Dual Based Heuristics for the Connected Facility Location Problem

M. Gisela Bardossy* S. Raghavan*

*Robert H. Smith School of Business, University of Maryland
Van Munching Hall, College Park, MD 20742-1815, USA

Abstract

The Connected Facility Location (ConFL) Problem arises in the design of telecommunication networks where open facilities need to communicate with each other. As it combines features of the uncapacitated facility location problem with the Steiner tree design problem, the ConFL is an NP-complete problem. Dual-ascent has been a successful solution strategy for both these problems. Consequently, we propose dual based heuristics that combine dual-ascent and local improvements and together yield lower and upper bounds to the optimal solution. We discuss a wide range of computational experiments, which indicate that our heuristic is a very effective procedure that finds high quality solutions very rapidly.

Keywords: *connected facility location, dual-ascent heuristics, facility location, network design, Steiner tree design.*

1 Introduction

The recent growth of telecommunication networks coupled with digital data management has motivated a wide range of network design problems that combine facility location with connectivity requirements. In this paper we transform and formulate the ConFL as a directed Steiner tree problem with a unit degree constraint and propose dual-based heuristics that complement dual-ascent with local improvements to obtain high quality solutions rapidly. Since our procedure provides both upper and lower bounds it allows us to immediately assess the quality of the solutions.

The ConFL problem arises in multiple applications; for example, in the design of networks, where a single data item needs to be distributed to multiple clients, (Karger and Minkoff, [4]), or in the construction of distributed networks, where efficient data caching is required (Krick et al., [5]). The problem requires the construction of a tree that minimizes the facility opening costs and the actual network cost. Given a graph $G = (V, E)$, and a partition of the nodes in V into three sets: $D \subseteq V$, set of *demand* nodes; $F \subseteq V$, set of potential *facility* nodes; and $S \subseteq V$, set of potential *Steiner* nodes, we seek to find a minimum cost tree such that every demand node is assigned to an open facility and open facilities are connected through a Steiner tree. Potential facilities can perform two roles. On one hand, they can be turned into operating facilities, in which case they must be included in the tree. On the other hand, they can be used as a Steiner node to link the operating facilities. There are facility opening costs $f_i \geq 0$ for each facility $i \in F$ (that are only incurred when customers are assigned to the facilities), assignment costs $a_{ij} \geq 0$ for each $i \in F$ and $j \in D$, and edge costs $b_{ij} \geq 0$ for each edge $\{i, j\} \in E$. Then, the final network cost is given by $\sum_{i \in D} a_{j(i)i} + \sum_{i \in Z} f_i + \sum_{\{i,j\} \in T} b_{ij}$, where $j(i)$ is the facility serving demand node i, Z is the set of open facilities, and T is a Steiner tree connecting the open facilities. Figure 1 illustrates an example of ConFL problem and a feasible solution.

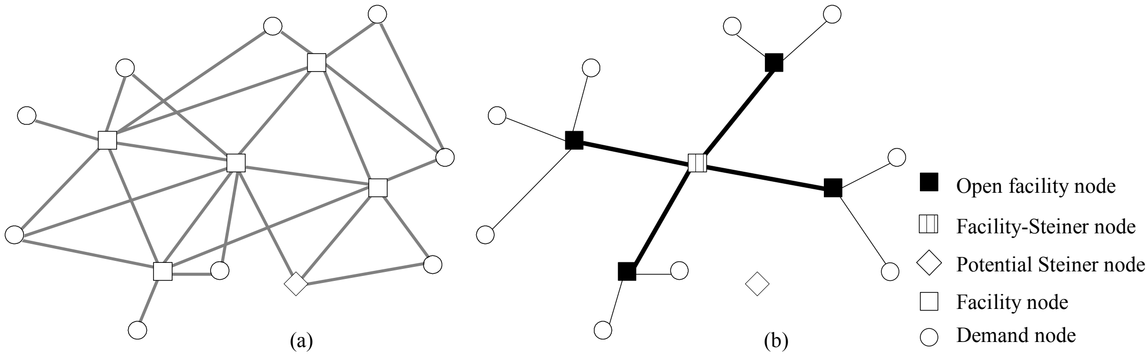

Figure 1: Connected Facility Location Example

Since the ConFL problem was initially introduced by Karger and Minkoff [4], it has received considerable attention from the computer science community, which has mainly focused on approximation algorithms. Gupta et al. [3] arrive to an algorithm with an approximation guarantee of 10.66 adapting a rounding technique of Shmoys et al. [8] on an integer formulation of the problem with an exponential number of constraints. Swamy and Kumar [9] use an integer programming formulation of the problem and the dual of its linear programming relaxation, and construct simultaneously an integer primal solution and a dual solution. They give an 8.55-approximation algorithm of the problem. Recently Eisenbrand et al. [2] presented a randomized algorithm that improves the approximation ratio to 4.00 that when derandomized degrades slightly to 4.23.

The operations research community has recently focused on the ConFL problem. Here the objective pivots around developing heuristics that behave efficiently in practice. Ljubic [6] introduces a heuristic that combines a variable neighborhood search method with a reactive tabu-search. In the same paper, she also proposes a branch-and-cut approach for solving the ConFL to provable optimality. For certain problem characteristics, the VNS approach finds solutions within 1% of the lower bound. However, VNS does not perform as efficiently for the entire set of test problems, with average gaps up to 10%. Tomazic and Ljubic [10] propose a greedy randomized adaptive search procedure (GRASP) algorithm, which they tested on randomly generated graphs with varying topologies and cost structures. On those instances, the heuristic performed fast and provided results whose average gap ranged up to 10%.

2 Transformation into a directed Steiner tree problem with a unit degree constraint

We first transform the ConFL into a directed Steiner tree problem with a unit degree constraint. To do so, we first split nodes into multiple nodes, such that each node may take only one role in the final solution; and then, we convert the resulting graph $G' = (V', E')$ into a directed graph with a root node.

In the definition of ConFL, the facility opening cost is only incurred when a demand node is assigned to it. Consequently, facility nodes may take two potential roles in the final solution. The role of active facility, if it serves a demand node; and the role of Steiner node, if it belongs to the tree T connecting open facilities but does not connect to any demand node. Consequently, we duplicate every node $i \in F$ and add it to the set of Steiner nodes S. We call the resulting set of Steiner nodes S', and the duplicate copy of a node $i \in F$ by $i' \in S'$. We create an edge of zero cost between node copies (i.e. $i \in F$ and $i' \in S'$) and add edges to the new node depending on the edges adjacent to node i as shown in Figure 2(a) and (b).

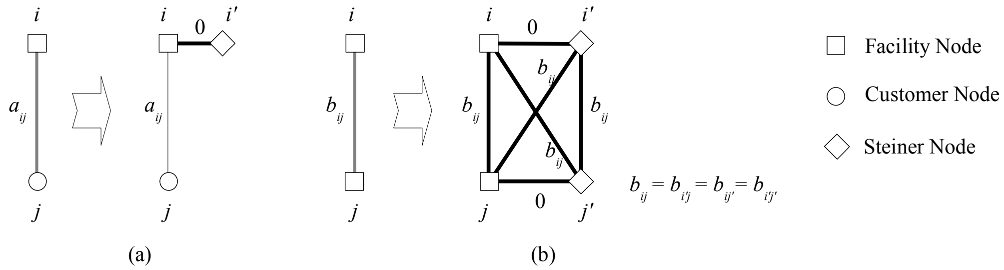

Figure 2: Connected Facility Location Transformations

Next, we construct a directed graph $G_A = (V, A)$ from the resulting graph $G' = (V', E')$. To do this, we replace every edge between nodes in $F \cup S'$ by two directed arcs with cost equal to the edge cost plus the opening cost of the end point, $\tilde{c}_{ij} = b_{ij} + f_j$, ($f_j = 0$ for nodes in S'). We also replace every assignment edge between facility nodes and demand nodes by one outcoming arc from the facility to the demand node with cost, $\tilde{c}_{ij} = a_{ij}$.

In addition, we create an artificial root node s and add a set of arcs from node s to every facility node $j \in F$ with cost equal to the facility opening cost. Finally, we impose a unit degree constraint on the root node to ensure that the open facilities are connected when the artificial node is removed from the solution. Note that in this directed graph, facility nodes have no weights since the opening costs have been added into the incoming arcs. It is easy to observe that the ConFL problem may be viewed as a unit degree Steiner tree problem in this transformed graph.

We now formulate the problem with flow variables to impose the connectivity requirements. For each node $i \in D$, we create a commodity i with the origin as the source node s and node i as the destination node. Let K be the set of commodities, and $D(k)$ denotes the destination of commodity k.

Directed flow formulation for the ConFL problem:

$$\text{Minimize} \sum_{(i,j)\in A} \tilde{c}_{ij} y_{ij} \tag{1a}$$

subject to

$$\sum_{(i,j)\in A} f_{ij}^k - \sum_{(j,l)\in A} f_{jl}^k = \begin{cases} -1, & \text{if } j = s; \\ 1, & \text{if } j = D(k); \forall j \in V \text{ and } k \in K \\ 0, & \text{otherwise}; \end{cases} \tag{1b}$$

$$f_{ij}^k \le y_{ij}, \quad \forall j \in F, k \in K \tag{1c}$$

$$\sum_{j\in F} y_{sj} = 1 \tag{1d}$$

$$f_{ij}^k \ge 0 \quad \forall (i,j) \in A, k \in K$$

$$y_{ij} \in \{0, 1\}$$

In this formulation, constraints (1b) require flow conservation at each node. Constraint (1c) requires that an arc is present if there is flow in a path. Finally, constraint (1d) imposes a unit degree constraint at the root node.

3 Heuristics

Our heuristic can be interpreted as a two-phase procedure. In the first phase we apply dual-ascent on the dual of a Lagrangian relaxation of formulation (1). We add constraint (1d) into the objective function with a Lagrange multiplier, which implies increasing the cost of the source node's outcoming arcs by a constant, μ. When we dualize constraint (1d), the resulting problem is a Steiner tree problem. We apply the dual-ascent procedure for the Steiner tree as described in Wong [11], Balakrishnan et al. [1] and refined in Raghavan [7]. This dual-ascent procedure gives both a lower bound, by approximately solving the dual to formulation (1); and a feasible primal solution, by exploiting the complementary slackness conditions together with the approximate dual solution. In other words dual-ascent yields a feasible solution for the dual and hence a lower bound to the optimal solution for the primal, and a feasible solution ($\mathcal{S}$, $\mathcal{F}$, T) and an upper bound for the ConFL problem. Since we have a unit degree constraint, if we set μ large enough the constraint is satisfied by the primal solution generated by the dual-ascent procedure.

In the second phase, to decrease the upper bound we implement a set of steps that reconstructs the tree on the set of Steiner nodes $\mathcal{S}$, open facilities $\mathcal{F}$ and demand nodes; closes certain open facilities; and reassigns demand nodes as needed. We divide the set of improvement steps into two groups: (1) sequential improvements that in Euclidean space never deteriorate the upper bound; and (2) local improvements that at each iteration strategically close one facility and compute the new solution savings or losses to make a decision on whether to permanently close the facility.

In the sequential improvements, we construct a minimum spanning tree T on the set of open facility nodes, $\mathcal{F}$, and Steiner nodes, $\mathcal{S}$. Next, we remove any Steiner node that has degree 2 or less and update the minimum spanning tree, T. We reiterate this process until all Steiner nodes have degree 3 or more.

Subsequently, in the local improvements we list open facility nodes first in order of node degree in the tree T and next by the number of assigned demand nodes. Then we move through the list at each iteration removing the next facility node from the solution, reassigning its demand nodes, restoring the tree on the remaining open facilities, and computing the change in the solution cost. If we observe a saving in the solution cost, the facility node is permanently closed and removed from $\mathcal{F}$; otherwise, the facility node is restored to the solution. We repeat this process for each facility in the list.

In the local improvements, the order in which open facilities are removed is critical. Ordering the nodes in increasing order of node degree in the Steiner tree and number of demand nodes assignments seeks to maximize the savings with each removal based on the two roles that a facility node plays in the final solution.

4 UFL Heuristic

We propose an UFL heuristic to evaluate the quality of the starting solution obtained from dual-ascent. In this procedure, we solve the uncapacitated facility location problem to optimality disregarding the connectivity requirements and then implement dual-ascent to obtain a Steiner tree on the set of open facilities. Later on, this solution is used as the starting solution in our improvement approaches. We compare the final solutions obtained by both heuristics.

5 Computational Results

Problem Characteristics

We generated a set of graphs in Euclidean space with various characteristics regarding number of demand nodes, facility nodes, potential Steiner nodes, facility opening costs and edge costs. Nodes are randomly located on a 100 x 100 square grid. The Euclidean distances rounded up to the next integer (to preserve triangle inequality) are used as edge lengths. Any demand node can be assigned to any facility node with assignment cost equal to the edge length. Edge costs between potential facility nodes and potential

$\|D\|$	$\|F\|$	Facility Opening Cost								
		5			15			25		
		DA	DH	UFL	DA	DH	UFL	DA	DH	UFL
10	90	1.54%	0.61%	11.91%	1.79%	0.25%	15.85%	0.66%	0.00%	12.89%
20	80	9.38%	1.98%	7.45%	10.88%	3.25%	7.28%	13.98%	3.70%	10.80%
30	70	7.86%	2.48%	5.95%	16.17%	3.37%	8.11%	13.75%	3.96%	8.03%
40	60	9.21%	2.89%	5.80%	11.13%	2.79%	7.07%	13.80%	4.22%	6.81%
50	50	6.14%	1.78%	4.09%	9.04%	2.44%	5.47%	9.47%	3.21%	5.12%
60	40	3.51%	1.14%	3.52%	4.01%	1.93%	6.43%	7.41%	2.61%	5.07%
70	30	2.12%	0.92%	2.04%	3.13%	1.23%	3.99%	5.25%	1.81%	3.97%
80	20	0.97%	0.39%	0.91%	1.58%	0.77%	1.47%	2.50%	0.99%	1.23%
90	10	0.18%	0.08%	0.10%	0.15%	0.04%	0.27%	0.30%	0.04%	0.38%

Table 1: Average Percentage Gaps - Set 1

Steiner nodes are equal to the edge length multiplied by the M factor. The M factor illustrates the difference in cost between assignment edges and edges in the tree T. No edges between demand nodes and potential Steiner nodes are created. The facility opening cost is the same for any facility node. In Set 1, the facility opening cost varies between 5 and 25 in steps of 10 while the M factor is fixed and equal to 3. Then in Set 2 the M factor varies with values 1, 3 and 7, while the facility opening cost is equal to 30. There are in addition 20 pure potential Steiner nodes in each instance. We generated 10 instances for each combination.

We coded our heuristics in Visual Studio 2005 (C++) and the uncapacitated facility location problem for the UFL heuristic using ILOG CPLEX 11.0. We conducted all runs on an AMD Athlon(tm) 62 X2 Dual, 2.61 GHz machine with 3.25GB of RAM.

Results

Table 1 and 2 summarize some of our computational results for dual ascent (DA), our dual based heuristic (DH) and the UFL heuristic (UFL) for Set 1 and Set 2, respectively. Clearly, dual ascent yields solutions with relatively tight optimality gaps for the extreme cases, with either a high proportion of demand nodes or facility nodes. However, as the proportion of one node type increases over the other, the average percentage gaps grow considerably reaching up to over 16%. On the other hand, the average percentage gaps from our dual based heuristic are always below 5%. In all cases the improvement steps achieve important reductions in the upper bounds. Out of 900 instances, the worst solution obtained by our dual based heuristic was within 8.40% from optimality.

Our dual based heuristic consistently found better solutions than the UFL heuristic indicating that the starting solution obtained from dual ascent is of higher quality than just solving an uncapacitated facility location problem to optimality. As we expected the UFL heuristic performs worse as the number of potential facilities increases since the choice space is larger.

Regarding the overall behavior of our dual based heuristic. We observe that average gaps follow a concave relation with the proportion of demand nodes and facility nodes. Instances with similar number of demand nodes and facility nodes show the higher gaps. However, the peak location varies with the facility opening cost and the M factor. Our results indicate that our heuristic's performance is stable to a wide range of parameters. Finally, to solve any instance our heuristic took less than 5 seconds (4.735 seconds in the worst case).

6 Conclusions

For our test problems the heuristic generated solutions that were on average within 3% from optimality, and in the worst case instances within 8.4% from optimality. This indicates that our heuristic generates high quality solutions on complete Euclidean networks. Our proposed heuristic also yields significantly

$\|D\|$	$\|F\|$	M factor								
		1			3			7		
		DA	DH	UFL	DA	DH	UFL	DA	DH	UFL
10	90	10.30%	3.25%	5.98%	0.00%	0.00%	15.14%	0.00%	0.00%	19.72%
20	80	12.21%	2.97%	4.38%	15.62%	3.74%	10.80%	0.94%	0.01%	9.47%
30	70	7.90%	2.48%	2.85%	14.13%	3.85%	7.33%	3.17%	1.44%	9.69%
40	60	7.18%	1.88%	3.01%	14.13%	3.95%	7.55%	8.22%	2.72%	12.05%
50	50	5.32%	1.61%	1.56%	10.08%	3.16%	5.16%	7.69%	3.20%	9.30%
60	40	4.78%	1.39%	1.94%	7.54%	2.93%	5.67%	8.39%	2.41%	9.77%
70	30	2.59%	0.84%	0.78%	5.44%	1.96%	3.50%	6.17%	2.43%	7.72%
80	20	1.56%	0.30%	0.28%	3.24%	1.08%	1.09%	4.43%	1.68%	3.68%
90	10	0.00%	0.00%	0.05%	0.09%	0.04%	0.33%	1.14%	0.42%	0.51%

Table 2: Average Percentage Gaps - Set 2

better solutions for the ConFL than the UFL heuristic. The starting solution obtained from dual-ascent has a significant impact on the quality of the final solution. This is especially relevant when the pool of potential facilities is large with respect to the number of customers.

References

[1] A. Balakrishnan, T. L. Magnanti, and R. T. Wong. A dual-ascent procedure for large-scale uncapacitated network design. *Operations Research*, 37(5):716–740, 1989.

[2] F. Eisenbrand, F. Grandoni, T. Rothvoß, and G. Schäfer. Approximating connected facility location problems via random facility sampling and core detouring. *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1174–1183, 2008.

[3] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi, and B. Yener. Provisioning a virtual private network: a network design problem for multicommodity flow. *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 389–398, 2001.

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

[5] C. Krick, H. Räcke, and M. Westermann. Approximation Algorithms for Data Management in Networks. *Theory of Computing Systems*, 36(5):497–519, 2003.

[6] I. Ljubic. A Hybrid VNS for Connected Facility Location. *Lecture Notes in Computer Science*, 4771:157–169, 2007.

[7] S. Raghavan. *Formulations and algorithms for network design problems with connectivity requirements*. PhD thesis, Massachusetts Institute of Technology, 1995.

[8] D. B. Shmoys, É. Tardos, and K. Aardal. Approximation algorithms for facility location problems (extended abstract). *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 265–274, 1997.

[9] C. Swamy and A. Kumar. Primal–Dual Algorithms for Connected Facility Location Problems. *Algorithmica*, 40(4):245–269, 2004.

[10] A. Tomazic and I. Ljubic. A GRASP Algorithm for the Connected Facility Location Problem. *International Symposium on Applications and the Internet*, pages 257–260, 2008.

[11] R. T. Wong. A dual ascent approach for steiner tree problems on a directed graph. *Mathematical Programming*, 28(3):271–287, 1984.