A Lagrangian Decomposition Based Heuristic for Capacitated Connected Facility Location*

Markus Leitner*,† Günther R. Raidl†

*Department of Telematics/Network Engineering, Carinthia University of Applied Sciences
Primoschgasse 8, 9020, Klagenfurt, Austria
markus.leitner@fh-kaernten.at

†Institute of Computer Graphics and Algorithms, Vienna University of Technology
Favoritenstraße 9-11, 1040, Vienna, Austria
raidl@ads.tuwien.ac.at

1 Introduction

We consider a generalized version of the rooted Connected Facility Location Problem (ConFL) with capacities and prizes on clients as well as capacity constraints on potential facilities. Furthermore, we are interested in selecting and connecting the most profitable client subset (i.e. a prize collecting variant) instead of mandatorily connecting all clients.

Connected Facility Location Problems occur for instance when increasing the bandwidth of existing networks to meet growing bandwidth requirements of customers [12, 13]. In such scenarios new routes are installed between some source and so called facilities acting as mediation points between the so far existing and the newly installed network. Each facility is able to meet the demands of several assigned customers up to some maximum available capacity (*capacity constraints*). Furthermore, next to costs for installing routes to facilities, facility installation costs (*opening costs*) as well as costs for assigning a customer to a facility (*assignment costs*) might occur. On the other hand for each supplied customer a return of invest (*customer prize*) is obtained.

Formally, we are given an undirected graph $G = (V, E)$ with a dedicated root node $0 \in V$ and edge costs $c_e \geq 0$, $\forall e = (u, v) \in E$, corresponding to the costs of installing a new route between u and v. Furthermore, we are given a set of potential facility locations $F \subseteq V$ with associated opening costs $f_i \geq 0$ and maximum assignable demands $D_i \in \mathbb{N}_0$, $\forall i \in F$, as well as clients C with individual demands $d_k \in \mathbb{N}_0$ and prizes $p_k \geq 0$, $\forall k \in C$, (i.e. the expected return on invest). Finally we are given costs $a_{ik} \geq 0$, $\forall i \in F$, $\forall k \in C$ for assigning the complete demand of client k to facility i. If a client k may not be assigned to a potential facility i we assume $a_{ik} = \infty$.

A solution to this Capacitated Connected Facility Location Problem (CConFL) $S = (R, T, F', C', \alpha)$ consists of a Steiner tree (R, T) with $R \subseteq V$, $T \subseteq E$ connecting the set of opened facilities $F' \subseteq F$

*This work is supported by the Austrian Research Promotion Agency (FFG) under grant 811378 and by the Austrian Science Fund (FWF) under grant P20342-N13.

with the root node 0 as well as a of subset of clients $C' \subseteq C$ and a mapping $\alpha : C' \to F'$ feasibly (i.e. respecting the capacity constraints) assigning clients $c \in C'$ to open facilities $i \in F'$. Each client may be assigned to at most one facility (single source). Let $\alpha_k \in F'$ denote the facility a customer $k \in C'$ is assigned to. The objective function of CConFL is given by

$$c(S) = \sum_{e \in T} c_e + \sum_{i \in F'} f_i + \sum_{k \in C'} a_{\alpha_k,k} + \sum_{k \in C \setminus C'} p_k$$

An optimal solution S^* (i.e. a most profitable one) is given by the minimal objective value, i.e. $c(S^*) \leq c(S)$ for all feasible solutions S. Figure 1 depicts an exemplary problem instance of CConFL while Figure 2 visualizes a possible solution to this instance. Since CConFL combines the Steiner Tree Problem (STP) on a graph with the Single Source Capacitated Facility Location Problem (SSCFLP) which are both NP-hard, CConFL is NP-hard, too.

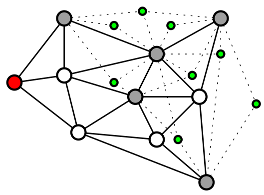


Figure 1: Problem instance.

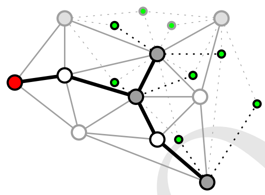
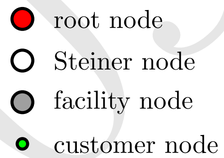


Figure 2: Exemplary solution.



Preprocessing: In a preprocessing step we remove all non-profitable assignments of customers $k \in C$ to facilities $i \in F$ (i.e. all assignments where $a_{ik} \geq p_k$). Furthermore, we identify and remove all facilities that are obviously not part of an optimal solution by solving a fractional knapsack problem for each facility $i \in F$ with knapsack size D_i, and one item with weight d_k and profit $p_k - a_{ik}$ for each assignable customer. Facilities $i \in F$ can be removed if the profit of the optimal solution to this knapsack problem is smaller than or equal to the facility opening costs f_i. For the remainder of this paper, we assume that the previously defined sets C and F represent the corresponding sets after all preprocessing steps. Furthermore, we denote by $C_i = \{k \in C \mid a_{ik} \leq p_k\}$ the resulting set of customers that may be assigned to a facility $i \in F$ and by $F_k = \{i \in F \mid k \in C_i\}$ the set of facilities a customer $k \in C$ may be assigned to. For better readability we use (u, v) for undirected as well as directed edges whenever the meaning is clear from the context.

We briefly sketch related work in Section 2 before presenting a detailed description of our Lagrangian decomposition approach in Section 3. While Section 4 explains the Lagrangian heuristic used to generate feasible solutions, Section 5 shows how these solutions are improved by local search in order to reduce the resulting optimality gaps. An exact approach to CConFL is presented in Section 6, computational results are given in Section 7 and we finally conclude in Section 8.

2 Related Work

Karger and Minkoff [10] motivated Connected Facility Location by the *maybecast* problem and described a constant factor approximation. Since then several approximation results for different versions of ConFL have been suggested. A factor 8.55 primal-dual approximation algorithm as well as a factor 4.55 approximation for the rent-or-buy problem (no opening costs, facilities may be opened at all nodes) have been presented by Swamy and Kumar [17]. Recently, Hasan et al. [9] described a factor 8.29 approximation for the case of edge costs obeying the triangle inequality based on LP rounding and a factor 7 approximation in case all opening costs are equal. Eisenbrand et al. [6] presented a randomized approximation algorithm for ConFL with an expected approximation ratio of 4 and showed how their algorithm can be derandomized with a resulting approximation factor of 4.23. Ljubic [13] described a Variable Neighborhood Search (VNS) approach as well as an exact method based on directed connection cuts for the rooted CConFL, while Tomazic et al. [18] suggested a Greedy Randomized Adaptive Search Procedure and an exact approach based on a transformation to the Minimum Steiner Aborescence problem for an unrooted variant of ConFL. In earlier work [12] we presented two VNS variants for a prize collecting version with capacity constraints but which does not consider opening and assignment costs. To the best of our knowledge our concrete variant of the Connected Facility Location Problem which contains those previously discussed problem variants as special cases has not been considered so far.

3 Lagrangian Decomposition

To model CConFL by means of an Integer Linear Program (ILP), we define an extended graph $G' = (V', E')$ combining G with the set of customers C as additional nodes and potential assignments between facilities and customers as additional edges (*assignment edges*). Formally, G' is given by its node set $V' = V \cup C$ and its edge set $E' = E \cup \{(i, j) \mid i \in F \wedge j \in C_i\}$. Edge costs $c'_e \geq 0$ are defined as

$$c'_e = \begin{cases} c_e & \text{if } e \in E \\ a_{ik} & \text{else} \end{cases}, \forall e = (i, k) \in E'.$$

We model CConFL using binary variables x_e, $\forall e \in E'$, indicating whether an edge e is part of the solution ($x_e = 1$). Variables $z_i \in \{0, 1\}$, $\forall i \in F$, specify if a facility i is opened ($f_i = 1$) while variables $y_k \in \{0, 1\}$, $\forall k \in C$, denote if a customer k is assigned to an opened facility ($y_k = 1$). Finally variables $s^k_e \in \{0, 1\}$, $\forall k \in C$, $\forall e \in E'$ indicate whether an edge $e \in E'$ is part of the unique path from the root node to a connected customer node k ($s^k_e = 1$). By $P_k \in \{0, 1\}^{|\mathrm{E}'|}$ we denote all incidence vectors corresponding to paths from 0 to customer k using exactly one assignment edge $(i, k) \in E' \setminus E$.

$$\min \quad \sum_{e \in E'} c'_e x_e + \sum_{i \in F} f_i z_i + \sum_{k \in C} p_k (1 - y_k) \tag{1}$$

$$\text{s.t.} \quad s^k_e \leq x_e \qquad \forall k \in C,\ \forall e \in E' \tag{2}$$

$$s^k \in P_k \text{ if } y_k = 1 \qquad \forall k \in C \tag{3}$$

$$x_{ik} \leq z_i \qquad \forall i \in F,\ \forall k \in C_i \tag{4}$$

$$\sum_{k \in C_i} d_k x_{ik} \leq D_i z_i \qquad \forall i \in F \tag{5}$$

$$s^k_e \in \{0, 1\} \qquad \forall k \in C,\ \forall e \in E' \tag{6}$$

$$x_e \in \{0, 1\} \qquad \forall e \in E' \tag{7}$$

$$z_i \in \{0, 1\} \qquad \forall i \in F \tag{8}$$

$$y_k \in \{0, 1\} \qquad \forall k \in C \tag{9}$$

We relax the coupling constraints (2) in a classical Lagrangian fashion (see e.g. [4] for an introduction to Lagrangian relaxation), yielding model $LR(\pi)$ which is parametrized by the Lagrangian multipliers $\pi_{k,e} \geq 0$, $\forall k \in C,\ \forall e \in E'$:

$$\begin{aligned} \min \quad & \sum_{e\in E'} c'_e x_e + \sum_{i\in F} f_i z_i + \sum_{k\in C} p_k(1-y_k) + \sum_{k\in C}\sum_{e\in E'} \pi_{k,e}\cdot(s^k_e - x_e) = \\ & = \sum_{k\in C} p_k + \sum_{k\in C}\left(\sum_{e\in E'} \pi_{k,e} s^k_e - p_k y_k\right) + \sum_{e\in E'}\left(c'_e - \sum_{k\in C}\pi_{k,e}\right) x_e + \sum_{i\in F} f_i z_i \\ \text{s.t.} \quad & (3)-(9) \end{aligned}$$

Our model decomposes into two independent subproblems $LD_{s,y}(\pi)$ for determining variables s^k_e, $\forall k \in C$, $\forall e \in E'$ and y_k, $\forall k \in C$, and $LD_{x,f}(\pi)$ for determining variables x_e, $\forall e \in E'$, and f_i, $\forall i \in F$.

$$\begin{aligned} (LD_{s,y}(\pi)) \quad \min \quad & \sum_{k\in C} p_k + \sum_{k\in C}\left(\sum_{e\in E'} \pi_{k,e} f^k_e - p_k y_k\right) && (10) \\ \text{s.t.} \quad & s^k \in P_k \text{ if } y_k = 1 \qquad \forall k \in C && (11) \\ & (6), (9) && (12) \end{aligned}$$

$LD_{s,y}(\pi)$ can be solved by computing $q \in P_k$ corresponding to a cheapest path to each customer node $k \in C$ including exactly one assignment edge $(i,k) \in E' \setminus E$ using edge costs $\pi_{k,e}$. The corresponding variables s^k_e, $\forall e \in E' \mid q_e = 1$, and y_k are set to one if the total costs of such a path are smaller than p_k.

$$\begin{aligned} (LD_{x,f}(\pi)) \quad \min \quad & \sum_{e\in E}\left(c_e - \sum_{k\in C}\pi_{k,e}\right) x_e + \\ & + \sum_{i\in F} f_i z_i + \sum_{\substack{e=(i,k)\in E' \mid \\ i\in F \wedge k\in C_i}} \left(c'_{ik} - \sum_{k\in C}\pi_{k,e}\right) x_{ik} && (13) \\ \text{s.t.} \quad & \sum_{k\in C_i} d_k x_{ik} \leq D_i z_i \qquad \forall i \in F && (14) \\ & x_{ik} \leq z_i \qquad \forall i \in F,\ \forall k \in C_i && (15) \\ & (7), (8) && (16) \end{aligned}$$

For edges $e \in E$, $LD_{x,f}(\pi)$ can be trivially solved by inspection (i.e. $x_e = 1 \Leftrightarrow c_e < \sum_{k\in C}\pi_{k,e}$). To determine variables x_e for edges $e \in E' \setminus E$ and variables z_i a knapsack problem for each facility $i \in F$ with knapsack size D_i, items $e = (i,k)$ with profits $\sum_{k\in C}\pi_{k,e} - c'_e$ and weights d_k need to be solved. Obviously, we only need to consider items with positive profits in the resulting knapsack problems, which we solve in our implementation using the Combo algorithm[1] [14]. If the objective value of the optimal solution K^* of such a knapsack problem exceeds f_i, z_i and all variables x_e corresponding to items used in K^* are set to one. Note that, since $LD_{x,f}(\pi)$ does not possess the integrality property, better lower bounds may be determined than by the simpler LP relaxation of model (1)–(9).

The Lagrangian dual problem which is to find the optimal Lagrangian multipliers π^* maximizing the obtained lower bound can be approximately solved using sugradient like methods (since this maximization problem is convex and piecewise linear). Here, we use the volume algorithm [2] since preliminary tests indicated that it generates better bounds than a standard subgradient method as presented in [7].

4 Lagrangian Heuristic

While solving the Lagrangian dual problem using the volume algorithm [2] we do not only compute a valid lower bound to a problem instance at each iteration but also derive integer values for variables s_e^k, x_e, z_i, and y_k. Since those values usually do not describe a feasible solution to CConFL we apply a Lagrangian heuristic (LH) to deduce feasible solutions using the information provided by the actual solutions to $LD_{s,y}(\pi)$ and $LD_{x,f}(\pi)$. A solution to $LD_{s,y}(\pi)$, represents a subgraph of G' connecting the root node with customers. This subgraph might contain redundant edges as well as violated capacity constraints. A solution to $LD_{x,f}(\pi)$ feasibly assigns a subset of customers to opened facilities. However, those facilities may not be connected to the root node.

To create a feasible solution $S = (R_S, T_S, F_S, C_S, \alpha_S)$ we set $F_S = \{i \in F \mid \exists k \in C : s_{ik}^k = 1\}$. Let $C_i' = \{k \in C \mid s_{ik}^k = 1\}$ be the set of customers connected to a facility $i \in F_S$ due to $LD_{s,y}(\pi)$, $W_{i,k} = \{e \in E \mid s_e^k = 1\}$, $\forall k \in C_i'$, be the corresponding subpaths from 0 to i and $W_i = \text{argmin}_{W_{i,k} \mid k \in C_i} \{\sum_{e \in W_{i,k}} c_e\}$ denote the cheapest of those subpaths. We consider facilities $i \in F_S$ by the costs of their cheapest individual connection to the root node W_i in increasing order. We connect i to a partially created Steiner tree (R_S, T_S) – which initially consists only of the root node 0 – by adding the necessary edge subset $\{(v_0 = i, v_1), (v_1, v_2), \ldots (v_l, v_m)\}$, $(v_a, v_b) \in W_i$, $0 \leq a, b \leq m$, $v_i \notin R_S$, $1 \leq i \leq l$, $v_m \in R_S$, of W_i. For each considered facility $i \in F_S$ we assign the optimal subset of customers $C'' \subseteq C_i'$ by either solving a knapsack problem – again using the Combo algorithm [14] – in case $\sum_{k \in C_i'} d_k > D_i$ or simply assign all customers $k \in C_i'$ otherwise. To further improve S we consider and add assignments $\{(i,k) \mid i \in F_S \wedge k \in C \wedge x_{ik} = 1\}$ greedily in decreasing order of their efficiency values $\frac{p_k - c_{ik}'}{d_k}$ in case the corresponding customer has not been assigned to another facility and the capacity constraints are still met afterwards. Additional assignments are then added to S by applying an identical greedy assignment using all possible assignments of facilities $i \in F_S$. In case S is better than the so far best solution S' created by LH, i.e. $c(S) < c(S')$, we try to further improve S using the neighborhood structures described in the next section (S' is the best so far found solution before applying these improvements). Finally, we remove non-profitable parts from S using strong pruning as described in [16].

[1] http://www.diku.dk/~pisinger/codes.html

5 Solution Improvement

So far existing metaheuristic approaches for Connected Facility Location Problems [12, 13, 18] typically represent a candidate solution by its open facilities and apply moves changing this set of active facilities to improve solutions. However, since each facility move subsequently involves updating the Steiner tree connecting them as well as updating client assignments the evaluation of each move might be quite time consuming. As the Steiner tree problem as well as – in case of a capacity constrained version – the problem of optimally assigning clients to a given set of facilities are NP hard typically some heuristic is used to estimate the changes due to a candidate move. In contrast to these approaches, we do only improve a candidate solution S with respect to the edges used in the Steiner tree as well as by means of clients assigned to currently opened facilities. As will be shown by our computational results, the initial solutions derived by our Lagrangian heuristic usually allow for generating near optimal solutions without further improving the set of opened facilities. For each candidate solution we apply a local search procedure based on one neighborhood structure for each remaining solution aspect. The neighborhood structures are searched using a best improvement strategy and applied in the same order as presented here.

The *Key-Path Exchange Neighborhood* which is well known for Steiner tree problems – see e.g. [15] – tries to improve a path between any two key nodes $\mathcal{K} = \{0\} \cup F_S \cup \{v \in R_\mathrm{S} \mid \deg_S(v) \geq 3\}$ of S by replacing it by the shortest connection between its end nodes using the remaining solution edges as infrastructure (i.e. zero edge costs are assumed for them).

The *Customer Swap Neighborhood*, which has also been used by Contreras et al. [5] for SSCFL, tries to improve a solution with respect to the assignment between facilities and customers. This neighborhood consists of all solutions S' reachable from S by swapping the assignment of exactly two customers, i.e. if $\alpha_\mathrm{S}(k) = i$ and $\alpha_\mathrm{S}(l) = j$ for customers $k, l \in C_\mathrm{S}$ and facilities $i, j \in F_\mathrm{S}$, those assignments will be $\alpha_{\mathrm{S}'}(k) = j$ and $\alpha_{\mathrm{S}'}(l) = i$ in S'.

6 Multi-Commodity Flow Formulation

To evaluate the performance of our Lagrangian decomposition approach, we further tried to solve a formulation for CConFL based on directed multi-commodity flows. For each facility $i \in F$ we define a corresponding set of relevant arcs $A_i = A_0 \cup \{(u, v), (v, u) \mid (u, v) \in E \wedge u, v \notin \{0, i\}\} \cup \{(v, i) \mid (v, i) \in E\}$ that may be used to connect i with the root node, where $A_0 = \{(0, v) \mid (0, v) \in E\}$.

We use binary variables x_e, $\forall e \in E'$, to specify whether an edge $e \in E'$ is part of the solution ($x_e = 1$) or not, and binary variables y_k, $\forall k \in C$, to indicate whether a customer $k \in C$ is assigned to an opened facility ($y_k = 1$) or not. Furthermore, we use variables $z_i \in [0, 1]$, $\forall i \in F$, to denote if a facility i is opened ($z_i = 1$), and flow variables $s^i_{uv} \in [0, 1]$, $\forall i \in F$, $\forall (u, v) \in A_i$, to indicate if arc $(u, v) \in A_i$ is used in the connection of facility $i \in F$ to the root node 0 ($s^i_{uv} = 1$) or not.

$$(dMCF_f) \quad \min \quad \sum_{e \in E'} c'_e x_e + \sum_{i \in F} f_i z_i + \sum_{k \in C} p_k (1 - y_k) \tag{17}$$

$$\text{s.t.} \quad \sum_{(u,v) \in A_i} s^i_{uv} - \sum_{(v,u) \in A_i} s^i_{vu} = \begin{cases} -z_i & \text{if } v = 0 \\ z_i & \text{if } v = i \\ 0 & \text{otherwise} \end{cases} \qquad \forall i \in F,\ \forall v \in V \tag{18}$$

$$s^i_{uv} + s^i_{vu} \leq x_{uv} \qquad \forall i \in F,\ \forall (u,v) \in E \qquad (19)$$

$$x_{ik} \leq z_i \qquad \forall (i,k) \in E' \mid k \in C \qquad (20)$$

$$\sum_{k \in C_i} d_k x_{ik} \leq D_i z_i \qquad \forall i \in F \qquad (21)$$

$$\sum_{i \in F_k} x_{ik} \geq y_k \qquad \forall k \in C \qquad (22)$$

$$0 \leq s^i_{uv} \leq 1 \qquad \forall i \in F,\ \forall (u,v) \in A_i \qquad (23)$$

$$0 \leq z_i \leq 1 \qquad \forall i \in F \qquad (24)$$

$$x_e \in \{0,1\} \qquad \forall e \in E' \qquad (25)$$

$$y_k \in \{0,1\} \qquad \forall k \in C \qquad (26)$$

Alternatively, we could define a directed multi-commodity flow model based on sending one unit of flow from the root node 0 to each customer node $k \in C$. Such a model would be a concrete instantiation of our ILP introduced in Section 3. However, since *dMCF*$_f$ is stronger than such a model, we decided to evaluate the performance of our Lagrangian decomposition approach against this theoretically stronger model.

7 Computational Results

For our experiments, we combined instances for the capacitated facility location (CFL) problem created with the instance generator[2] from [11] with instances for the Steiner tree problem from the OR-library[3] in the following way. We select the first node from the STP instance as root node and randomly select $|F|$ other nodes as potential facility locations. Customers, assignment costs, demands as well as capacities for each facility are given in the CFL instance. Let $\overline{a}(k) = \frac{\sum_{i \in F} a_{ik}}{|F_k|}$ denote the average assignment costs of customer k, $a_{\max}(k) = \max_{i \in F_k}\{a_{ik}\}$ the maximal assignment costs of customer k and $\overline{f} = \frac{\sum_{i \in F} f_i}{|F|}$ the average facility opening costs. We randomly select the prize $p_k \in \mathbb{N}_0$ of customer k from the interval $\left[\overline{a}(k), a_{\max}(k) + \overline{f}\right]$ ensuring that each customer may be assigned to the majority of potential facilities in a profitable way. Finally we apply degree-one and degree-two filtering on the potential edges as described in [3]. All instances can be downloaded at `http://www.ads.tuwien.ac.at/people/mleitner/cconfl/instances.tar.gz`.

Table 1 summarizes our computational results which have been computed using a single core of an Intel Xeon 5150 with 2.66GHz. ILOG CPLEX 11.2 has been used for solving *dMCF*$_f$ and its linear relaxation *dMCF*$_f$* and we used an absolute time limit of 3600 seconds for each experiment. *LDP* denotes the Lagrangian decomposition approach with primal improvement as explained in Section 5 while *LD* does not apply primal improvement. Since *dMCF*$_f$ never terminated before the time limit was reached we do not report its runtime in Table 1. We use the volume algorithm as described in [8] initializing Lagrangian multipliers by $\pi_{k,e} = c_e$ for assignment edges $e \in E' \setminus E$ and by $\pi_{k,e} = c_e/|C|$ for edges $e \in E$. The target value is initially set to $T = 1.2$ and multiplied by 1.1 in case $z_{\text{LB}} > 0.9 z_{\text{UB}}$ where z_{UB} and z_{LB} denote the so far best upper and lower bound. ρ is initially set to 0.1, multiplied by 0.67 after 20 non improving iterations in case $\rho > 10^{-4}$ and multiplied by 1.5

[2] http://alas.matf.bg.ac.yu/~kratica/instances/splp_gen_w32.zip
[3] http://people.brunel.ac.uk/~mastjjb/jeb/orlib/steininfo.html

in each improving iteration if $\rho < 5$ and if $\bar{v} \cdot v^t \geq 0$. Instead of computing λ_{OPT} as suggested in [8], we always use $\lambda = \lambda_{\mathrm{MAX}}$ which we initialize by $\lambda_{\mathrm{MAX}} = 0.01$. After every 100 iterations we multiply λ_{MAX} by 0.85 in case the lower bound did improve less than 1% and if $\lambda_{\mathrm{MAX}} > 10^{-5}$. The volume algorithm is terminated after 250 consecutive non improving iterations or if the time limit is reached.

We conclude that $dMCF_f$ is not only theoretically stronger but also in practice generates better lower bounds given enough time. However, even for medium sized instances which could be solved quite fast by *LD* and *LDP*, the linear relaxation of $dMCF_f$ could not always be computed within one hour. $dMCF_f$ could not solve any instances to proven optimality within the given time. While the feasible solutions (upper bounds) found by *LD* are for approximately 64% of the test instances better than those found by $dMCF_f$, *LDP* produces the best results with respect to upper bounds for 86% of the test instances. Therefore, *LDP* clearly outperforms the other two approaches by means of primal solution quality which documents the effectiveness of our neighborhood structures. Furthermore, especially for larger instances the resulting gaps by *LDP* are significantly smaller than those of the other two methods even though the lower bounds of $dMCF_f$ are typically better. Finally, with respect to CPU times we conclude that *LD* and *LDP* often need less CPU time than solving the linear relaxation of $dMCF_f$. Since we apply primal improvement only to the most promising candidate solutions generated by our Lagrangian heuristic no significant difference between *LD* and *LDP* with respect to the needed CPU time could be observed. Since the resulting gap between upper and lower bound of *LDP* never exceeded 5%, we recommend *LDP* for generating high quality solutions to CConFL with tight gaps in relatively short time.

8 Conclusions and Outlook

In this artice we introduced a new variant of the Connected Facility Location Problem with capacity constraints which includes several so far considered variants of ConFL as special cases. We modeled CConFL using directed multi-commodity flows. Furthermore, we presented a Lagrangian decomposition approach for CConFL as well as a hybrid method combining this approach with local search. Our computational results indicate, that the proposed hybrid method is able to generate high quality solutions with tight gaps in relatively short time.

In future we want to integrate a very large scale neighborhood search approach similar to the one presented by Ahuja et al. [1] in our hybrid Lagrangian decomposition approach as well as implement a pure metaheuristic approach to approximately solve very large instances of CConFL. Additionally, we are currently working on exact methods for medium sized instances based on branch-and-cut and branch-and-cut-and-price.

References

[1] R.K. Ahuja, J.B. Orlin, S. Pallottino, M.P. Scaparra, and M.G. Scutella. A multi-exchange heuristic for the single-source capacitated facility location problem. *Management Science*, 50(6):749–760, 2004.

[2] F. Barahona and R. Anbil. The volume algorithm: producing primal solutions with a subgradient method. *Mathematical Programming*, 87(3):385–399, 2000.

[3] J.E. Beasley. An algorithm for the Steiner problem in graphs. *Networks*, 14(1):147–159, 1984.

[4] J.E. Beasley. Lagrangean relaxation. In C.R. Reeves, editor, *Modern heuristic techniques in combinatorial problems.*, pages 243–303. Blackwell Scientific Publications, 1993.

[5] I.A. Contreras and J.A. Diaz. Scatter search for the single source capacitated facility location problem. *Annals of Operations Research*, 157(1):73–89, 2008.

[6] F. Eisenbrand, F. Grandoni, T. Rothvo, and G. Schfer. Approximating connected facility location problems via random facility sampling and core detouring. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 1174–1183, 2008.

[7] M. L. Fisher. The Lagrangian relaxation method for solving integer programming problems. *Management Science*, 27(1):1–18, 1981.

[8] M. Haouari and J.C. Siala. A hybrid Lagrangian genetic algorithm for the prize collecting Steiner tree problem. *Computers and Operations Research*, 33(5):1274–1288, 2006.

[9] M.K. Hasan, H. Jung, and K. Chwa. Approximation algorithms for connected facility location problems. *Journal of Combinatorial Optimization*, 16(2):155–172, 2008.

[10] D.R. Karger and M. Minkoff. Building Steiner trees with incomplete global knowledge. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 613–623. IEEE Computer Society, 2000.

[11] J. Kratica, D. Tosic, V. Filipovic, and I. Ljubic. Solving the simple plant location problem by genetic algorithm. *RAIRO Operations Research*, 35:127–142, 2001.

[12] M. Leitner and G.R. Raidl. Variable neighborhood search for a prize collecting capacity constrained connected facility location problem. In *Proceedings of the 2008 International Symposium on Applications and the Internet*, pages 233–236. IEEE Computer Society, 2008.

[13] I. Ljubić. A hybrid VNS for connected facility location. In T. Bartz-Beielstein et al., editors, *Hybrid Metaheuristics, 4th International Workshop, HM 2007*, volume 4771 of *LNCS*, pages 157–169. Springer, 2007.

[14] S. Martello, D. Pisinger, and P. Toth. Dynamic programming and strong bounds for the 0-1 knapsack problem. *Management Science*, 45(3):414–424, 1999.

[15] S.L. Martins, M.G.C. Resende, C.C. Ribeiro, and P.M. Pardalos. A parallel GRASP for the Steiner tree problem in graphs using a hybrid local search strategy. *Journal of Global Optimization*, 17(1-4):267–283, 2000.

[16] M. Minkoff. The prize collecting Steiner tree problem. Master's thesis, Massachusetts Institute of Technology, 2000.

[17] C. Swamy and A. Kumar. Primal-dual algorithms for connected facility location problems. *Algorithmica*, 40(4):245–269, 2004.

[18] A. Tomazic and I. Ljubić. A GRASP algorithm for the connected facility location problem. In *Proceedings of the 2008 International Symposium on Applications and the Internet*, pages 257–260. IEEE Computer Society, 2008.

Table 1: Computational Results

Instance					lower bound				upper bound			gap in %			CPU time		
Name	$\|F\|$	$\|C\|$	$\|V\|$	$\|E\|$	$dMCF_f{}^*$	$dMCF_f$	LD	LDP	$dMCF_f$	LD	LDP	$dMCF_f$	LD	LDP	$dMCF_f{}^*$	LD	LDP
c10-mo75	75	75	408	908	2878.7	**2882.8**	2851.9	2851.7	**2959.5**	2988.2	2962.4	**2.7**	4.8	3.9	154	74	73
c10-mq75	75	75	405	905	7095.2	**7100.5**	7079.2	7077.7	7182.9	7239.1	**7162.6**	**1.2**	2.3	**1.2**	181	95	67
c10-ms75	75	75	407	907	9506.3	**9510.7**	9479.9	9479.9	9573.1	9629.8	**9563.3**	**0.7**	1.6	0.9	299	143	141
c15-mo75	75	75	500	2500	2747.5	**2749.1**	2737.5	2738.2	2879.0	2855.2	**2815.3**	4.7	4.3	**2.8**	2092	91	111
c15-mq75	75	75	500	2500	7466.5	**7469.8**	7457.2	7455.7	7548.6	7576.1	**7510.7**	1.1	1.6	**0.7**	1815	127	94
c15-ms75	75	75	500	2500	9354.6	**9358.0**	9341.7	9342.3	10841.9	9487.5	**9428.8**	15.9	1.6	**0.9**	2008	195	159
d10-mo75	75	75	771	1770	2772.6	**2777.0**	2741.6	2741.8	**2845.3**	2921.9	2849.3	**2.5**	6.6	3.9	906	145	173
d10-mq75	75	75	775	1774	7295.0	**7302.3**	7280.9	7281.0	7373.1	7432.9	**7364.6**	**1.0**	2.1	1.1	292	125	154
d10-ms75	75	75	781	1780	10069.3	**10075.9**	10019.0	10017.9	10219.1	10257.3	**10166.9**	**1.4**	2.4	1.5	698	164	141
d15-mo75	75	75	1000	5000	-	**2645.0**	2636.6	2636.9	3370.2	2743.2	**2697.1**	27.4	4.0	**2.3**	3600	175	220
d15-mq75	75	75	1000	5000	-	**7379.4**	7370.8	7368.5	8493.6	7473.5	**7444.1**	15.1	1.4	**1.0**	3600	245	150
d15-ms75	75	75	1000	5000	-	**9234.8**	9221.6	9222.5	10933.6	9334.3	**9293.1**	18.4	1.2	**0.8**	3600	223	345
c10-mo100	100	100	406	906	3330.9	**3334.8**	3303.2	3297.7	**3399.9**	3486.1	3437.3	**2.0**	5.5	4.2	298	178	367
c10-mq100	100	100	406	906	9352.6	**9360.3**	9322.5	9322.6	9521.8	9610.5	**9491.5**	**1.7**	3.1	1.8	347	166	187
c10-ms100	100	100	416	916	11740.1	**11747.6**	11697.9	11696.6	11890.9	11979.1	**11877.1**	**1.2**	2.4	1.5	247	217	187
c15-mo100	100	100	500	2500	3422.6	**3424.7**	3413.8	3413.6	3881.8	3562.4	**3542.6**	13.3	4.4	**3.8**	3112	220	226
c15-mq100	100	100	500	2500	-	**9124.9**	9118.6	9113.3	9676.6	9331.9	**9214.8**	6.0	2.3	**1.1**	3600	188	151
c15-ms100	100	100	500	2500	-	**11281.1**	11264.0	11263.6	12619.2	11533.2	**11412.0**	11.9	2.4	**1.3**	3600	297	244
d10-mo100	100	100	788	1787	3376.7	**3381.2**	3337.8	3339.6	3492.1	3540.9	**3474.6**	**3.3**	6.1	4.0	401	252	328
d10-mq100	100	100	778	1777	9179.2	**9186.0**	9129.6	9128.1	**9354.7**	9406.5	9374.9	**1.8**	3.0	2.7	997	244	212
d10-ms100	100	100	783	1782	11049.0	**11056.6**	11000.4	11002.7	11185.5	11348.6	**11170.3**	**1.2**	3.2	1.5	678	209	260
d15-mo100	100	100	1000	5000	-	**3310.6**	3297.8	3298.6	3806.1	3454.6	**3424.1**	15.0	4.8	**3.8**	3600	412	367
d15-mq100	100	100	1000	5000	-	-	9149.5	**9150.4**	23780.0	9422.2	**9232.0**	-	3.0	**0.9**	3600	291	331
d15-ms100	100	100	1000	5000	-	-	**11309.2**	11309.1	30343.0	11549.3	**11501.1**	-	2.1	**1.7**	3600	334	438
c10-mo200	200	200	433	933	7116.2	**7120.5**	7052.9	7053.2	7489.8	7440.1	**7325.2**	5.2	5.5	**3.9**	529	2561	3600
c10-mq200	200	200	428	928	19270.3	**19277.1**	19211.7	19212.7	21089.5	19673.0	**19561.6**	9.4	2.4	**1.8**	1619	2566	2554
c10-ms200	200	200	431	931	25190.6	**25199.1**	25115.0	25115.1	31477.3	25680.7	**25532.3**	24.9	2.3	**1.7**	1617	3600	3600
c15-mo200	200	200	500	2500	-	**7135.2**	7108.8	7105.5	8360.9	**7427.4**	7450.5	17.2	**4.5**	4.9	3600	2714	2258
c15-mq200	200	200	500	2500	-	-	**19171.1**	19170.4	56699.0	19495.3	**19330.9**	-	1.7	**0.8**	3600	2358	2771
c15-ms200	200	200	500	2500	-	**24679.5**	24654.4	24655.1	26697.0	25302.2	**25043.5**	8.2	2.6	**1.6**	3600	3079	2948
d10-mo200	200	200	816	1815	7194.1	**7199.0**	7107.3	7106.9	7864.9	7599.7	**7448.3**	9.3	6.9	**4.8**	2635	3600	3048
d10-mq200	200	200	814	1813	-	**18798.4**	18720.6	18719.7	21130.5	19245.3	**19075.7**	12.4	2.8	**1.9**	3600	3600	3600
d10-ms200	200	200	806	1805	-	**24516.8**	24426.2	24425.6	27785.1	24856.3	**24762.7**	13.3	1.8	**1.4**	3600	3600	3600
d15-mo200	200	200	1000	5000	-	-	**7129.0**	7126.9	23975.0	7448.1	**7381.0**	-	4.5	**3.6**	3600	3600	3600
d15-mq200	200	200	1000	5000	-	-	19457.0	**19457.3**	61778.0	19880.4	**19673.0**	-	2.2	**1.1**	3600	3139	3195
d15-ms200	200	200	1000	5000	-	-	24007.7	**24008.9**	73434.0	24539.0	**24222.4**	-	2.2	**0.9**	3614	2998	3055