food sources. In nature, ants leave a trail of *pheromones* on paths between the nest and the food sources. Other ants are more likely to follow strong pheromone trails, leaving even more pheromones, increasing the 'strength' of frequently chosen paths. Furthermore, pheromone evaporates, decreasing the probability that infrequently walked paths are chosen.

In the ACO algorithm, solutions are created by ants that walk around probabilistically, based on the pheromones on route segments. Once a route has been created, the route of each truck (i.e., each subtour) is optimized by a local search procedure. Finally, the algorithm reinforces the pheromone on arcs that were selected, and decreases pheromone on arcs that were not. These steps are shown in Algorithm 1.

Algorithm 1 Basic ACO algorithm

```
Initialization
for Number of iterations do
   For each ant, generate a new solution (Section III-A1)
   Improve solutions with local search (Section III-A2)
   Update pheromone information (Section III-A3)
```

1) Generation of Solutions: In many versions of ACO applied to the VRP (see, e.g., [2]), solutions are constructed sequentially. In planning routes, ants choose cities to visit until the maximum capacity of the truck has been reached. Each ant starts at a different initial city and builds its route from there in order to diversify the search. However, this way, ants were likely to start in the same cities for the remaining subtours, thereby creating solutions that are similar for all but the first subtour. Doerner et al. [9] present a different way of creating routes called *SavingsAnts*, which is based on the *Savings algorithm* [5].

In the Savings algorithm, first presented by Clarke and Wright [5], tours are initialized by assigning each customer to a separate subtour. Subsequently, for each combination of customers v_i and v_j, the *savings value* s_{ij} is calculated,

$$s_{ij} = d_{i0} + d_{0j} - d_{ij}, \tag{4}$$

where v_0 is the location of the depot. The savings value represents the costs that are saved by visiting client v_j after v_i, as a vehicle no longer has to cover the distances d_{i0} and d_{j0}, but instead travels the distance d_{ij} between nodes i and j. Note that this value cannot be negative when using Euclidean distances due to the triangle inequality. In the original Savings algorithm, the savings values are added to a sorted list. Subtours are then combined according to this list, starting with the highest savings, until no more feasible combinations exist. A combination of subtours $(v_0, \ldots, v_i, v_0)$ and $(v_0, v_j, \ldots, v_0)$ becomes $(v_0, \ldots, v_i, v_j, \ldots, v_0)$.

In the *SavingsAnts* algorithm, subtours are combined in a similar way. However, instead of only the savings values, pheromone information is also taken into account. τ_{ij} denotes the amount of pheromone on the edge between node v_i and v_j. A sorted list is created, this time consisting of the following values ξ:

$$\xi_{ij} = \tau_{ij}^{\alpha} \cdot s_{ij}^{\gamma}, \tag{5}$$

where the parameters α and γ represent the relative influence of the pheromones and the savings, respectively. In a following decision step, each ant probabilistically selects one of the k highest values in the list, and combines the two subtours. The probability p_{ij} of combining customers v_i and v_j is computed as follows:

$$p_{ij} = \begin{cases} \dfrac{\xi_{ij}}{\sum_{(h,l)\in\Omega_k} \xi_{hl}} & \text{if } \xi_{ij} \in \Omega_k \\ 0 & \text{otherwise} \end{cases}, \tag{6}$$

where Ω_k represents the list of the k maximum values. In one iteration of the ACO algorithm, we create m solutions, equal to the number of ants. As in the Savings algorithm, ants combine subtours, until no more feasible combinations exist with respect to capacity constraints.

2) 2-opt: The *2-opt heuristic* [7] is a local search method which is applied to each solution obtained by the ants in the previous step, before updating the pheromones. The 2-opt algorithm exchanges two edges with two new ones until no more improvements are possible. When finished, the result is called a *2-optimal tour*. The 2-opt algorithm is applied to each subtour, removing all crossing paths, which are always longer when using Euclidean distances as a measure.

TABLE I: *Comparison of VRP-Solvers.* The minimum tour lengths are given of the best known solutions from literature, the ACO algorithm applied in this paper and the ACO approach by Doerner et al [9].

Instance	Best known solution	Proposed method (ACO)	Deviation from best known (%)	Doerner et al. [9]	Deviation from best known (%)
C1	**524.61** [20]	**524.61**	**0.00**	**524.61**	**0.00**
C2	**835.32** [20]	842.15	0.82	838.60	0.39
C3	**826.14** [20]	834.14	0.97	838.38	1.48
C4	**1028.42** [20]	1049.12	2.01	1040.86	1.21
C5	**1291.45** [14]	1352.60	4.73	1307.78	1.26
C11	**1042.11** [20]	1044.65	0.24	1043.89	0.17
C12	**819.56** [20]	**819.56**	**0.00**	**819.56**	**0.00**
Avg. dev. (%)			1.25		0.86
E-n22-k4	**375** [11]	**375**	**0.00**	-	-
E-n23-k3	**569** [11]	**569**	**0.00**	-	-
E-n30-k4	-	505	-	-	-
E-n33-k4	**835** [11]	840	0.60	-	-
F-n45-k4	**724** [11]	**724**	**0.00**	-	-
F-n72-k4	**237** [11]	242	2.11	-	-
F-n135-k7	**1162** [11]	1173	0.95	-	-
Avg. dev. (%)			0.61		-

3) Pheromone Update: In the original *SavingsAnts* algorithm (see [9]), a rank-based pheromone update scheme is used. Instead, the update strategy devised in the *MAX-MIN Ant System* [19] by Stützle and Hoos is used.

In the MMAS, pheromones are placed as follows:

- Only one ant is allowed to place pheromones on edges. This can either be the *iteration-best* ant, the ant that produced the best solution in this iteration, or the *global-best* ant, the ant that found the best solution in the run so far;
- The range of pheromones is limited between the values τ_{min} and τ_{max};
- Pheromone trails are initialized to τ_{max}.

The following pheromone update rule is applied,

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij}^{\text{best}}, \quad (7)$$

where $\tau_{ij}(t)$ represents the amount of pheromone on the edge between vertex i and j at time t, ρ represents the pheromone evaporation rate and $\Delta\tau_{ij}^{\text{best}} = 1/f_{\text{best}}$, if node v_j is visited after node v_i, and $\Delta\tau_{ij}^{\text{best}} = 0$ otherwise. f_{best} is the fitness of either the iteration-best or the global-best ant.

After updating the pheromones, the values of τ are clamped between τ_{min} and τ_{max} in order to prevent the search from stagnating. Otherwise, it can occur that the pheromone on one edge is much higher than on all the others, causing the ant to pick this edge with a very high probability. This again increases the probability that this edge is chosen, as the ant will lay down more pheromones. By bounding the pheromone levels of infrequently and frequently covered edges, pheromone differences cannot become too high and search stagnation is prevented [19].

For the choice between the iteration-best and the global-best solution, we use a similar scheme as in [19], where a mixed strategy is applied. The frequency of using the global-best changes throughout a run of the algorithm. This frequency determines in which iteration i the global-best instead of the iteration-best is used.

- $1 \le i < 15$, global-best is not used.
- $15 \le i < 30$, global-best is used every 5^{th} iteration.
- $30 \le i < 50$, global-best is used every 3^{rd} iteration.
- $50 \le i < 75$, global-best is used every 2^{nd} iteration.
- $i \ge 75$, global-best is used in every iteration.

4) Parameters: We choose the relative influence of both the pheromone information $\alpha = 3$ and the savings values $\gamma = 3$. The number of ants m is set to 10, the pheromone evaporation rate ρ is set to 0.98. The number of iterations is fixed at 150.

In order to show that the ACO solver is comparable with other algorithms, results are presented in Table I. The presented algorithm shows similar results to the approach from Doerner et al. [9] on the small test cases, but falls short on the larger ones. This can be explained by the fact that in their approach the amount of iterations scales with the number of clients, while in the proposed ACO a fixed number of iterations is used.

TABLE II: *Comparison of Multiple PSFLRP Algorithms.* The compared algorithms perform better than known solvers so far, although better (pure) VRP solutions are known for some problem instances.

Instance	Best solution	(1+1)-ES (minimum)	Weiszfeld's-based (minimum)	Manzour-al-Ajdad et al. [12]	Schwardt and Fischer [17]	Schwardt and Dethloff [16]	(1+1)-ES (mean)	Weiszfeld's-based (mean)
C1	**521.41**	**521.41**	**521.41**	538.5	522.5	**521.4**	521.63	521.47
C2	***835.82**	**835.82**	835.83	849.8	879.0	888.8	838.77	838.48
C3	**825.60**	829.31	**825.60**	858.8	835.6	837.7	831.46	828.45
C4	***1044.69**	**1044.69**	1045.42	1088.2	1058.4	1068.8	1047.95	1047.36
C5	***1330.60**	1333.51	**1330.60**	1354.8	1344.0	1363.5	1339.11	1336.81
C11	**895.70**	897.23	**895.70**	902.7	905.7	907.4	898.31	896.37
C12	**818.65**	**818.65**	**818.65**	821.6	828.9	827.0	**818.65**	**818.65**
E-n22-k4	**374.66**	**374.66**	**374.66**	**374.7**	**374.7**	**374.7**	**374.66**	**374.66**
E-n23-k3	**488.22**	488.24	**488.22**	488.8	503.1	509.3	488.58	**488.22**
E-n30-k3	**498.63**	**498.63**	498.66	498.7	512.3	506.8	498.65	498.66
E-n33-k4	**480.11**	**480.11**	**480.11**	483.4	480.3	481.0	**480.11**	**480.11**
F-n45-k4	**719.74**	**719.74**	743.20	727.8	733.6	–	721.32	743.21
F-n72-k4	**216.58**	**216.58**	**216.58**	217.5	232.5	–	217.59	**216.58**
F-n135-k7	**1151.88**	1155.48	**1151.88**	1166.3	1216.4	–	1171.68	1154.07

* A better VRP solution (i.e., with the original depot) is known (see Table I).

B. Weiszfeld's-based Procedure

The first approach for tackling the PSFLRP (taken from Salhi and Nagy [15]) is based on Weiszfeld's algorithm. Given a solution for a VRP instance, the location of the depot only influences the distances between the facility and the *end-points* of the solution, if all subtours are kept the same. The end-points are the first or last customers a truck visits in each subtour, before leaving/returning to the facility (that is, $E = \{v_i \in V | z_{0i} = 1 \vee z_{i0} = 1\}$).

Weiszfeld's algorithm [22] can be used to find the geometric median, if all points are weighed equally. The geometric median is the point that minimizes the sum of Euclidean distances to a set of points J. The algorithm starts at a given coordinate, moving it to a new point (x, y) given by the following formula:

$$x = \frac{\sum_{j \in J}(w_j x_j / d_{0j})}{\sum_{j \in J}(w_j / d_{0j})}, y = \frac{\sum_{j \in J}(w_j y_j / d_{0j})}{\sum_{j \in J}(w_j / d_{0j})}, \qquad (8)$$

where x_j and y_j indicate the coordinates of node j. This is iteratively executed until the algorithm converges or the maximum number of evaluations is reached. One of the drawbacks of the algorithm is that it does not converge well when the initial point is placed at the location of an element of the set of points. If this occurs, we set the distance between the two points to a small value ϵ to avoid dividing by zero.

The Weiszfeld's-based procedure for solving PSFLRP, given in Algorithm 2, is used as follows. First, the geometric median of all customers is computed with Weiszfeld's algorithm, where equal weights $w_j = 1$ are assigned to all customers. This gives us an initial point for the depot. Subsequently the ACO algorithm described in Section III-A is run to compute the routes given the depot location and the set of customers. From the end-points of the initial tour, the new location for the facility is computed using Weiszfeld's algorithm (customers are weighed equally, setting all $w_j = 1$ and taking $J = E$). Finally, the new found facility location is used in the next iteration of the algorithm.

Algorithm 2 Weiszfeld's-based Procedure.

```
Initialization
for Number of iterations do
    Generate a solution for the VRP (Section III-A)
    Select end-points of solution (Section III-B)
    Recompute facility location
```

Because ACO is non-deterministic, the solutions (and therefore the end-points) that are returned by the ACO are not always the same. We adopt the end-points of the solution returned by the last iteration of the ACO algorithm, instead of the end-points of the best tour ever found. In a way, the non-deterministic nature of the ACO algorithm is used to decrease the chance of getting stuck in a local minimum.

C. Evolution Strategy

A classical (1+1)-Evolution Strategy (ES) [18] is used as a comparison to the Weiszfeld's-based procedure. Pseudocode is given in Algorithm 3.

The algorithm is initialized in the same way as the Weiszfeld's-based procedure, by computing the geometric median of all clients with Weiszfeld's algorithm to determine the initial parent. The (1+1)-ES works as follows: In each iteration one offspring is generated from a single parent, based on a normal distribution. The σ of this distribution is dependent on the success rate of the algorithm. For optimal convergence, success should be achieved once every five iterations. If the success rate is too low, the value of σ is *decreased*, which causes the search to focus more on solutions that are close to the current point. This should increase the success rate. If the success rate is too high, the value of σ is *increased*, taking larger steps towards the presumed optimum.

The algorithm runs for a fixed maximum number of iterations. The dimensionality of the solution to the PSFLRP; $n = 2$ is used to set how often σ is updated. The current success rate p_s is calculated over intervals of $10 \cdot n = 20$ trials. c is set to 0.85 following Schwefel [18].

Algorithm 3 (1+1)-ES algorithm, using 1/5-th success rule

```
Initialize and evaluate initial parent
while not terminated do
  Mutate parent into child
  Evaluate child
  If child is better, make it the new parent
  if t mod n = 0 then
    if p_s > 1/5 then
      σ(t) = σ(t − n)/c
    if p_s < 1/5 then
      σ(t) = σ(t − n) · c
    if p_s = 1/5 then
      σ(t) = σ(t − n)
  else
    σ(t) = σ(t − 1)
  t = t + 1
```

TABLE III: *Tested Problem Instances.*

Instance	Number of customers	Distribution of customers
C1	50	U
C2	75	U
C3	100	U
C4	150	U
C5	199	U
C11	120	C
C12	100	C
E-n22-k4	21	S
E-n23-k3	22	S
E-n30-k3	29	S
E-n33-k4	32	S
F-n45-k4	44	S/R
F-n72-k4	71	S/R
F-n135-k7	134	S/R

U: uniformly distributed, C: clustered, S: scattered, R: real-world problem data.

IV. Experiments

Weiszfeld's-based procedure, and (1+1)-ES are applied to the same set of well-known problem instances as used by Manzour-al-Ajdad et al. in [12] (note that their naming of instances does not correspond with the naming in this paper). The instances C1–C12 are taken from Christofides, Mingozzi and Toth [4], the E-instances are from Christofides and Eilon [3], and the F-instances are taken from Fischer [10]. These problem instances are commonly used to test algorithms for VRP, whereas in this paper the optimal depot location is searched for by the algorithms instead of using the one given in the instances. The problem instances contain a varying number of customers, as well as different distributions in order to test the algorithms. In Table III, these values are presented for each of the problem instances. Both proposed methods were run for 100 iterations of the ACO algorithm.

A. Results

In Table II, a comparison between the two proposed algorithms and methods presented by other papers on the same subject is shown. Reported values are the minimum and mean over 10 runs. As both tested methods use the ACO algorithm to solve the routing problem, we see that the Weiszfeld's-based procedure and the Evolution Strategy show very comparable minimum results. Even though the ES is only run for a limited number of evaluations, it can still achieve similar total tour lengths to the Weiszfeld's-based procedure. Looking at the mean values, we see that the Weiszfeld's-based procedure usually performs better, though marginally.

In the plot in Figure 3, the convergence of the two methods on the E-n33-k4 test problem is compared. The averages over 10 runs of both algorithms are presented, keeping track of the shortest route length found so far. Only the first 60 iterations are shown, since both algorithms have then converged to the common best found, with length 480.11. We see that the Weiszfeld's-based procedure converges faster than the ES.

Comparing Table I and Table II, we see that in most cases the results are better than the given location in the original test problems, but the analyzed algorithms do not always beat the minimum length of the best VRP-solvers. This is due to the fact that the ACO algorithm we use is not as good as some of the current VRP-solvers (see Table I), and the fact that usually only a slight decrease in route length is achieved by placing the depot at a different location.

V. Conclusions and Outlook

We compare two iterative approaches for solving the Planar Single-Facility Location–Routing Problem, that is, the problem of finding the location of a single depot in a planar landscape, such that the total distance travelled by trucks for servicing all clients is minimized. For each candidate depot location, it requires determining the total length of the route that visits all clients. This underlying problem, known as the Vehicle Routing Problem (VRP), is solved using an Ant Colony Optimization (ACO) algorithm, a non-deterministic method derived from the way in which ants find food sources in nature. The overlying locational problem is solved using a Weiszfeld's-based procedure, derived from work by Salhi and Nagy [15], and a classical (1+1)-Evolution Strategy (ES) employing the 1/5-th success rule [18].

The Weiszfeld's-based procedure uses Weiszfeld's algorithm at the heart for computing the geometric median, i.e., the point that minimizes the total distance to a given set of points, the end-points, of a VRP solution that is supplied by

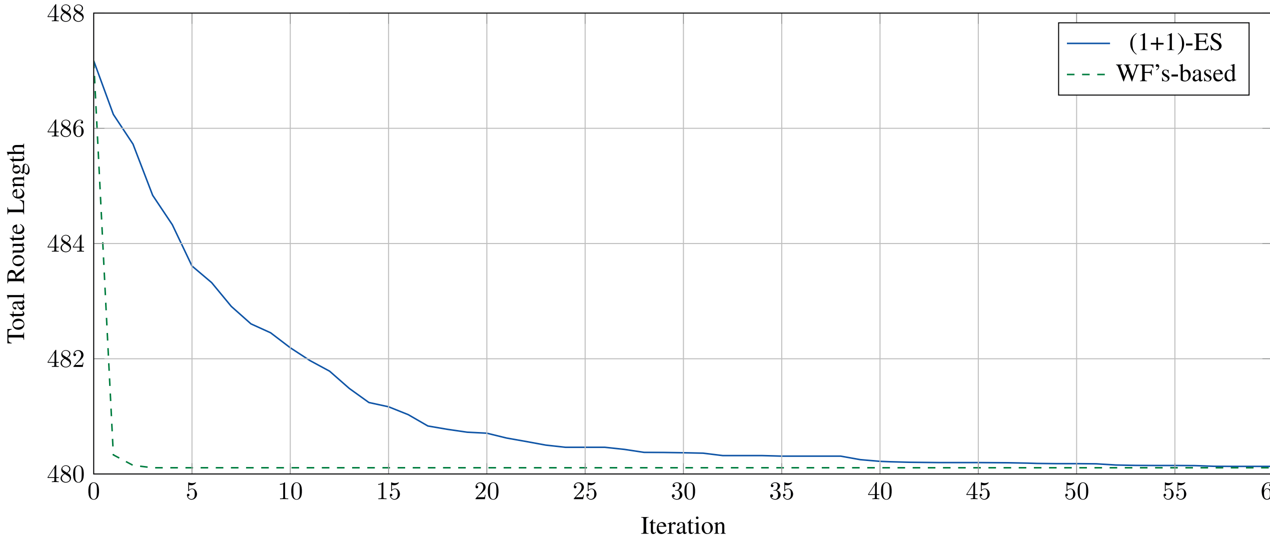

Fig. 3: *Comparison of Convergence.* The convergence of (1+1)-ES and Weiszfeld's-based procedure on E-n33-k4, illustrating that Weiszfeld's-based procedure converges faster, while the ES eventually obtains similar solution quality. Values are the averages over 10 runs of the shortest length found so far per algorithm.

the ACO algorithm. The end-points of a solution are those travelled to directly after or before stopping at the depot. Given these end-points, the geometric median provides the optimal depot location, after which the new depot location gives rise to a new VRP instance. Solving this new instance, using the ACO algorithm, can result in a different set of end-points, thus calling for another iteration of the Weiszfeld's-based procedure, until convergence. As comparison method, a steady state (1+1)-ES, i.e., generating a single new solution per iteration, is used. Both methods are run for a budget of 100 VRP evaluations by the ACO algorithm.

It is shown that the Weiszfeld's-based procedure converges faster, i.e., using less VRP evaluations, to a final solution than the (1+1)-ES. However, judging from the final results obtained after the full evaluation budget, it does not strictly outperform the ES. The ES is an inherently non-deterministic method, whereas the Weiszfeld's-based procedure is deterministic apart from the non-determinism introduced by the ACO algorithm.

Employing a more advanced non-deterministic optimization algorithm, such as the state-of-the-art active CMA-ES [1], may lead to outperforming the Weiszfeld's-based procedure on the tested evaluation budget. However, because of the relatively small differences between the two compared methods, the routing phase of the algorithm seems to be the most impactful on the quality of the final solution. We believe that future work should be focused on implementing a different routing-problem solver, such as the Robust branch-and-cut-and-price algorithm presented by Fukusawa [11]. This would allow for an even better comparison between the location-problem solvers.

Acknowledgments

The authors would like to thank Joost Leuven for his comments and suggestions and DENC Netherlands for enabling this study.

References

[1] D. V. Arnold and N. Hansen. Active covariance matrix adaptation for the (1+1)-cma-es. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, GECCO '10, pages 385–392, New York, NY, USA, 2010. ACM.

[2] B. Bullnheimer, R. F. Hartl, and C. Strauss. Applying the ant system to the vehicle routing problem. In S. Voss, S. Martello, I. H. Osman, and C. Roucairol, editors, *Meta-heuristics: Advances and trends in local search paradigms for optimization*, pages 87 – 100. Boston: Kluwer, 1999.

[3] N. Christofides and S. Eilon. An algorithm for the vehicle-dispatching problem. *Operational Research Quarterly*, 20:309–318, 1969.

[4] N. Christofides, A. Mingozzi, and P. Toth. The vehicle routing problem. *Combinatorial Optimization*, pages 315–338, 1979.

[5] G. Clarke and J. W. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4):568–581, 1964.

[6] J.-F. Cordeau, M. Gendreau, A. Hertz, G. Laporte, and J.-S. Sormany. New heuristics for the vehicle routing problem. In A. Langevin and D. Riopel, editors, *Logistics Systems: Design and Optimization*, pages 279–297. Springer US, 2005.

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

[8] M. S. Daskin. What you should know about location modeling. *Naval Research Logistics (NRL)*, 55(4):283–294, 2008.

[9] K. Doerner, M. Gronalt, R. F. Hartl, M. Reimann, C. Strauss, and M. Stummer. Savingsants for the vehicle routing problem. In S. Cagnoni, J. Gottlieb, E. Hart, M. Middendorf, and G. Raidl, editors, *Applications of Evolutionary Computing*, volume 2279 of *Lecture Notes in Computer Science*, pages 11–20. Springer Berlin Heidelberg, 2002.

[10] M. L. Fisher. Optimal solution of vehicle routing problems using minimum k-trees. *Operations Research*, 42(4):626–642, 1994.

[11] R. Fukasawa, H. Longo, J. Lysgaard, M. P. de Aragão, M. Reis, E. Uchoa, and R. F. Werneck. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical Programming*, 106(3):491–511, 2006.

[12] S. M. H. Manzour-al-Ajdad, S. Torabi, and S. Salhi. A hierarchical algorithm for the planar single-facility location routing problem. *Computers & Operations Research*, 39(2):461–470, 2012.

[13] C. Prodhon and C. Prins. A survey of recent research on location-routing problems. *European Journal of Operational Research*, 238(1):1 – 17, 2014.

[14] Y. Rochat and É. Taillard. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics*, 1(1):147–167, 1995.

[15] S. Salhi and G. Nagy. Local improvement in planar facility location using vehicle routing. *Annals of Operations Research*, 167(1):287–296, 2009.

[16] M. Schwardt and J. Dethloff. Solving a continuous location-routing problem by use of a self-organizing map. *International Journal of Physical Distribution & Logistics Management*, 35(6):390–408, 2005.

[17] M. Schwardt and K. Fischer. Combined location-routing problems – a neural network approach. *Annals of Operations Research*, 167(1):253–269, 2009.

[18] H.-P. Schwefel. *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie.* Birkhäuser Basel, 1977.

[19] T. Stützle and H. Hoos. Max-min ant system and local search for the traveling salesman problem. In *IEEE International Conference on Evolutionary Computation, 1997*, pages 309–314, Apr 1997.

[20] É. Taillard. Parallel iterative search methods for vehicle routing problems. *Networks*, 23(8):661–673, 1993.

[21] A. Weber. *Über den Standort der Industrien*. Tübingen, 1909.

[22] E. Weiszfeld. Sur le point pour lequel la somme des distances de n point donne est minimum. *Tohoku Mathematical Journal*, 43:335–386, 1937.