Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in DISCRETE MATHEMATICS

Electronic Notes in Discrete Mathematics 55 (2016) 21–24

www.elsevier.com/locate/endm

Heuristics for static cloudlet location

Alberto Ceselli, Marco Premoli [1]

Department of Computer Science, Università Degli Studi di Milano, Crema, Italy

Stefano Secci [2]

UPMC University Paris 06, UMR 7606, LIP6, F-75005, Paris, France

Abstract

Major interest is currently given to the integration of clusters of virtualization servers, also referred to as ‘cloudlets’, into the access network to allow higher performance and reliability in the access to mobile edge computing services. We tackle the facility location problem arising in the planning of these networks. Due to the complexity of the network topology, and the number of operational constraints, methods from the literature are hard to adapt. While in [1] we discussed the application issues, considering a real test case, in this paper we focus on the algorithmic ones, providing matheuristics solution algorithms for the static case, and an experimental insight on their computational behavior.

Keywords: telecommunications, facility location, matheuristics

Model. Let B be a set of access point (AP) locations. Let I, J and K be a set of sites where aggregation, core nodes and cloudlet facilities can be installed, resp.. Our static cloudlet location problem asks to design a two-level AP-aggregation-core network, to locate cloudlets on it, and to assign APs to cloudlets, minimizing installation costs, respecting cloudlet capacities and service level agreements on maximum delay and available bandwidth on paths between APs and cloudlets. We assume a superposition of stars topology: any AP is connected to a single aggregation node, and each aggregation node to a single core node, while a full mesh is built among cores. For each AP $s \in B$, let δ_s^u be the number of users connecting to s and δ_s^b their overall

[1] Email: {`alberto.ceselli, marco.premoli`}`@unimi.it`
[2] Email: `stefano.secci@upmc.fr`

http://dx.doi.org/10.1016/j.endm.2016.10.006

bandwidth consumption. Let l_i, m_j, c_k be the fixed cost for activating an aggregation node in $i \in I$, a core node in $j \in J$ and a cloudlet facility in $k \in K$, resp.. Let C denote the number of users that each cloudlet can serve. Let $d_{i,j}$ and $u_{i,j}$ be the length and bandwidth capacity of each link $(i,j) \in E = (B \times I) \cup (I \times J) \cup (J \times J)$. We assume low latency to be enforced by imposing both a maximum sum of links' length ($\bar{D}$) and number of hops ($\bar{H}$) in a path from AP to its cloudlet, and a maximum distance ($\bar{d}$) between connected nodes. We define as S^{sk} the set of paths from APs to cloudlets such that $\sum_{(i,j)\in p} d_{(i,j)} \leq \bar{D}$, $|p| \leq \bar{H}$ and $d_{(i,j)} \leq \bar{d}$ for all $(i,j) \in p\}$, with $|p|$ denoting the number of links forming path p. We introduce three sets of variables. The first corresponds to binary location variables: x_i, y_j and z_k take value 1 if sites $i \in I$, $j \in J$ and $k \in K$, resp., are selected to host facilities. The second corresponds to binary routing variables: $r_p^{s,k}$ take value 1 if users in AP $s \in B$ are served by a cloudlet in $k \in K$, and the corresponding traffic is routed along path $p \in \bar{S}^{sk}$. The third corresponds to network topology binary variables: $t_{s,i}$, $w_{i,j}$ and $o_{m,n}$ take value 1 if a link is established between an AP s and an aggregation node i, an aggregation node i and a core node j, two core nodes m and n, resp.. Moreover, let $U \in [0,1]$, represent the maximum allowed link utilization ratio. We formulate our problem as follows.

$$\min \sum_{i\in I} l_i x_i + \sum_{j\in J} m_j y_j + \sum_{k\in K} c_k z_k \tag{1}$$

$$\text{s.t.} \sum_{p\in S^{sk}|i\in p} r_p^{s,k} \leq x_i \ , \ \forall s \in B, \forall k \in K, \forall i \in I \tag{2}$$

$$\sum_{p\in S^{sk}|j\in p} r_p^{s,k} \leq y_j \ , \ \forall s \in B, \forall k \in K, \forall j \in J \tag{3}$$

$$\sum_{p\in S^{sk}} r_p^{s,k} \leq z_k \ , \ \forall s \in B, \forall k \in K \tag{4}$$

$$\sum_{k\in K} \sum_{p\in S^{s,k}} r_p^{s,k} = 1 \ , \ \forall s \in B \tag{5}$$

$$\sum_{s\in B} \sum_{p\in S^{s,k}} \delta_s^u r_p^{s,k} \leq C z_k \ , \ \forall k \in K \tag{6}$$

$$\sum_{s\in B} \sum_{k\in K} \sum_{\substack{p\in S^{s,k} \\ |(i,j)\in p}} \delta_s^b r_p^{s,k} \leq u_{(i,j)} U (w_{i,j} + o_{i,j} + t_{i,j}) \ , \ \forall (i,j) \in E \tag{7}$$

We minimize installation costs (1); (2)-(4) impose that no path can be selected unless devices are installed on its sites; (5)-(7) ensure that each AP is assigned to a cloudlet; (6) impose that active cloudlets serve at most C users; (7) are link utilization constraints. Moreover a set of topology constraints need to be imposed, that are omitted here for the sake of brevity.

Algorithms. We devised matheuristics that consist of five phases: (i) clustering of the APs in B, aggregating their demands in *centers* (ii) dynamic generation of the center-cloudlet path variables $r_p^{s,k}$ (iii) retrieval of a feasible solution with a hierarchical rounding and pricing process (iv) refinement of the solution with local branching (v) restart.

During phase (i) we create $|B|/\alpha$ clusters of APs by selecting *centers*. To ensure feasibility we enforce that no AP is placed in a cluster if its distance from the center is greater than $\bar{d}$, and the distance between two centers is computed as the maximum distance between one center and each of the APs of the other cluster. To initialize the clustering we use a simplified model in which cloudlets, aggregation and core nodes coincide. Therefore, a routing path is always a direct link, the resolution process needs to find only cloudlet locations, and only capacity constraints need to be enforced.

During phase (ii), as the cardinality of feasible paths sets S^{sk} grows combinatorially, we perform column generation on the set of variables $r_p^{s,k}$. The pricing problem is a resource constrained shortest path problem on an acyclic network, that we solve in pseudo polynomial time by dynamic programming.

At the end of the column generation process (phase iii) we start rounding by selecting the location variable with highest fractional value, fix it to one, and propagate that fixing. If the solution is still fractional, we resume column generation to restore optimality, and we repeat the rounding and propagation process. If infeasibility is detected we backtrack, fixing the last rounding variable to zero, and column generation is resumed. If infeasibility is obtained also in this way, we stop in a FAIL status. Whenever a feasible integer solution is achieved, instead, we stop in a SUCCESS status. Instead of choosing an arbitrary location variable for rounding, we consider in sequence variables z_k, y_j, x_i and $r_p^{s,k}$. Variables related to the topology are never rounded explicitly: in case of SUCCESS, a small MILP problem remains to fix them. In case of FAIL, instead, the solution produced in phase (i) is considered. That is, in any case a feasible solution $\hat{S}$ is obtained after phase (iii), unless the instance itself is infeasible.

During phase (iv) we try to improve the feasible integer solution $\hat{S}$ with an ILP-based very large scale neighborhood search strategy, exploring a κ-*OPT neighborhood*: we consider the restricted model produced by the last column generation round, and we include the following local-branching con-

Table 1
Results on 100 nodes instances

		$\alpha = 2$			$\alpha = 3$			$\alpha 2 \to 3$	
	z^{init}	t	z^*	Δ z	t	z^*	Δ z	Δ z	Δ t
μ	12.65	9055.70	11.37	10.01%	320.50	11.85	6.21%	-4.43%	96.35%
σ	0.74	1884.33	0.70	5.26	75.80	0.64	4.33	5.41	0.95
max	13.32	11984.00	12.92	18.84%	489.00	13.12	10.73%	-11.10%	97.27%

straint: $\sum_{k \in K | \bar{z}_k = 1}(1 - z_k) + \sum_{k \in K | \bar{z}_k = 0} z_k \leq \lceil \kappa \cdot \sum_{k \in K} \bar{z}_k \rceil$ where parameters $\bar{z}_k$ represent the values of the variables z_k in $\hat{S}$, and parameter κ represents the fraction of z_k variables whose values are allowed to flip with respect to the current solution. We solve this restricted model with a general purpose ILP Solver, setting a limit τ on the execution time. As a restart strategy (phase v) we update the clustering and iterate steps (ii)–(iv). The information given by the fractional solution found at the end of phase (ii) is used to perform such an update: when a center i is fully associated to a cloudlet k through a single path, the two clusters represented by i and k are joined and a new representative is found by aggregating them; otherwise if center i is fractionally associated through multiple paths to different cloudlets, then the corresponding cluster is split, trying to improve a suitable *connectivity measure* that we devised, and whose formal definition is omitted for brevity. A fixed number of restarts are performed, and the best solution found is retained.

Computational results. We implemented our algorithms in C++, using CPLEX 12.6 to solve both LP and MILP problems. Our tests ran on an Intel Core 2 Duo 3 GHz workstation with 2 GB of RAM. Parameters are set as in [1]. We considered a dataset adapted from capacitated p-median instances from the literature. Table 1 reports an overview of results on instances with 100 nodes. We first report the value of the solution found by the initial clustering heuristics (z^{init}). Then we indicate average computing time, value of the solutions and gap with respect to the initial solutions, comparing two settings: clustering with $\alpha = 2$ (second block) and $\alpha = 3$ (third block). In the last block we summarize the effect of moving from $\alpha = 2$ to $\alpha = 3$. Our matheuristics lead to an average improvement of the initial solution of $\sim 10\%$ and $\sim 6\%$, resp.. We also note that using fewer clusters leads to major savings in CPU time ($\sim 96\%$) with a mild quality worsening ($\sim 4\%$).

References

[1] A. Ceselli, M. Premoli, S. Secci, "Cloudlet Network Design Optimization", in *Proc. of IFIP Networking 2015*, 20-25 May, 2015, Toulouse, France.