Variable Neighborhood Search for Capacitated Connected Facility Location

Markus Leitner and Günther R. Raidl

Institute of Computer Graphics and Algorithms
Vienna University of Technology, Vienna, Austria
{leitner|raidl}@ads.tuwien.ac.at
http://www.ads.tuwien.ac.at

1 Introduction

The Capacitated Connected Facility Location Problem (CConFL) is an $\mathcal{NP}$-hard combinatorial optimization problem which arises in the design of last mile communication networks (fiber-to-the-curb scenarios) [1]. Formally, we are given an undirected, weighted graph $G = (V, E)$, with edge costs $c_e \geq 0$, $\forall e \in E$. The node set $V = \{r\} \cup F \cup T$ is the disjoint union of the root node r, potential facility locations F, and possible Steiner nodes T. Each facility $i \in F$ has associated opening costs $f_i \geq 0$ and a maximum assignable capacity $D_i \in \mathbb{N}$. Furthermore, we are given a set of potential customers C, with individual capacity demands $d_k \in \mathbb{N}$ and prizes $p_k \geq 0$, $\forall k \in C$, the latter corresponding to the expected profit when supplying customer k. Each customer $k \in C$ may be assigned to one facility of a subset $F_k \subseteq F$, with assignment costs $a_{i,k} \geq 0$, $\forall i \in F_k$. A solution to CConFL $S = (R_S, T_S, F_S, C_S, \alpha_S)$ consists of a Steiner Tree (R_S, T_S), $R_S \subseteq V$, $T_S \subseteq E$, connecting the set of opened facilities $F_S \subseteq F$ and the root node r. $C_S \subseteq C$ is the set of customers feasibly (i.e. respecting the capacity constraints) assigned to facilities F_S, whereas the actual mapping between customers and facilities is described by $\alpha_S : C_S \to F_S$. The objective value of a feasible solution S is given by $c(S) = \sum_{e \in T_S} c_e + \sum_{i \in F_S} f_i + \sum_{k \in C_S} a_{\alpha_S(k),k} + \sum_{k \in C \setminus C_S} p_k$, and we aim to identify a most profitable solution minimizing this function. This variant of CConFL has already been tackled by exact methods based on mixed integer programming [2] and hybrid approaches based on Lagrangian relaxation [1]. Here, we present the first pure metaheuristic approach, which computes high quality solution faster than existing approaches.

2 Greedy Solution Construction

Given a partial – possibly empty – solution S, a score $\delta_i = \frac{(-f_i + \sum_{k \in C'_i} p_k - a_{i,k})}{\hat{c}_i}$ is computed for each facility $i \in F \setminus F_S$. By $C'_i \subseteq C \setminus C_S$ we denote the optimal set of customers still assignable to i, i.e. the set of customers maximizing the resulting profit which are not yet assigned to another facility in S. C'_i is computed using the Combo algorithm [3] and we further denote by $\hat{c}_i \geq 0$ the costs for connecting i to the partially constructed Steiner tree by means of an additional least-cost path. In each step, we add the facility with maximal score, connect it to the partially constructed Steiner tree, and assign the customers C'_i to it, as long as at least one facility i with $\delta_i > 1$ exists.

3 General Variable Neighborhood Search

In a variable neighborhood descent (VND) [4] we apply the following neighborhood structures to improve different aspects of a solution: Key-path improvement $[5, 1]$ to reduce the costs of the Steiner tree and customer swap $[6, 1]$ as well as single customer cyclic exchange [1] to optimize realized assignments. For improving the set of opened facilities, we consider a single facility swap neighborhood which adds or removes exactly one facility. A new opened facility is connected by an additional least-cost path while redundant edges are removed after closing a facility in case it has been a leaf of the Steiner tree.

In order to escape from local optima, we embed the VND in a variable neighborhood search (VNS) performing shaking by swapping $l = 2, \ldots, l_{\max}$ randomly chosen potential facility locations.

4 Greedy Randomized Adaptive Search Procedure

For comparison purposes we further embed above described VND into a greedy randomized adaptive search procedure (GRASP) [7] based on a randomized version of aforementioned constructive heuristic. Let $\delta_{\min} = \text{argmin}_{\delta_i}\{i \in F \setminus F_S \mid \delta_i > 1\}$ and $\delta_{\max}$ be the minimal and maximal scores among all relevant facilities, respectively. Instead of always adding the facility with maximal score, at each step we randomly choose one among the facilities $i \in F \setminus F_S$ for which $\delta_{\max} - \beta(\delta_{\max} - \delta_{\min}) \leq \delta_i \leq \delta_{\max}$ holds.

5 Preliminary Results

We performed computational experiments using the the instances from $[1, 2]$. VNS is terminated after ten consecutive non-improving iterations of the outermost largest shaking move of size $l_{\max} = \min\{|F|, 10\}$. For GRASP we set $\alpha = 0.2$ and generated 100 initial solutions and each experiment has been repeated 30 times on a single core of an Intel Core 2 Quad with 2.83GHz and 8GB RAM. For the VND, we apply the neighborhood structures in the same order as introduced above, but switch back to the first – i.e. the key-path – neighborhood after changing the set of opened facilities only. The single customer cyclic exchange neighborhood is searched using a next improvement strategy, while best improvement is applied for all other neighborhood structures.

Table 1 summarizes relative average objective values in percent, corresponding standard deviations, and relative median CPU times of the VNS and GRASP compared to the branch-and-cut-and-price approach (dBCP) from [2] which performed best among the previously presented methods. An absolute CPU-time limit of 7200 seconds has been applied to dBCP. We conclude, that both GRASP and VNS are able to compute solutions which are only slightly worse that those of dBCP, while usually needing much less CPU-time. For some instances with $|F| = 200$ and $|C| = 75$, which are particularly hard for dBCP, the obtained solutions are even significantly better than the ones obtained by dBCP within two hours. VNS seems to have small advantages over GRASP with respect to solution quality, while GRASP is usually slightly faster using the current settings. We are currently analyzing the usage of additional large neighborhood structures based on the mixed integer programming models from [2] and plan to include a more detailed computational study with additional results on further, larger instances in the full paper.

Table 1. Relative average solution values in %, corresponding standard deviations, and median CPU times. Instances have been grouped according to $|F|$ and $|C|$. Each experiment has been repeated 30 times for GRASP and VNS.

			average objective values				median CPU times	
$\|F\|$	$\|C\|$	#	$\frac{GRASP-dBCP}{dBCP}$	in %	$\frac{VNS-dBCP}{dBCP}$	in %	$\frac{GRASP}{dBCP}$	$\frac{VNS}{dBCP}$
75	75	12	5.02	(2.08)	3.90	(1.45)	0.09	0.17
100	100	12	4.09	(1.30)	3.45	(1.16)	0.23	0.27
200	200	12	4.07	(2.03)	3.25	(1.14)	1.79	1.81
75	200	12	1.86	(0.32)	0.95	(0.28)	0.58	0.59
200	75	12	-6.23	(23.59)	-5.62	(23.71)	0.01	0.01

References

1. Leitner, M., Raidl, G.R.: Combining Lagrangian decomposition with very large scale neighborhoood search for capacitated connected facility location. In: Post-Conference Book of the Eight Metaheuristics International Conference – MIC 2009. (to appear)
2. Leitner, M., Raidl, G.R.: Branch-and-cut-and-price for capacitated connected facility location. Technical Report TR 186–1–10–01, Vienna University of Technology, Vienna, Austria (2010)
3. Martello, S., Pisinger, D., Toth, P.: Dynamic programming and strong bounds for the 0–1 knapsack problem. Management Science **45**(3) (1999) 414–424
4. Hansen, P., Mladenović, N.: Variable neighborhood search: Principles and applications. European Journal of Operational Research **130**(3) (2001) 449–467
5. Voß, S.: Steiner's problem in graphs: heuristic methods. Discrete Applied Mathematics **40** (1992) 45–72
6. Contreras, I.A., Diaz, J.A.: Scatter search for the single source capacitated facility location problem. Annals of Operations Research **157**(1) (2008) 73–89
7. Feo, T., Resende, M.: Greedy randomized adaptive search procedures. Journal of Global Optimization **6**(2) (1995) 109–133