Local Search Based Approximation Algorithms for Two-Stage Stochastic Location Problems

Felix J.L. Willamowski[1(✉)] and Andreas Bley[2]

[1] Operations Research, RWTH Aachen University, 52072 Aachen, Germany
willamowski@or.rwth-aachen.de
[2] Institute of Mathematics, University of Kassel, 34132 Kassel, Germany
abley@mathematik.uni-kassel.de

Abstract. We present a nested local search algorithm to approximate several variants of metric two-stage stochastic facility location problems. These problems are generalizations of the well-studied metric uncapacitated facility location problem, taking uncertainties in demand values and costs into account. The proposed nested local search procedure uses three facility operations: adding, dropping, and swapping. To the best of our knowledge, this is the first constant-factor local search approximation for two-stage stochastic facility location problems.

Besides traditional direct assignments from clients to facilities, we also investigate shared connections via capacitated trees and tours. We obtain the first constant-factor approximation algorithms for both connection types in the setting of two-stage stochastic optimization. Our algorithms admit order-preserving metrics and thus significantly generalize and improve the allowed mutability of the metric in comparison to previous algorithms, which only allow scenario-dependent inflation factors.

1 Introduction

In this paper we study stochastic generalizations of the metric uncapacitated facility location (UFL) problem. The UFL problem was introduced in the early 1960's and is one of the most studied problems in the discrete optimization literature. The first constant-factor approximation algorithm for the metric case, where the assignment costs satisfy the triangle inequality, was presented in the late 1990's by Shmoys et al. [12]. From that time onward, many other constant-factor approximations have been developed, decreasing the approximation factor rapidly to 1.488, the currently best known proposed by Li [9]. Ye and Zhang [16] observed that so far each algorithm for approximating the metric UFL problem uses at least one of the following three paradigms: LP rounding, primal-dual, or local search techniques. LP rounding and primal-dual techniques were also applied to the two-stage stochastic version of the problem, but, to the best of our knowledge, no pure local search approaches have been used. One purpose of this paper is to close this gap, especially because local search turned out to be a powerful tool for approximating capacitated location problems. Moreover, the proposed local search approach allows more mutability of the metrics than previous approaches and it is very easy to implement in practice.

K. Jansen and M. Mastrolilli (Eds.): WAOA 2016, LNCS 10138, pp. 197–209, 2017.
DOI: 10.1007/978-3-319-51741-4_16

The metric two-stage stochastic uncapacitated facility location (tsUFL) problem was introduced in 2004 by Ravi and Sinha [10]. It models the task of locating *facilities* to serve demands of *clients* as a two-stage stochastic optimization problem with recourse, where a set of scenarios depict the possible outcomes of the future. The decision making process, essentially deciding which facilities to open, is divided into two stages. In a first stage, decisions are made with incomplete knowledge about the future, i.e., only the probability distribution of the scenarios with their parameters is known. In a second (recourse) stage, information is revealed about which scenario is realized and additional recourse decisions are made. The goal is to minimize the fixed first-stage and the expected second-stage cost. There are two main concepts to express the probability distribution of the scenarios in the literature. In the *scenario model* each scenario with its parameters and its associated probability is explicitly given as part of the input. An assumption commonly made in this model is that the number of scenarios is polynomial bounded by the other input parameters (e.g., number of facilities and clients). In the *black-box model*, the probability distribution is only given implicitly by an algorithm that draws independent samples of the distribution. Although the black-box model is more general than the scenario model, Charikar et al. [3] were able to show that, under reasonable assumptions on the distribution and losing only a factor $(1 + O(\varepsilon))$ in the objective, the black-box model reduces to the scenario model with only a polynomial number of samples. For this reason, we only consider the scenario model.

In the tsUFL problem we assume that the facilities opened in the first stage are present in each scenario, whereas facilities opened in the second stage exist only for their specific scenario. For each scenario, the clients have to be served by either an open facility of the first stage or by a facility opened in the second stage for this specific scenario. The service costs form a metric. Clearly, the approximability depends on how much the metric varies over the scenarios. We will extend the (rather restrictive) concept of scenario-dependent inflation factors used in previous works to a more general scenario-dependent *mutable metric*. The currently best known approximation algorithm for tsUFL with inflation factors is given by Ye and Zhang [16] with a factor of 1.86.

Formally, an instance of the tsUFL problem with mutable metric is given by a complete graph $G = (V, E)$ on the node set $V = \mathcal{C} \cup \mathcal{F}$ of clients $\mathcal{C}$ and facilities $\mathcal{F}$, *first-stage facility opening costs* $f_i \in \mathbb{Q}_{\geq 0}$, $i \in \mathcal{F}$, and a set of m possible scenarios. For the sake of simplicity, we index the scenarios by $k \in [m] := \{1, \dots, m\}$ and say scenario k instead of scenario indexed by k. Scenario k occurs with probability p_k and is defined by *second-stage facility opening costs* $f_i^k \in \mathbb{Q}_{\geq 0}$, $i \in \mathcal{F}$, a metric *service cost function* $c^k : E \to \mathbb{Q}_{\geq 0}$, and client demands $d_j^k \in \mathbb{Q}_{\geq 0}$, $j \in \mathcal{C}$. The goal is to find a set of *first-stage facilities* $F \subseteq \mathcal{F}$, which is independent of the realization of the scenario, and, for each scenario $k \in [m]$, a set of *second-stage facilities* $F^k \subseteq \mathcal{F}$ and an assignment $\sigma^k : \mathcal{C} \to F \cup F^k$, which minimize first-stage and expected second-stage costs

$$\sum_{i \in F} f_i + \sum_{k=1}^{m} p_k \cdot \Big(\sum_{i \in F^k} f_i^k + \sum_{j \in \mathcal{C}} d_j^k \cdot c^k(\sigma^k(j), j) \Big).$$

In order to appropriately model problem variants where multiple clients may share parts of a network that connect them to the facilities, we introduce the two-stage stochastic facility location problem with tree-connections (tsUFL-T). Formally, this problem is defined as follows. The graph, the first-stage facility opening costs, and the set of m possible scenarios with their parameters are given as in the tsUFL problem. Additionally, let $\mathcal{C}_+^k := \{j \in \mathcal{C} \mid d_j^k > 0\}$ denote the set of clients with positive demand in scenario k. The goal is to find a set of first-stage facilities $F \subseteq \mathcal{F}$ and, for each scenario k, a set of second-stage facilities $F^k \subseteq \mathcal{F}$ and a set $\mathcal{T}^k$ of trees in $G[F \cup F^k \cup \mathcal{C}_+^k]$ such that each tree contains exactly one facility, i.e., $|V(T) \cap \big(F \cup F^k\big)| = 1$ for all $T \in \mathcal{T}^k$, and all clients with positive demand are served, i.e., $\mathcal{C}_+^k \subseteq \bigcup_{T \in \mathcal{T}^k} V(T)$, which minimize

$$\sum_{i \in F} f_i + \sum_{k=1}^{m} p_k \cdot \Big(\sum_{i \in F^k} f_i^k + \sum_{T \in \mathcal{T}^k} \sum_{e \in E(T)} c^k(e) \Big).$$

As an intermediate step towards approximation algorithms for problems with capacitated trees and tours later in the paper, we first combine the connection types of tsUFL and tsUFL-T and study the two-stage stochastic uncapacitated facility location problem with direct and tree-connections (tsUFL-DT), where each client is served twice, directly and via a shared tree. This problem also may be of independent interest for some applications.

Since in many applications the connection network cannot handle unlimited amounts of flow, we examine capacitated network connection types like the metric two-stage stochastic capacitated-cable facility location (tsCCFL) problem. In this problem, we additionally need to select edge capacities that permit to route the clients' demands simultaneously to the open facilities. Formally, an instance of the tsCCFL problem is given by a complete graph $G = (V, E)$ with $\mathcal{F} \cup \mathcal{C} \subseteq V$. The first-stage facility opening costs and the set of scenarios with their parameters are defined as in the tsUFL problem. Additionally, there is a *cable capacity* $u \in \mathbb{Z}_{>0}$ limiting the demand flow. The task is to choose a set of first-stage facilities $F \subseteq \mathcal{F}$ and, for each scenario k, a set of second-stage facilities $F^k \subseteq \mathcal{F}$, a set $\mathcal{T}^k$ of trees in G such that each tree is rooted at an open facility and each client with positive demand is served, and a number of cables $z_e^k \in \mathbb{Z}_{\geq 0}$ for each edge $e \in \bigcup_{T \in \mathcal{T}^k} E(T)$ such that the flow given by routing all demands simultaneously via the tree edges to the open facilities does not exceed the edge capacities $z_e^k \cdot u$. As before, we wish to minimize the expected costs

$$\sum_{i \in F} f_i + \sum_{k=1}^{m} p_k \cdot \Big(\sum_{i \in F^k} f_i^k + \sum_{T \in \mathcal{T}^k} \sum_{e \in E(T)} z_e^k \cdot c^k(e) \Big).$$

As the second problem with a capacitated connection we consider the two-stage stochastic capacitated location routing (tsCLR) problem. It combines the tsUFL problem with the well-studied capacitated vehicle routing problem. Formally, an instance of tsCLR is given by a complete graph, first-stage facility opening costs, and a set of scenarios with parameters as in the tsUFL problem. Additionally, there is a *vehicle capacity* $u \in \mathbb{Z}_{>0}$. The task is to find a set of first-stage facilities $F \subseteq \mathcal{F}$ and, for each scenario k, a set of second-stage facilities $F^k \subseteq \mathcal{F}$, a set of tours $\mathcal{T}^k$ with *demand assignment* $x^k : \mathcal{C} \times \mathcal{T}^k \to \mathbb{Q}_{\geq 0}$ such that each tour is routed at a facility, i.e., $|V(T) \cap (F \cup F^k)| = 1$, each client is served, i.e., $\sum_{T \in \mathcal{T}^k : j \in V(T)} x^k(j, T) = d_j^k$ for all $j \in \mathcal{C}$, and the *capacity constraints* $\sum_{j \in \mathcal{C}} x^k(j, T) \leq u$ are satisfied for all $T \in \mathcal{T}^k$. The objective is to minimize the sum of fixed first-stage and expected second-stage costs

$$\sum_{i \in F} f_i + \sum_{k=1}^{m} p_k \cdot \Big(\sum_{i \in F^k} f_i^k + \sum_{T \in \mathcal{T}^k} \sum_{e \in E(T)} c^k(e) \Big).$$

The remainder of this paper is organized as follows. In Sect. 2, we discuss the complexity of the presented problems and introduce the type of service cost mutability that our local search approach can handle. Afterwards, in Sect. 3, we present our Nested Local Search algorithm for tsUFL, tsUFL-T, and tsUFL-DT and prove its constant approximation guarantees. In Sects. 4 and 5 we construct constant-factor approximations for tsCCFL and tsCLR by applying our local search to instances of the tsUFL-DT problem. Concluding remarks are given in Sect. 6. All omitted proofs can be found in a full version [15].

2 Hardness of Approximation

The tsUFL, tsCCFL, and tsCLR problem generalize the metric UFL problem with uniform demands. So, all hardness results are preserved and these problems are strongly NP-hard. In particular, the inapproximability results of Guha and Khuller [7] and Sviridenko [13] carry over. Hence, there is no 1.463-approximation algorithm for the problems, even when restricted to instances with a fixed metric and service cost 1 and 3, unless P = NP. The tsCLR problem also generalizes the capacitated vehicle routing problem, which is not approximable within a factor less than 1.5, unless P = NP [6]. By a reduction from UFL we obtain the following inapproximability result for tsUFL-T and tsUFL-DT.

Theorem 1. *There is no* 1.463*-factor approximation algorithm for the tsUFL-T and the tsUFL-DT problem, unless P = NP.*

The approximability of the stochastic problems depends on the mutability of the metric, since the hardness result for minimum set cover [5] carries over.

Theorem 2. *For* $\varepsilon > 0$*, there is no* $(1 - \varepsilon) \ln(m)$*-approximation algorithm for tsUFL(-T, -DT), tsCCFL, and tsCLR with a general mutable metric, if P ≠ NP.*

We show in Sect. 3 that the following class of metrics allows constant-factor approximations for the tsUFL, tsUFL-T, and tsUFL-DT problem.

Definition 3. *A family of metrics* $(c^k : (\mathcal{C} \cup \mathcal{F})^2 \to \mathbb{Q}_{\geq 0})_{k \in [m]}$ *is called* order-preserving*, if for each facility* $i \in \mathcal{F}$ *there exists an ordered list of* $\mathcal{F} \setminus \{i\}$ *that is (simultaneously) non-decreasingly sorted w.r.t.* $c^k(i, \cdot)$ *for each scenario* k*.*

Note that order-preserving metrics restrict only the distances among the facilities to form scenario-independent orders. Distances between clients and facilities may vary heavily from one scenario to another. In particular, the closest (open) facility from any client may change from one scenario to another. This generalizes the concept of inflation factors.

3 Nested Local Search Algorithm

In this section we present our Nested Local Search for the tsUFL, tsUFL-T, and tsUFL-DT problem. Given a feasible solution for one of these problems, we say a *feasible move* is an operation that adds an unchosen, deletes a chosen, swaps a chosen with an unchosen facility, or maintains the given facilities, and results in a feasible solution. Speaking of a first-stage or second-stage feasible move, we refer to these operations on first-stage or second-stage facilities, respectively.

Without any bounds on the cost reduction, local search algorithms may have exponential running time. To avoid this, we use the concept of δ-locally optimal solutions. If we guarantee a cost reduction by a factor of $0 < (1 - \delta) < 1$ in each iteration and choose δ appropriately, we prove a polynomial running time.

Definition 4. *A solution is denoted as* δ-locally optimal*, if no feasible first-stage move linked with any feasible second-stage move in each scenario decreases the total cost by more than a factor* $0 < (1 - \delta) < 1$*.*

3.1 Algorithm

As the scenarios are linked only to the first stage, we can consider them sequentially, exploring only polynomial many moves in total. Combining all described ideas, we get Nested Local Search illustrated below. The (re-)assignment of the clients to the chosen facilities is done optimally in all solution update steps. We may also assume that the sets of chosen first-stage and second-stage facilities are disjoint. Let `solution` be a feasible solution for one of the problems and denote the total cost by $C($`solution`$)$. We call a feasible first-stage move *unexplored* if this move was not even attempted to apply to `solution`. A feasible second-stage move is called *cost-reducing*, if applying the move does not increase the cost.

Input: Constant $0 < \delta < 1$ and a feasible solution `solution`.
Output: δ-locally optimal solution `solution`.
while *unexplored first-stage move of* `solution` *exists* **do**
  Select unexplored first-stage move, create solution `current`.
  Select most cost-reducing move for each scenario, update `current`.
  while $C(\texttt{current}) \leq (1-\delta) \cdot C(\texttt{solution})$ **do**
    `solution` := `current`
    Select most cost-reducing move for each scenario, update `current`.
return `solution`

Nested Local Search

Testing each unexplored move without changing the solution stops the algorithm. Therefore, the algorithm terminates with `solution`. Also, every feasible first-stage move has been evaluated in combination with a most cost-reducing second-stage move for each scenario, but the cost reduction was less than a factor of $(1-\delta)$. By definition, `solution` thus is δ-locally optimal.

3.2 Analysis

Applying any feasible move to a δ-locally optimal solution does not decrease the cost by more than a factor of $(1-\delta)$, even if all clients are reassigned optimally afterwards. We use this observation to create new solutions. By comparison of costs we get bounds on the service and the facility cost.

Lemma 5. *Let C_S, C_S^* denote the service costs and C_F, C_F^* the facility costs of a δ-locally optimal and an arbitrary feasible solution, respectively. Then*

$$C_S - \delta m \cdot |\mathcal{F}| \cdot (C_F + C_S) \leq C_F^* + C_S^*.$$

Lemma 6. *Let C_S, C_S^* denote the service costs and C_F, C_F^* the facility costs of a δ-locally optimal and an arbitrary feasible solution, respectively. Then*

$$C_F - \delta m \cdot |\mathcal{F}| \cdot (C_F + C_S) \leq C_F^* + 2 \cdot C_S^*.$$

Theorem 7. *Let $0 < \varepsilon \leq 1$. Then,* **Nested Local Search** *is a polynomial-time $(3+\varepsilon)$-approximation for tsUFL(-T, -DT) with order-preserving metrics.*

Proof. The number of feasible first-stage and second-stage moves in each scenario is bounded by $|\mathcal{F}|^2 + |\mathcal{F}|$ each. Updating a solution and finding a most cost-reducing move runs in polynomial time. Choosing $\delta := \varepsilon/(8m \cdot |\mathcal{F}|)$ and $0 < \varepsilon \leq 1$ results in a polynomial running time. With Lemmas 5 and 6 we obtain the bound $C_F + C_S \leq 3/(1-\varepsilon/4) \cdot (C_F^* + C_S^*)$ and the claim follows.

This result is tight, since Arya et al. [1] showed it for the UFL problem.

3.3 Improvements via Cost-Scaling and Greedy Augmentation

The cost-scaling technique introduced by Charikar and Guha [4] can be applied to the problems in a straightforward way (cf. [15]). Applying this technique, we obtain the following strengthened version of Theorem 7.

Theorem 8. *Let* $0 < \varepsilon \leq 1$*. Then,* **Nested Local Search** *with cost-scaling is a* $(1+\sqrt{2}+\varepsilon)$*-approximation for tsUFL(-T, -DT) with order-preserving metrics.*

Also, the well-known greedy augmentation technique for facility location problems can be applied in a straightforward way in combination with our Nested Local Search (cf. [15]). Combining all three techniques local search, cost-scaling, and greedy augmentation, we obtain the following stronger result.

Theorem 9. *Let* $0 < \varepsilon \leq 1$*. Then,* **Nested Local Search** *with cost-scaling and greedy augmentation is a* $(2.375+\varepsilon)$*-approximation algorithm for the tsUFL, tsUFL-T, and tsUFL-DT problem with order-preserving metrics.*

4 Two-Stage Capacitated-Cable Facility Location

In this section we introduce an approximation algorithm for tsCCFL. Initially, we transform an instance of tsCCFL to an instance of tsUFL-DT and show that the costs of a tsUFL-DT solution can be bounded by the costs of a tsCCFL solution. We then transform a solution to tsUFL-DT to one for tsCCFL.

Lemma 10. *Consider an instance* $\mathcal{I}$ *of tsCCFL and the instance* $\mathcal{J}$ *of tsUFL-DT obtained by scaling the demand values with* $1/u$*, omitting the capacity, and restricting the problem to* $G[\mathcal{F} \cup \mathcal{C}]$*. Then, for each solution of* $\mathcal{I}$ *with costs* $C_F^* + C_S^*$ *there is a solution of* $\mathcal{J}$ *with costs* $C'_F + C'_S$ *that* $C'_F \leq C_F^*$ *and* $C'_S \leq 3 \cdot C_S^*$*.*

4.1 Algorithm tsCCFL

We introduce at first an approximation algorithm for the tsCCFL problem with unit demands which we extend to general demand values later. We transform an instance of tsCCFL to an instance of tsUFL-DT as stated in Lemma 10. Then, we apply Nested Local Search with cost-scaling ($\beta = 6.67$) and greedy augmentation, open all obtained facilities and install one unit of capacity on each edge of the obtained trees. If a tree's demand exceeds the capacity we have to relieve this tree. Therefore, we adapt a procedure to relieve overloaded trees used by Ravi and Sinha [11] to approximate a deterministic version of the problem.

In detail, consider each node x where the subtrees of its children have demand at most u and the total demand of the (sub-)tree T_x is greater than u. To relieve overloaded trees, we choose the clients in the subtree of the children of x which are closest to an open facility $F \cup F^k$ and install unit capacity on each edge of the $\lfloor |D_x|/u \rfloor$ closest (w.r.t. c^k) client-facility pairs, but at most one per subtree. Considering one of those client-facility pairs (j_ℓ, i_ℓ) we reroute the

Input: Instance $\mathcal{I}$ of tsCCFL with unit demands and order-preserving metrics.
Output: Approximated solution of the tsCCFL instance $\mathcal{I}$.
Obtain tsUFL-DT instance $\mathcal{J}$ from $\mathcal{I}$ by scaling demand values with $1/u$.
Apply scaling ($\beta = 6.67$), Nested Local Search, and greedy augmentation to $\mathcal{J}$.
Obtain solution $\left(F, F^1, \ldots, F^m, \sigma, \mathcal{T}\right)$ and open all facilities $\left(F, F^1, \ldots, F^m\right)$.
for all *scenarios* k **do**
    Let T_x be the subtree of $T \in \mathcal{T}^k$ rooted at $x \in V(T)$ and $D_x := V(T_x) \cap \mathcal{C}$.
    for all *facilities* $i \in F \cup F^k$ **do**
        Install one copy of the cable on each edge in $E(T_i)$.
        while $|D_i| > u$ **do**
            Let $V' := \{x \in V(T_i) \mid |D_x| > u \text{ and } |D_\ell| \leq u \text{ for each child } \ell \text{ of } x\}$.
            for all $x \in V'$ **do**
                Let $(j_\ell, i_\ell) := \arg\min_{j' \in D_\ell, i' \in F \cup F^k} c^k(j', i')$ if ℓ is child of x.
                Install one cable on each edge (j_ℓ, i_ℓ) for the $\lfloor |D_x|/u \rfloor$ cheapest pairs (at most one for each child subtree of x).
                Route the whole demand in T_ℓ to i_ℓ via j_ℓ.
                Route remaining demand (in other subtrees T_ℓ of children of x) to a chosen pair or to x such that all new cables are saturated.
                Remove demands in D_i which are satisfied through a new cable.
Remove all cables with flow value zero and all facilities which serve no demand.

Algorithm tsCCFL

demand $|D_\ell| \leq u$ of the subtree T_ℓ to the facility i_ℓ. If a newly installed cable is not saturated, this means the demand flow on the arc is less than u, we reroute not satisfied demand of sibling subtrees via x to this facility. We repeat the relieve procedure, until the remaining demand assigned to any x is at most u. In the end, we clean up our solution by removing all unused cables and facilities.

4.2 Analysis

Theorem 11. *Let* $\varepsilon > 0$*. Then,* ***Algorithm tsCCFL*** *is a* $(3.9+\varepsilon)$*-approximation algorithm for tsCCFL with unit demands and order-preserving metrics.*

Proof. First, we show that the solution produced by Algorithm tsCCFL is feasible. Consider a subtree T_i with $|D_i| > u$ in scenario k and let $x \in V'$. We add as many additional cables and reroute demand in subtrees as long as the remaining demand assigned to x is at most u. Hence, V' decreases and therefore $|D_i|$ does. In the end, all edges of the subtrees fulfill the capacity constraint. However, we maybe reroute some demand via a client j_ℓ to a facility i_ℓ. And so we have to ensure that on these paths no capacity constraint is violated. It is maybe the case that after routing demand (via j_ℓ) to i_ℓ and using an arc (j, x), in a further step demand is routed using the arc (x, j). We use flow cancellation to reassign demand flow properly. In particular, flow cancellation only reduces flow in the direction toward the root of a considered tree. If any cable in a scenario k has flow toward the root, its value is, like mentioned before, at most u. Flow away from the root on a cable is only routed once and all the clients in the involved subtree are removed afterwards. The flow value is also at most u, ensuring satisfied cable capacities. The demand routed to a newly installed cable is exactly u. Each client whose demand is assigned to one new cable has a distance to any open facility of at least the length of the new cable. The cost of these cables can be bounded by the cost of the direct connections by aggregating demand. Hence, the total cable cost is bounded by the service costs of the tsUFL-DT solution.

Let C_F^* denote the facility costs and C_S^* the service costs of an optimal solution to an instance of tsCCFL. We know from Lemma 10 that there is a solution to the transformed tsUFL-DT instance with cost $C_F' + C_S'$ such that $C_F' \leq C_F^*$ and $C_S' \leq 3 \cdot C_S^*$. Since our analysis for Nested Local Search permits us to bound the costs by an arbitrary solution, we obtain with Lemmas 5 and 6, rescaling ($\beta = 6.67$), and greedy augmentation a solution with costs

$$\begin{aligned} C_F + C_S &\leq (2 + \ln(6.67) + \varepsilon') \cdot C_F' + \left(1 + \frac{2}{6.67} + \varepsilon'\right) \cdot C_S' \\ &\leq (3.9 + \varepsilon) \cdot (C_F^* + C_S^*). \end{aligned}$$

The best known guarantee for the deterministic version of the problem is $(\rho_{UFL} + \rho_{ST}) \leq 2.88$ [11], with the currently best approximation ratios of Steiner tree [2] and UFL [9]. If we consider the problem spanning only clients with positive demand values, our algorithm ($\beta = 3.33$) yields a $(3.203 + \varepsilon)$-approximation.

4.3 General Demands

Theorem 12. *Let $\varepsilon > 0$. There is a $(6.236 + \varepsilon)$-approximation algorithm for the tsCCFL problem with general demands and order-preserving metrics.*

Proof. The modification of Algorithm tsCCFL to deal with general demand values can be adapted from [11]. In the following we outline briefly the main changes in order to analyze the modifications. Again, we transform the tsCCFL instance as in Lemma 10 and apply rescaling ($\beta = 25.43$), Nested Local Search, and greedy augmentation. For each client which exceeds the capacity ($d_j^k > u$) we install $\lceil d_j^k / u \rceil$ cables on the edge $\{j, \sigma^k(j)\}$ and route its complete demand directly to the facility $\sigma^k(j)$. The service cost for each of these clients can be bounded by twice the costs of their direct connections. The remaining demands are processed as before except that we now accumulate demand to lie in between u and $2u$. Instead of installing one cable, we now install two copies of a cable and route the demand to the corresponding facility. Hence, we now can bound these costs by twice the direct connection costs. Since after greedy augmentation we have $C_S \leq C_S' + C_F'$, we obtain a solution for the tsCCFL problem with general demand values and order-preserving metrics with costs

$$\begin{aligned} C_F + 2 \cdot C_S &\leq (3 + \ln(25.43) + \varepsilon') \cdot C_F' + \left(2 + \frac{2}{25.43} + \varepsilon'\right) \cdot C_S' \\ &\leq (6.236 + \varepsilon) \cdot (C_F^* + C_S^*). \end{aligned}$$

The best guarantee in the deterministic case is $(2\rho_{UFL}+\rho_{ST}) \leq 4.37$ [2,9,11]. If we consider the problem spanning only clients with positive demand values, our algorithm ($\beta = 5.572$) yields a $(4.718 + \varepsilon)$-approximation.

5 Two-Stage Capacitated Location Routing

In this section we introduce an approximation algorithm for the tsCLR problem. Initially, we transform a tsCLR instance to one of tsUFL-DT and show that the costs of a tsUFL-DT solution can be bounded by the costs of a tsCLR solution. We then use a solution to tsUFL-DT to build one for tsCLR.

Lemma 13. *Consider an instance $\mathcal{I}$ of tsCLR and the instance $\mathcal{J}$ of tsUFL-DT obtained by scaling the demand values with $2/u$ and omitting the vehicle capacity. Then, for each solution of $\mathcal{I}$ with costs $C_F^* + C_S^*$ there exists a solution of $\mathcal{J}$ with costs $C'_F + C'_S$ such that $C'_F \leq C_F^*$ and $C'_S \leq 2 \cdot C_S^*$.*

5.1 Algorithm

We introduce an approximation algorithm for tsCLR by using our Nested Local Search with scaling ($\beta = 5.572$) and greedy augmentation on the tsUFL-DT instance obtained by the transformation described in Lemma 13. Consider a tree T_i with demand value D_i routed at facility $i \in F \cup F^k$. If the total demand of the tree satisfies the capacity constraint we obtain a feasible tour by doubling the edges and short-cutting. Otherwise, we relieve the tree by adapting a procedure by Harks et al. [8] for approximating a deterministic version of the problem.

In more detail, we open all obtained facilities. For each client j with demand value at least u we create $\lceil d_j^k/u \rceil$ times the tour $(\sigma^k(j), j, \sigma^k(j))$. Consider a node v where each children's subtree has demand at most u and the total demand of the tree T_v is greater than u. Find a partition $I = I_0 \dot{\cup} \ldots \dot{\cup} I_q$ of the children's subtrees such that the trees of each part obey the capacity constraint and all parts except I_0 have total demand greater than $u/2$. Note that the (sub-)tree structures remain unchanged while generating the partition. Such a partition can be found by a greedy algorithm. Consider a part I_p $(p \geq 1)$ and let j be the client in I_p with the smallest distance to an open facility. We construct a tour by doubling the edge $\{\sigma^k(j), j\}$ and all edges contained in I_p and short-cutting. In the end there is only part I_0 with total demand at most u. Again, we create a tour by doubling the edges and short-cutting. Finally, we remove unused facilities to save costs.

5.2 Analysis

Theorem 14. *Let $\varepsilon > 0$. Then, **Algorithm tsCLR** is a $(4.718+\varepsilon)$-approximation algorithm for the tsCLR problem with order-preserving metrics.*

Input: Instance $\mathcal{I}$ of the tsCLR problem.
Output: Approximated solution of the tsCLR instance $\mathcal{I}$.
Obtain tsUFL-DT instance $\mathcal{J}$ from $\mathcal{I}$ by scaling demand values with $2/u$.
Apply scaling ($\beta = 5.572$), Nested Local Search, and greedy augmentation to $\mathcal{J}$.
Obtain solution $\left(F, F^1, \ldots, F^m, \sigma, \mathcal{T}\right)$ and open all facilities $\left(F, F^1, \ldots, F^m\right)$.
for all *scenarios* k **do**
  for all $j \in \mathcal{C}$ *with* $d_j^k \geq u$ **do**
    Add $\lceil d_j^k/u \rceil$ copies of the tour $(\sigma^k(j), j, \sigma^k(j))$ and remove d_j^k.
  Let T_x be the subtree of $T \in \mathcal{T}^k$ rooted at x, and $D_x := \sum_{j \in \mathcal{C} \cap V(T_x)} d_j^k$.
  for all *facilities* $i \in F \cup F^k$ **do**
    while $D_i > u$ **do**
      Let $v \in \{x \in V(T_i) \mid D_x > u,\ D_\ell \leq u \text{ for all children } \ell \text{ of } x\}$.
      Let $I = \{V(T_\ell) \mid \ \ell \text{ is child of } v\} \cup \{\{v\}\}$.
      Find a partition of the trees $I = I_0 \dot{\cup} \ldots \dot{\cup} I_q$ such that
        $\sum_{x \in I_p} d_x^k \leq u$ for all $p \in \{0, \ldots, q\}$ and
        $\sum_{x \in I_p} d_x^k > u/2$ for all $p \in \{1, \ldots, q\}$.
      for all $p \in \{1, \ldots, q\}$ **do**
        Let $(i_\ell, x_\ell) := \arg\min_{i' \in F \cup F^k, x' \in V(I_p)} c^k(i', x')$.
        Construct a tour containing all clients in I_p and facility i_ℓ by doubling (i_ℓ, x_ℓ) and edges of all trees in I_p and short-cutting.
        Add the tour to the solution and remove corresponding subtrees.
    Construct a tour from T_i by doubling all edges and short-cutting.
    Add the tour to the solution.
Remove all facilities that are not contained in any tour.

Algorithm tsCLR

Proof. For all clients j with demand value $d_j^k \geq u$ in some scenario k we add $\lceil d_j^k/u \rceil$ copies of the tour $(\sigma^k(j), j, \sigma^k(j))$. Such a tour containing client j in scenario k has costs of at most $p_k \cdot \lceil d_j^k/u \rceil \cdot 2 \cdot c^k(\sigma^k(j), j)$. Since $\lceil d_j^k/u \rceil$ is bounded by $2 \cdot d_j^k/u$ for $d_j^k \geq u$, the costs for these clients are bounded by twice the direct connection costs of of these clients.

Consider a tour $T \in \mathcal{T}^k$ in scenario k containing facility i_ℓ and clients in I_p. The costs for T are at most $2 \cdot c^k(i_\ell, x_\ell)$ plus twice the costs of the corresponding subtrees. Since the choice of (i_ℓ, x_ℓ) was minimal w.r.t. c^k and the whole demand in T is at least $u/2$ we obtain $\sum_{x \in V(T)} 2 \cdot d_x^k/u \cdot c^k(\sigma^k(x), x) \geq c^k(i_\ell, x_\ell) \cdot \sum_{x \in V(T)} 2 \cdot d_x^k/u \geq c^k(i_\ell, x_\ell)$. Hence, the clients, carried by such tours, contribute to the costs with at most twice their direct connection costs and twice the costs of the corresponding subtrees. All other tours are built by doubling the edges of corresponding subtrees and short-cutting. These tours contribute to the costs with at most twice the costs of the corresponding subtrees. Summation over all scenarios and clients shows that the tour costs are bounded by twice the direct and twice the tree-connection costs in the constructed solution.

Let C_F^* denote the facility costs and C_S^* the service costs of an optimal solution to an instance of tsCLR. We know with Lemma 13 that there is a solution to the transformed tsUFL-DT instance with costs $C_F' + C_S'$ such that $C_F' \leq C_F^*$ and $C_S' \leq 2 \cdot C_S^*$. Since our analysis for Nested Local Search permits us to bound the costs by an arbitrary solution and $C_S \leq C_S' + C_F'$ holds after greedy augmentation, we obtain with cost-scaling ($\beta = 5.572$), Lemmas 5 and 6, and greedy augmentation a solution with costs

$$C_F + 2 \cdot C_S \leq (3 + \ln(5.572) + \varepsilon') \cdot C_F' + \left(2 + \frac{2}{5.572} + \varepsilon'\right) \cdot C_S'$$
$$\leq (4.718 + \varepsilon) \cdot (C_F^* + C_S^*).$$

The best known approximation algorithm for the deterministic problem has a guarantee of 4.38 and is due to Harks et al. [8]. So our algorithm produces only a slightly worse approximation factor in the two-stage stochastic case.

6 Conclusion

In this paper we introduced Nested Local Search, showing that pure local search applies to metric two-stage stochastic facility location problems. Our analysis lead to a tight $(3 + \varepsilon)$-approximation for the pure local search and to a $(2.375 + \varepsilon)$-factor approximation algorithm for local search combined with rescaling and greedy augmentation techniques. Moreover Nested Local Search allows us to generalize the mutability of the metric in contrast to previous algorithms, which only permit scenario-dependent inflation factors, to order-preserving metrics. Furthermore, we obtained the first constant-factor approximation algorithms for tsCCFL and tsCLR with guarantees $(6.236 + \varepsilon)$ and $(4.718 + \varepsilon)$, respectively.

It would be interesting to know if our new approach combining direct and tree-connections in one facility location problem could lead to improved approximation ratios also for the deterministic problems. Moreover, it would be interesting to study local search techniques for variants of two-stage stochastic capacitated facility location problems, as they proved to be very useful in the deterministic case.

References

1. Arya, V., Garg, N., Khandekar, R., Meyerson, A., Munagala, K., Pandit, V.: Local search heuristics for k-median and facility location problems. SIAM J. Comput. **33**, 544–562 (2004)
2. Byrka, J., Grandoni, F., Rothvoss, T., Sanità, L.: Steiner tree approximation via iterative randomized rounding. J. ACM **60**, 6 (2013)
3. Charikar, M., Chekuri, C., Pál, M.: Sampling bounds for stochastic optimization. In: Chekuri, C., Jansen, K., Rolim, J.D.P., Trevisan, L. (eds.) APPROX/RANDOM -2005. LNCS, vol. 3624, pp. 257–269. Springer, Heidelberg (2005). doi:10.1007/11538462_22

4. Charikar, M., Guha, S.: Improved combinatorial algorithms for the facility location and k-median problems. In: 40th Annual Symposium on Foundations of Computer Science (1999)
5. Dinur, I., Steurer, D.: Analytical approach to parallel repetition. In: Proceedings of the 46th Annual ACM Symposium on Theory of Computing, STOC 2014 (2014)
6. Golden, B.L., Wong, R.T.: Capacitated arc routing problems. Networks **11**, 305–315 (1981)
7. Guha, S., Khuller, S.: Greedy strikes back: improved facility location algorithms. In: Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (1998)
8. Harks, T., König, F.G., Matuschke, J.: Approximation algorithms for capacitated location routing. Transp. Sci. **47**, 3–22 (2013)
9. Li, S.: A 1.488 approximation algorithm for the uncapacitated facility location problem. In: Aceto, L., Henzinger, M., Sgall, J. (eds.) ICALP 2011. LNCS, vol. 6756, pp. 77–88. Springer, Heidelberg (2011). doi:10.1007/978-3-642-22012-8_5
10. Ravi, R., Sinha, A.: Hedging uncertainty: approximation algorithms for stochastic optimization problems. In: Bienstock, D., Nemhauser, G. (eds.) IPCO 2004. LNCS, vol. 3064, pp. 101–115. Springer, Heidelberg (2004). doi:10.1007/978-3-540-25960-2_8
11. Ravi, R., Sinha, A.: Approximation algorithms for problems combining facility location and network design. Oper. Res. **54**, 73–81 (2006)
12. Shmoys, D.B., Tardos, E., Aardal, K.: Approximation algorithms for facility location problems (extended abstract). In: Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing (1997)
13. Sviridenko, M.: Unpublished (cf. [14])
14. Shmoys, D.B.: Approximation algorithms for facility location problems. In: Jansen, K., Khuller, S. (eds.) APPROX 2000. LNCS, vol. 1913, pp. 27–32. Springer, Heidelberg (2000). doi:10.1007/3-540-44436-X_4
15. Willamowski, F., Bley, A.: Local search based approximation algorithms for two-stage stochastic location problems. Technical report 2016–036, Operations Research, RWTH Aachen University, October 2016
16. Ye, Y., Zhang, J.: An approximation algorithm for the dynamic facility location problem. In: Cheng, M.X., Li, Y., Du, D.-Z. (eds.) Combinatorial Optimization in Communication Networks. Combinatorial Optimization, vol. 18. Springer, Heidelberg (2006)