Conf-9508113--1

LA-UR 95-1293

TITLE: COMPACT LOCATION PROBLEMS WITH BUDGET AND COMMUNICATION CONSTRAINTS

AUTHOR(S): S. O. Krumke, H. Noltemeier, S. S. Ravi, M. V. Marathe

SUBMITTED TO: First Annual International Computing and Combinatorics Conference
Xian, China
August 24-24, 1995

Los Alamos

Los Alamos National Laboratory
Los Alamos New Mexico 87545

Compact Location Problems with Budget and Communication Constraints
(Extended Abstract)

S.O. Krumke[1] H. Noltemeier[1] S.S. Ravi[2] M. V. Marathe[3]

March 31, 1995

Abstract

We consider the problem of placing a specified number p of facilities on the nodes of a given network with two nonnegative edge-weight functions so as to minimize the diameter of the placement with respect to the first distance function under diameter- or sum-constraints with respect to the second weight function.

Define an (α, β)-approximation algorithm as a polynomial-time algorithm that produces a solution within α times the optimal function value, violating the constraint with respect to the second distance function by a factor of at most β.

We observe that in general obtaining an (α, β)-approximation for any fixed $\alpha, \beta \geq 1$ is $\mathcal{NP}$-hard for any of these problems. We present efficient approximation algorithms for the case, when both edge-weight functions obey the triangle inequality.

For the problem of minimizing the diameter under a diameter constraint with respect to the second weight-function, we provide a $(2,2)$-approximation algorithm. We also show that no polynomial time algorithm can provide an $(\alpha, 2-\varepsilon)$- or $(2-\varepsilon, \beta)$-approximation for any fixed $\varepsilon > 0$ and $\alpha, \beta \geq 1$, unless $\mathcal{P} = \mathcal{NP}$. This result is proved to remain true, even if one fixes $\varepsilon' > 0$ and allows the algorithm to place only $2p/|V|^{1/6-\varepsilon'}$ facilities.

Our techniques can be extended to the case, when either the objective or the constraint is of sum-type and also to handle additional weights on the nodes of the graph.

[1]Department of Computer Science, University of Würzburg – Am Hubland, 97074 Würzburg, Germany. Email:{krumke,noltemei}@informatik.uni-wuerzburg.de

[2]Dept. of Computer Science, University at Albany - SUNY, Albany, NY 12222. Email:ravi@cs.albany.edu

[3]Los Alamos National Laboratory P.O. Box 1663, MS M986, Los Alamos NM 87545. Email: madhav@c3.lanl.gov. Research supported by the Department of Energy under Contract W-7405-ENG-36.

Here our heuristics provide performance guarantees of $(2 - 2/p, 2)$ and $(2, 2 - 2/p)$ respectively.

Keywords: Approximation algorithms, Bicriteria problems, Network design, Location Theory and Combinatorial algorithms.

1 Introduction and Basic Definitions

Several fundamental problems in location theory can be modeled as finding a placement obeying certain covering constraints. The goal in such location theory problems is usually to minimize a certain measure of cost associated with the placement. The cost may reflect the price of placing the network, or it may reflect the maximum communication cost between the two facilities. Examples of such cost measures are the total edge cost and the diameter respectively. Consider for instance a computational task consisting of a number of communicating subtasks. At a given time, some of the processors may be already allocated and the remaining processors are available. The problem is to select a subset of processors from the currently available processors, one per subtask, such that the cost of communication among the processors executing the subtasks is minimized.In this application, the processors must be allocated quickly, and this may conflict with the goal of minimum communication cost among the selected processors. Compact location problems also arise in a number of other applications such as allocation of manufacturing sites for the components of a system so as to minimize the cost of transporting components, distributing the activities of a project among geographically dispersed offices so as to minimize the transportation and communication costs among the offices, etc.

Finding a placement of facilities of sufficient generality minimizing even one of these measures is often $\mathcal{NP}$-hard [GJ79]. Moreover, in applications that arise in real-life situations, it is often the case that the network to be built is required to minimize more than one cost measure simultaneously. In this paper, we consider bicriteria problems motivated by practical instances arising in the location theory.

The problems we consider in this paper can be termed as *Compact Location* problems; since we will typically be interested in finding a "compact" placement of facilities. Consider, the problem of placing a specified number p of facilities on the nodes of a given complete network $G = (V, E_c)$ so as to minimize some measure of the distances between the facilities. This problem has been studied for both diameter and sum objectives (see e.g. [RKM+93]). The problems have applications in statistical clustering, pattern recognition, processor allocation and load-balancing.

In this paper we consider extensions of these problems, where we are given *two* weight-functions δ_c, δ_d on the edges of the network. The first of the functions, δ_c, will represent the cost of constructing an edge, while the second one, δ_d stands for the actual transportation- or communication-cost over an edge (once it has been constructed).

A general bicriteria problem, $(\mathcal{A}, \mathcal{B})$, is defined by identifying two minimization objectives of interest from a set of possible objectives. The problem specifies a budget value on the first objective, $\mathcal{A}$, and seeks to find a network having minimum possible value for the second objective, B, such that this solution obeys the budget on the first objective. An (α,β)-approximation algorithm is defined as a polynomial-time algorithm that produces a solution in which the first objective value is at most α times the budget, and the second objective value is at most β times the minimum for any solution obeying the budget on the first objective. As an example, consider the following *diameter-bounded minimum diameter problem* or (Diameter, Total cost) bicriteria problem: given an undirected graph $G = (V, E)$ with two different integral nonnegative weights f_e (modeling the cost) and g_e (modeling the delay) for each edge $e \in E$, a integer p denoting the number of facilities to be placed and an integral bound B (on the total delay), find a placement of p facilities with minimum diameter under the f–cost such that the diameter of the placement under the g–costs (the maximum delay between any pair of nodes) is at most B.

Let $G = (V, E_c)$ be a complete undirected graph with $n := |V|$ nodes and let $2 \leq p \leq n$ be the number of facilities to be placed. We call a subset $P \subseteq V$ of cardinality p a *placement*. Given a nonnegative weight– or cost–function $\delta : E_c \to \mathbb{Q}_+$, we will use $\mathcal{D}_\delta(P)$ to denote the *diameter* of a placement P with respect to δ, that is

$$\mathcal{D}_\delta(P) = \max_{\substack{u,v \in P \\ u \neq v}} \delta(u, v).$$

Similarly, we will let $\mathcal{S}_\delta(P)$ stand for the *sum of the distances* between facilities in the placement P, i.e.

$$\mathcal{S}_\delta(P) = \sum_{\substack{u,v \in P \\ u \neq v}} \delta(u, v).$$

We note that the average length of an edge in a placement P equals $\frac{2}{p(p-1)}\mathcal{S}_\delta(P)$.

As usual, we say that a nonnegative distance δ on the edges of G satisfies the *triangle inequality*, if we have

$$\delta(v, w) \leq \delta(v, u) + \delta(u, w)$$

for all $v, w, u \in V$.

The Minimum *Diameter Placement Problem*, or MDP for short, is to find a placement of minimum diameter. Similarly, for the *Minimum Average Placement Problem*, also denoted MAP, one wants to find a placement P such that $\mathcal{S}_\delta$ is minimized. Both problems are known to be $\mathcal{NP}$-hard, even when the distance δ obeys the triangle inequality ([RKM+93]). Moreover, if the distances are not required to satisfy the triangle inequality, then there can be no polynomial time relative approximation algorithm for MDP or MAP unless $\mathcal{P} = \mathcal{NP}$.

In the sequel we will restrict ourselves to those instances of the problems, where the weights on the edges obey the triangle inequality. Given a problem PROB, we will use PROB-TI to denote the corresponding subset of instances, where the triangle inequality is satisfied.

We will be mainly concerned with problems, where the objective is to minimize either the diameter, although our results can be extended to sum objectives as well, as we will indicate briefly in the last section. The constraints with respect to the second weight function are either of sum or diameter type.

A constraint with respect to δ_c of the form

$$\mathcal{S}_{\delta_c}(P) \leq \Omega \quad \text{or} \quad \mathcal{D}_{\delta_c}(P) \leq \Omega$$

will be called *budget constraint*. Here, we are given a budget Ω and the aim is to find a best-possible placement P that does not involve total building costs of more than Ω (if we consider the sum $\mathcal{S}_{\delta_c}(P)$) or that does not involve a maximum building cost of an edge of more than Ω (in the case that we look at $\mathcal{D}_{\delta_c}$) respectively.

Similarly, we call a diameter or sum constraint with respect to δ_d a *communication constraint*. Here, we face the problem of finding a placement, which is as cheap as possible and where we want to have control over the diameter or sum of the distances.

Assume that we are given the problem of finding a placement P of p nodes, minimizing $\mathcal{M}(P)$ subject to the constraint

$$\mathcal{M}'(P) \leq \Omega, \tag{1}$$

where the constraint (1) is either of budget or communication type, and $\mathcal{M}$ stands for either the sum of the distances or the diameter of the placement with respect to the second weight function. Then we define an (α, β)*-approximation algorithm* to be a polynomial-time algorithm, which for any instance I does one of the following:

(a) It produces a solution within α times the optimal function value, violating the constraint with respect to the second distance function by a factor of at most β

(b) It returns the information that no feasible placement exists at all.

Notice that, if there is no feasible placement but a placement violating the constraint by a factor of at most β, an (α, β)-approximation algorithm has the choice of performing either action (a) or (b).

Following [HS86], the *bottleneck graph* bottleneck(G, δ, Δ) of $G = (V, E_c)$ with respect to δ and a bound Δ is defined by

$$\text{bottleneck}(G, \delta, \Delta) := (V, E'), \text{ where } E' := \{e \in E_c : \delta(e) \leq \Delta\}.$$

2 Related Work

2.1 Minimizing one cost measure

In contrast to the $\mathcal{NP}$-hardness results contained in Section 1, which hold for general distance matrices, geometric versions of MDP and MVP were shown to be solvable in polynomial time in [AIKS91]. In the geometric versions of these problems, the nodes are points in space and the distance between a pair of nodes is their Euclidean distance. For points in the plane, [AIKS91] contains an $O(p^{2.5}n \log p + n \log n)$ algorithm for the MDP problem and an $O(p^2 n \log n)$ algorithm for the MVP problem, and it is observed that these algorithms extend to higher dimensions. These algorithms are based on the construction of p^{th} order Voronoi diagrams [Lee82, PS85].

Other work has addressed placement problems where the objective functions are different from the above. For example, the traditional facility location problems are concerned with minimizing the maximum distance from a node to a nearest facility (p-center problem) or minimizing the sum of the distances from each node to the nearest facility (p-median problem) [HM79, MF90]. However, for the problems considered in this paper, the objective functions involve only the distances between facilities. Similarly, the clustering problems considered in the literature [HS86, FG88, Gon85] involve partitioning the given set of nodes into clusters so as to minimize a given objective function. The location problems considered in this paper are of a different flavor since the objective functions involve only a subset of nodes. Facility location problems where the objective is to place facilities so as to *maximize* some function of the distances between facilities have been considered in the literature; see for example [EN89, RRT91]. Problems in which the placement of facilities is not restricted to the nodes of the network have also been studied [MF90]. We consider only location problems in which the facilities are placed at the nodes of the network.

Hochbaum and Shmoys [HS86] and Dyer and Frieze [DF85] consider the node weighted versions of center problem. Location problems with other optimizing criteria have also been considered in the literature. Lin and Vitter [LV92] provide approximations for the s-median problem where s median nodes must be chosen so as to minimize the sum of the distances from each node to its nearest median. The solution method is approximate in terms of both the number of median-nodes used and the sum of the distances from each node to the nearest median. Bar-Ilan and Peleg [BIP90] consider the balanced center problem. They provide approximation algorithms for problem of allocating network centers wherein each center is allowed to service only a bounded number of nodes.

2.2 Bicriteria approximations

While there has been much work on finding minimum-cost networks for each of the cost measures that we simultaneously minimize, there has been relatively little work on approximations for multi-objective network-design. In this direction, Bar-Ilan, Kortsarz and and Peleg [BIP90] considered balanced versions of the problem of assigning network centers, where a bound is imposed on the number of nodes that any center can service. They extended existing approximation algorithms for center problems to the balanced versions. Wong [Won80] examined a budget network design problem in which a network is to be built whose cost is at most a certain budget such that the sum of the path-lengths of commodities to be routed using this network is minimized. He showed that even finding near-optimal solutions is $\mathcal{NP}$-hard if we are required to conform to the budget requirement and approximate only the sum of path-lengths. Lin and Vitter [LV92] provided approximations for the s-median problem where s median nodes must be chosen so as to minimize the sum of the distances from each vertex to its nearest median. The solution output is approximate in terms of both the number of median-nodes used and the sum of the distances from each vertex to the nearest median. Iwainsky et al. [ICTV86] formulated a version of the minimum-cost Steiner problem with an additional cost based on node-degrees. Given costs and lengths on the edges of an undirected graph, Warburton [War87] presented a fully polynomial approximation scheme (FPAS) for the problem of constructing a path of shortest length between two specified nodes subject to a given constraint on the cost of the path. Hassin [Has92] presented a faster FPAS for the same problem.

Other researchers have addressed multi-objective approximation algorithms for problems arising in areas other than network design. This includes research in the areas of computational geometry [AFMP94], numerical analysis, network design [ABP90, KRY93, Fis93] and scheduling [ST93].

3 Finding Placements with Small Diameter Under Constraints

First we will consider the problem, where the objective is to minimize the maximum construction cost of an edge subject to communication constraints of bottleneck type.

Definition 3.1 (MDP with Bottleneck Constraints (MDP$_{bott}$))
Input: *An undirected complete graph $G = (V, E_c)$ with two nonnegative weight functions $\delta_c, \delta_d : E_c \to \mathbf{Q}_+$, an integer $2 \leq p \leq n$ and a number $\Omega \in \mathbf{Q}_+$.*

Output: *A set $P \subseteq V$, $|P| = p$, minimizing the objective*

$$\mathcal{D}_{\delta_c}(P) = \max_{\substack{v,w \in P \\ v \neq w}} \delta_c(v,w)$$

subject to the constraint

$$\mathcal{D}_{\delta_d}(P) = \max_{\substack{v,w \in P \\ v \neq w}} \delta_d(v,w) \leq \Omega$$

The interpretation of the model is as follows: We have a measure Ω of what we want to have a "guaranteed response time" between two facilities. Now the goal is to find a placement subject to that constraint minimizing the maximum building cost of a link.

If one is given an instance I of MDP$_{bott}$–TI, the first question that arises is, whether there is *any* feasible placement, i.e. whether there exists a placement P satisfying the constraint $\mathcal{D}_{\delta_d}(P) \leq \Omega$. Unfortunately, this question turns out to be hard (cf. [RKM+93]). Indeed, we have the following stronger hardness result:

Proposition 3.2 *Let $\varepsilon > 0$ and $\varepsilon' > 0$ be arbitrary. Suppose that* A *is a polynomial time algorithm that, given any instance of* MDP$_{bott}$*–TI, either returns a subset $S \subseteq V$ of at least $\frac{2p}{|V|^{1/6-\varepsilon'}}$ nodes satisfying $\mathcal{D}_{\delta_d}(S) \leq (2-\varepsilon)\Omega$, or provides the information that no placement of p nodes having communication diameter of at most Ω does exist. Then $\mathcal{P} = \mathcal{NP}$.*

Proof: See the appendix. □

We can swap the roles of δ_c' and δ_d' in the proof of the last proposition to show that the optimal value of the problem can not be approximated by a factor of $(2-\varepsilon)$. Moreover, replacing 2 by a suitable function $f \in \Theta(2^{\mathrm{poly}(|V|)})$, which given an input length of $\Omega(|V|)$ is polynomial time computable, it is easy to see that, if the triangle inequality is not required to hold, there can be no polynomial time approximation with performance ratio $O(2^{\mathrm{poly}(|V|)})$ for neither the optimal function value nor the constraint (modulo $\mathcal{P} = \mathcal{NP}$). Thus we obtain:

Lemma 3.3 *Unless $\mathcal{P} = \mathcal{NP}$, for any fixed $\varepsilon > 0$ and $\varepsilon' > 0$ there can be no polynomial time approximation algorithm for* MDP$_{bott}$*–TI that is required to place at least $2p/|V|^{1/6-\varepsilon'}$ facilities and has a performance guarantee of $(\alpha, 2-\varepsilon)$ or $(2-\varepsilon, \beta)$. If the triangle inequality is not required to hold, then the existence of an $(f(|V|), g(|V|))$-approximation algorithm for any $f, g \in O(2^{\mathrm{poly}(|V|)})$ implies that $\mathcal{P} = \mathcal{NP}$.*

We now present a heuristic Heur–MDP–MAP$_{bott}$ with a $(2,2)$–performance guarantee. In the sense of lemma 3.3, this is the best approximation we can expect to obtain in polynomial time. The heuristic is quite simple. The details of the algorithm are shown in Figure 1.

PROCEDURE Heur–MDP–MAP$_{bott}$($\mathcal{M}$)

1 G^v := bottleneck(G, δ_d, Ω)

2 $V_{cand} := \{v \in G^v : \deg(v) \geq p-1\}$

3 IF $V_{cand} = \emptyset$ THEN RETURN "certificate of failure"

4 Let $best := +\infty$

5 Let $P_{best} := \emptyset$

6 FOR each $v \in V_{cand}$ DO

(a) Let $N(v)$ be the set of $p-1$ nearest neighbors of v in G with respect to δ_c

(b) Let $P(v) := N(v) \cup \{v\}$

(c) IF $\mathcal{M}_{\delta_c}(P(v)) < best$ THEN $P_{best} := P(v)$
$best := \mathcal{M}_{\delta_c}(P(v))$

7 OUTPUT P_{best}

Figure 1: Details of the heuristic for MDP$_{bott}$ and MAP$_{bott}$

Theorem 3.4 *Let I be any instance of of* MDP_{both}*–TI such that an optimal solution P^* of cost diameter $OPT(I) = \mathcal{D}_{\delta_c}(P^*)$ exists. Then the algorithm* Heur–MDP–MAP$_{both}$*, called with $\mathcal{M}_{\delta_d} := \mathcal{D}_{\delta_d}$, returns a placement P satisfying $\mathcal{D}_{\delta_d}(P) \leq 2\Omega$ and $\mathcal{D}_{\delta_c}(P)/OPT(I) \leq 2$.*

Proof: See the appendix. □

We now turn to the case, when the objective is to minimize the distance diameter $\mathcal{D}_{\delta_d}$ subject to budget–constraints of sum type. Here, we want to find a placement, that is as "compact as possible", while the total construction costs of all the edges between the facilities in the placement do not exceed a given budget Ω.

Definition 3.5 (MDP with Sum–Budget–Constraints (MDP_{sum}))
Input: *An undirected complete graph $G = (V, E_c)$ with two nonnegative weight functions $\delta_c, \delta_d : E_c \to \mathbb{Q}_+$, an integer $2 \leq p \leq n$ and a number $\Omega \in \mathbb{Q}_+$.*
Output: *A set $P \subseteq V$, $|P| = p$ minimizing the objective*

$$\mathcal{D}_{\delta_d}(P) = \max_{\substack{v,w \in P \\ v \neq w}} \delta_d(v, w)$$

and satisfying the budget–constraint

$$S_{\delta_c}(P) = \sum_{\substack{v_i, v_j \in P \\ v_i \neq v_j}} \delta_c(v_i, v_j) \leq \Omega.$$

Again, it is not an easy task to find a placement P satisfying the budget–constraint or providing the information that there is no such placement respectively. Using a reduction from CLIQUE one obtains:

Proposition 3.6 *If the distances δ_c, δ_d are not required to satisfy the triangle inequality, there can be no polynomial time (α, β)–approximation algorithm for* MDP_{sum} *for any fixed $\alpha, \beta \geq 1$, unless $\mathcal{P} = \mathcal{NP}$. Moreover, if there is a polynomial time $(\alpha, 1)$–approximation algorithm for* MDP_{sum}*–TI for any fixed $\alpha \geq 1$, then $\mathcal{P} = \mathcal{NP}$.*

The details of the proof are deferred to the appendix. We proceed to present a heuristic for MDP_{sum}–TI. The main procedure shown in Figure 2 uses the test procedure from Figure 3. We have the following simple

Lemma 3.7 *Assume I is an instance of* MDP_{sum}*–TI such than there is an optimal placement P^*. If the test procedure* test(G_i, δ_c, Ω) *returns a "certificate of failure", then we have $OPT(I) > \delta_d(e_i)$.*

Procedure Heur–Generic

1 Sort the edges of G in ascending order with respect to δ_d

2 Assume now that $\delta_d(e_1) \le \delta_d(e_2) \le \cdots \le \delta_d(e_{\binom{n}{2}})$

3 Let P_{best} := "certificate of failure"

4 $i := 1$

5 **Do**

(a) $G_i :=$ bottleneck$(G, \delta_d, \delta_d(e_i))$

(b) $P_{best} :=$ test$(G_i, \delta_c|_{G_i}, \Omega)$

(c) $i := i + 1$

6 **Until** $P_{best} \ne$ "certificate of failure"

7 **Output** P_{best}

Figure 2: Generic bottleneck procedure

PROCEDURE **test**(G, δ, Ω)

1 $V_{cand} := \{v \in G : \deg(v) \geq p - 1\}$

2 **IF** $V_{cand} = \emptyset$ **THEN RETURN** "certificate of failure"

3 **Let** $best := +\infty$

4 **Let** $P_{best} := \emptyset$

5 **FOR each** $v \in V_{cand}$ **DO**

   (a) **Let** $N(v)$ **be the set of** $p - 1$ **nearest neighbors of** v **in** G **with respect to** δ

   (b) **Let** $P(v) := N(v) \cup \{v\}$

   (c) **IF** $S_\delta(P(v)) < best$ **THEN** $P_{best} := P(v)$
   $best := S_\delta(P(v))$

6 **IF** $best > (2 - 2/p)\Omega$ **THEN RETURN** "certificate of failure"
   ELSE RETURN P_{best}

Figure 3: Test Procedure used for Heur–MDP_{sum}

Now we can establish the result about the performance guarantee of the heuristic:

Theorem 3.8 *Let I denote any instance of* MDP$_{sum}$*–TI and assume that there is an optimal placement P^* of diameter $OPT(I) = \mathcal{R}_{\delta_d}(P^*)$. Then the generic bottleneck procedure* Heur-Generic *with the test procedure* test *returns a placement P with $S_{\delta_c}(P) \leq (2-2/p)\Omega$ and $\mathcal{D}_{\delta_1}(I)/OPT(I) \leq 2$.*

Proof: See the appendix. □

The techniques in the heuristic for MDP$_{sum}$ can be used to obtain a heuristic for an extended version of the problem, where we are given additional weights on the *nodes* on the graph G, representing construction costs of a facility at a specific node. The budget–constraint here is that the total sum of the construction costs of the edges plus the sum of the cost of the nodes should not exceed a given budget Ω. We formalize the problem in the following definition:

Definition 3.9 (MDP$_{sum}$ with Node Weights (MDP$^{node}_{sum}$))
Input: *An undirected complete graph $G = (V, E_c)$, two edge–weight function $\delta_c, \delta_d : E_c \to \mathbf{Q}_+$, a node weight function $\omega : V \to \mathbf{Q}_+$, an integer $2 \leq p \leq n$ and a number $\Omega \in \mathbf{Q}_+$.*
Output: *A set $P \subseteq V$, $|P| = p$ minimizing the objective*

$$\mathcal{D}_{\delta_d}(P) = \max_{\substack{v,w \in P \\ v \neq w}} \delta_d(v, w)$$

such that

$$\sum_{v \in P} \omega(v) + \sum_{\substack{v,w \in P \\ v \neq w}} \delta_c(v.w) \leq \Omega.$$

To obtain an approximate solution for MDP$^{node}_{sum}$, we transform a given instance I of MDP$^{node}_{sum}$ into a suitable instance I' of MDP$_{sum}$ according to the following

Lemma 3.10 *Given any instance I of* MDP$_{node}$*–TI, we can construct an instance I' of* MDP$_{sum}$*–TI in polynomial time with the following property: Given a placement for the instance I' with diameter D and costs C, the same placement has diameter D and costs C with respect to the instance I.*

Proof: Given the instance I we define an instance I' of MDP$_{sum}$–TI as follows. We let $G' = G$, $\delta'_d := \delta_d$, $p' := p$ and define the distance function $\delta'_c : E \to \mathbf{Q}_+$ by

$$\delta'_c(v, w) := \delta_c(v, w) + \frac{1}{2p}(\omega(v) + \omega(w)).$$

It is easy to check that the triangle inequality is satisfied for δ_c'. Moreover, by a straightforward calculation, for any placement P of p nodes

$$\sum_{\substack{u,v\in P\\ u\neq v}} \delta_c'(u,v) = \sum_{v\in P} \omega(v) + \sum_{\substack{u,v\in P\\ u\neq v}} \delta_c(u,v).$$

□

Thus using the heuristic for MDP_{sum} for the instance I' we can immediately obtain a $(2, 2-2/p)$-approximation for MDP^{node}_{sum}. This results in the following

Theorem 3.11 *Let I denote any instance of MDP^{node}_{sum}-TI and suppose P^* is an optimal placement of diameter $OPT(I) = \mathcal{D}_{\delta_d}(P^*)$. Then there is a polynomial time algorithm that returns a placement P of total weight no more than $(2-2/p)\Omega$ and diameter $\mathcal{D}_{\delta_d}(I)/OPT(I) \leq 2$.*

In the special case that $\delta_c(e) = 0$ for all $e \in E_c$, i.e. that there are only costs involved with the nodes of the network G, the last result can be improved substantially. In fact, we will now present an easy heuristic with a $(2,1)$-performance guarantee, i.e. a heuristic which will *not* violate the budget constraint on the costs of the nodes in the placement.

Lemma 3.12 *Assume I is an instance of MDP_{node}-TI and the test procedure* node–test$(G_i, \delta_d, \Omega, \omega)$ *returns a "certificate of failure", then we have $OPT(I) > \delta_d(e_i)$.*

Proof: Straightforward. □

Theorem 3.13 *Let I denote any instance of MDP_{node}-TI and suppose P^* is an optimal placement of diameter $OPT(I) = \mathcal{D}_{\delta_d}(P^*)$. Then the generic bottleneck procedure* Heur–Generic *with the test procedure* node–test *returns a placement P of weight no more than Ω and diameter $\mathcal{D}_{\delta_d}(I)/OPT(I) \leq 2$.*

Proof: Straightforward. □

4 Extensions to the Diameter–constrained Sum of Weights

In this section we will show briefly, how the techniques in the heuristic for MDP_{bott} can be used to obtain an approximation for the case, when the objective is of sum–type.

PROCEDURE node-test$(G, \delta, \Omega, \omega)$

1 $V_{cand} := \{v \in G : \deg(v) \geq p-1\}$

2 **IF** $V_{cand} = \emptyset$ **THEN RETURN** "certificate of failure"

3 Let $best := +\infty$

4 Let $P_{best} := \emptyset$

5 **FOR** each $v \in V_{cand}$ **DO**

(a) Let $N(v)$ be the set of $p-1$ lightest neighbors of v

(b) Let $P(v) := N(v) \cup \{v\}$

(c) **IF** $\omega(P(v)) \leq \Omega$ **AND** $\mathcal{D}_\delta(P(v)) < best$ **THEN** $P_{best} := P(v)$
$best := \mathcal{D}_\delta(P(v))$

6 **IF** $best = +\infty$ **THEN RETURN** "certificate of failure"
ELSE RETURN P_{best}

Figure 4: Test Procedure used for Heur–MDP$_{node}$

Here it seems adequate to interpret the problem in the following sense. Given a bound Ω on the communication diameter of the placement, minimize the construction cost. We formalize this problem:

Definition 4.1 (MAP with Bottleneck Constraints (MAP_{bott}))
Input: *An undirected complete graph $G = (V, E_c)$ with two nonnegative weight functions $\delta_c, \delta_d : E_c \to \mathbb{Q}_+$, an integer $2 \le p \le n$ and a number $\Omega \in \mathbb{Q}_+$.*

Output: *A set $P \subseteq V$, $|P| = p$ minimizing the objective*

$$\mathcal{S}_{\delta_c}(P) = \sum_{\substack{v_i, v_j \in P \\ v_i \neq v_j}} \delta_c(v_i, v_j)$$

subject to the constraint

$$\mathcal{D}_{\delta_d}(P) = \max_{\substack{v, w \in P \\ v \neq w}} \delta_d(v, w) \le \Omega$$

We can use the proof for proposition 3.2, which in fact did not use any arguments involved with the objective function, to obtain the following hardness result for MAP_{bott}:

Proposition 4.2 *Unless $\mathcal{P} = \mathcal{NP}$, for any fixed $\varepsilon > 0$ and $\varepsilon' > 0$, there can be no polynomial time algorithm for MAP_{bott}-TI that is required to place at least $\frac{2p}{|V|^{1/6-\varepsilon'}}$ facilities and that provides a performance guarantee of $(\alpha, 2 - \varepsilon)$.*

We now consider the heuristic Heur–MDP–MAP_{bott} from Figure 1, which has already been used for MDP_{bott}, but this time called with $\mathcal{M}_{\delta_d} = \mathcal{S}_{\delta_d}$ instead of $\mathcal{D}_{\delta_d}$. Then we have

Theorem 4.3 *Let I be any instance of of MAP_{bott}-TI such that an optimal solution P^* of diameter $OPT(I) = \mathcal{D}_{\delta_d}(P^*)$ exists. Then the algorithm Heur–MDP–MAP_{bott}, called with $\mathcal{M}_{\delta_c} := \mathcal{S}_{\delta_d}$, returns a placement P satisfying $\mathcal{D}_{\delta_d}(P)/OPT(I) \le (2 - 2/p)$ and $\mathcal{D}_{\delta_c}(P) \le 2$.*

Proof: Use the arguments from the proof of theorem 3.4 to obtain the fact that the constraint $\mathcal{D}_{\delta_d}(P) \le \Omega$ is violated by a factor of at most 2.

Then use the techniques in the proof of theorem 3.8 to prove the performance guarantee of $(2 - 2/p)$ with respect to the sum of the construction costs. □

References

[ABP90] B. Awerbuch, A. Baratz, and D. Peleg. Cost–sensitive analysis of communication protocols. In *Proceedings of the 9th Symposium on the Principles of Distributed Computing (PODC)*, pages 177–187, 1990.

[AFMP94] E.M. Arkin, S.P. Fekete, J.S.B. Mitchell, and C.D. Piatko. Optimal Covering Tour Problems. In *Proceedings of the Fifth International Symposium on Algorithms and Computation*, 1994.

[AIKS91] A. Aggarwal, H. Imai, N. Katoh, and S. Suri. Finding k points with Minimum Diameter and Related Problems. *J. Algorithms*, 12(1):38–56, March 1991.

[BIP90] Judit Bar-Ilan and David Peleg. Approximation Algorithms for Selecting Network Centers. *LNCS*, 519:343–354, 1990.

[BS94] Mihir Bellare and Madhu Sudan. Improved Non–Approximability Results. In *Proceedings of the 26th annual ACM Symposium on the Theory of Computing*, 5 1994.

[DF85] M.E. Dyer and A.M. Frieze. A Simple Heuristic for the p–Center Problem. *Operations Research Letters*, 3(6):285–288, Feb. 1985.

[EN89] E. Erkut and S. Neuman. Analytical Models for Locating Undesirable Facilities. *European J. of Operations Research*, 40:275–291, 1989.

[FG88] T. Feder and D. Greene. Optimal Algorithms for Approximate Clustering. In *ACM Symposium on Theory of Computing*, pages 434–444, 1988.

[Fis93] T. Fischer. Optimizing the Degree of Minimum Weight Spanning Trees. Technical report, Department of Computer Science, Cornell University, Ithaca, New York, April 1993.

[GJ79] M.R. Garey and D.S. Johnson. *Computers and Intractability.* W.H. Freeman, 1979.

[Gon85] T.F. Gonzalez. Clustering to Minimize the Maximum Intercluster Distance. *Theoretical Computer Science*, 38:293–306, 1985.

[Has92] R. Hassin. Approximation schemes for the restricted shortest path problem. *Mathematics of Operations Research*, 17(1):36–42, 1992.

[HM79] G.Y. Handler and P.B. Mirchandani. *Location on Networks: Theory and Algorithms.* MIT Press, Cambridge, MA, 1979.

[HS86] Dorit S. Hochbaum and David B. Shmoys. A Unified Approach to Approximation Algorithms for Bottleneck Problems. *Journal of the ACM*, 33(3):533–550, July 1986.

[ICTV86] A. Iwainsky, E. Canuto, O. Taraszow, and A. Villa. Network decomposition for the optimization of connection structures. *Networks*, 16:205–235, 1986.

[KRY93] S. Khuller, B. Raghavachari, and N. Young. Balancing Minimum Spanning and Shortest Path Trees. In *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 243–250, 1993.

[Lee82] D.T. Lee. On k–nearest neighbor Voronoi diagrams in the plane. *IEEE TRans. Comput.*, C–31:478–487, 1982.

[LV92] Jyh-Han Lin and Jeffrey Scott Vitter. ε–Approximations with Minimum Packing Constraint Violation. In *STOC*, pages 771–781, 5 1992.

[MF90] P.B. Mirchandani and R.L. Francis. *Discrete Location Theory.* Wiley–Interscience, New York, NY, 1990.

[PS85] F.P. Preparata and M.I. Shamos. *Computational Geometry: An Introduction.* Springer–Verlag Inc., New York, NY, 1985.

[RKM+93] V. Radhakrishnan, S.O. Krumke, M.V. Marathe, D.J. Rosenkrantz, and S.S. Ravi. Compact Location Problems. In *13th Conference on the Foundations of Software Technology and Theoretical Computer Science*, volume 761 of *LNCS*, pages 238–247, December 1993.

[RMR+93] R. Ravi, M.V. Marathe, S.S. Ravi, D.J. Rosenkrantz, and H.B. Hunt III. Many birds with one stone: Multi-objective approximation algorithms. In *Proceedings of the 25th Annual ACM Symposium on the Theory of Computing*, pages 438–447, 1993.

[RRT91] S.S. Ravi, D.J. Rosenkrantz, and G.K. Tayi. Facility Dispersion Problems: Heuristics and Special Cases. In *Proc. 2nd Workshop on Algorithms and Data Structures (WADS)*, pages 355–366, Ottawa, Canada, August 1991. Springer Verlag, LNCS vol. 519.

[ST93] D.B. Shmoys and E. Tardos. Scheduling unrelated parallel machines with costs. In *Proceedings of the 4th Annual ACM-SIAM SODA*, pages 438–447, 1993.

[War87] A. Warburton. Approximation of Pareto optima in multiple–objective, shortest path problems. *Operations Research*, 35:70–79, 1987.

[Won80] R.T. Wong. Worst case analysis of network design problem heuristics. *SIAM J. Alg. Disc. Meth.*, 1:51–63, 1980.

Appendix: Some Proofs

Proof of Proposition 3.2:

Let I' be an arbitrary instance of MAX–CLIQUE, given by a graph $G' = (V', E')$. Without loss of generality we can assume that $E' \neq \emptyset$.

For each $2 \leq k \leq |V|$ we construct an instance $I^{(k)}$ of MDP$_{both}$–TI in the following way: We let $G^{(k)} = (V', E_c)$ $(E_c = \{(u,v) : u, v \in V, u \neq v\})$ and define $\delta_c^{(p)}, \delta_d^{(p)} : E_c \to \mathbb{N}$ by $\delta_c^{(p)}(e) := 1$ for all $e \in E_c$ and

$$\delta_d^{(k)}(e) := \begin{cases} 1 & \text{if } e \in E' \\ 2 & \text{else} \end{cases}$$

It is trivial to see that both weight functions obey the triangle inequality. We let $\Omega^{(k)} := 1$ and $p^{(k)} := k$. Notice that the size of an instance $I^{(k)}$ is still polynomial in the size of I', and that we have constructed only polynomially many (namely $O(|V|)$) instances.

Now consider an instance $I^{(k)}$.

Note that any placement P of $p^{(k)} = k$ nodes that has communication diameter $\mathcal{D}_{\delta'_d}(P) \leq (2-\varepsilon)\Omega = (2-\varepsilon)$ must have diameter 1. Also, any subset $S \subseteq V$ having diameter $\mathcal{D}_{\delta'_d}(S) = 1$ must form a clique in the original graph G''.

Assume that the original graph G' has a clique C of size $p^{(k)} = k$. Then this clique will satisfy $\mathcal{D}_{\delta'_d}(C) = \mathcal{D}_{\delta'_d}(C) = 1 = \Omega'$. By our assumption, the algorithm A must return a set S of at least $\frac{2p}{|V|^{1/6-\varepsilon'}}$ nodes with communication diameter at most $(2-\varepsilon)\Omega = (2-\varepsilon) < 2$. Thus, as noted above, the algorithm will have to find a placement of diameter 1, and this set will form a clique in the original graph G''.

If there is no clique of size $p^{(k)} = k$ in G'', any placement P of $p^{(k)} = k$ nodes in G' will have to include at least one edge e of length $\delta'_d(e) = 2 > (2-\varepsilon)$. Now, according to our assumptions about A, the algorithm has the choice of either returning a set of size at least $\frac{2p}{|V|^{1/6-\varepsilon'}}$ that will form a clique in the original graph or providing the information that there is no placement P of diameter at most $\Omega = 1$.

Thus, the output of the algorithm A can be used to either obtain the information that G' does not contain a clique of size $p^{(k)} = k$ or that G' does have a clique of size at least $\frac{2p}{|V|^{1/6-\varepsilon'}}$.

Now, we can run A for all the instances $I^{(k)}$ $(2 \leq k \leq |V|)$. Since the size of each instance $I^{(k)}$ is polynomial in the size of I' and we only have $O(|V|)$ instances, this will result in an overall polynomial time algorithm, according to our assumptions about A. Let $m := \max\{k : \text{A}$ returns a set S of diameter $1\}$. Then, by our observations from above, we can conclude that G' has a clique of size at least $\frac{2m}{|V|^{1/6-\varepsilon'}}$ and that there is no clique of size $m+1$ in G'. Hence, we can approximate the maximum clique number of G'' by a factor of at most $\frac{m+1}{2m} \cdot |V|^{1/6-\varepsilon'} \leq |V|^{1/6-\varepsilon'}$. By the results in [BS94] this will imply that $\mathcal{P} = \mathcal{NP}$. □

Proof of Theorem 3.4:
If there is an optimal solution P^* that does involve a communication diameter $\mathcal{D}_{\delta_d}(P^*)$ of no more than Ω, then by definition this placement does form a clique of size p in $G'' :=$ bottleneck(G, δ_d, Ω). Thus in this case V_{cand} is non-empty and the heuristic will not output a "certificate of failure".

Moreover, any placement $P(v)$ considered by the heuristic will form a clique in $(G'')^2$. By definition of G'' as the bottleneck graph with respect to δ_d and the bound Ω, it follows by the triangle inequality that no edge e in $(G'')^2$ has weight $\delta_d(e)$ more than 2Ω. Thus *every* placement $P(v)$ considered by the heuristic has communication diameter of at most $\mathcal{D}_{\delta_d}(P(v))$ of no more than 2Ω.

Let $v \in P^*$ be arbitrary. Then we clearly have $v \in V_{cand}$. Consider the step, when the heuristic considers v. For any $w \in N(v)$ we have $\delta_c(v, w) \leq OPT(I)$, by definition of $N(v)$ as the set of nearest neighbors of v and by the fact that every node from the optimal solution is adjacent to v in G''. Thus for $w, w' \in N(v)$ we have $\delta_c(w, w') \leq \delta_c(v, w) + \delta_c(v, w') \leq 2OPT(I)$ by the triangle inequality. Consequently, $\mathcal{D}_{\delta_c}(P(v)) = \mathcal{D}_{\delta_c}(N(v) \cup \{v\}) \leq 2OPT(I)$.

Now, as the algorithm chooses the placement with minimal diameter among all the placements produced, the claimed performance guarantee with respect to the cost diameter $\mathcal{D}_{\delta_c}$ follows. □

Proof of Proposition 3.6:
We show that an algorithm A with the performance guarantee mentioned in the proposition can be used to decide an arbitrary instance I of CLIQUE, given by a graph $G = (V, E)$ and an integer $1 \leq p \leq n$.

We define the graph in instance I' of MDP$_{sum}$–TI by $G' = (V, E_c)$, set $\Omega' := p(p-1)$ and define the weight functions via $\delta'_d(e) := 1$ for all $e \in E_c$ and

$$\delta'_c(e) := \begin{cases} 1 & \text{, if } e \in E \\ 2 & \text{, else.} \end{cases}$$

It is now easy to see that there is a placement P satisfying $S_{\delta'_c}(P) \leq \Omega'$ if and only if the original graph G has a clique of size p.

If we do not require the distances to obey the triangle inequality, we can simply replace the 2 from above by a suitable large constant M and the desired result follows. □

Proof of Theorem 3.8:
Consider the case when $\delta_d(e_i) = OPT(I)$. Since in G_i we have deleted only edges e having

weight $\delta_d(e) > OPT(I)$ and we assume that there is a feasible solution within our budget-constraints, it follows that the bottleneck graph G_i must contain a clique C of size p such that $S_{\delta_c}(C) \leq \Omega$.

For a node $v \in C$ let

$$S_v := \sum_{\substack{w \in C \\ w \neq v}} \delta_c(v, w).$$

Then we have

$$S_{\delta_c}(C) = \sum_{v \in C} S_v.$$

Now let $v \in C$ be so that S_v is minimal among all nodes in C. Then clearly

$$S_{\delta_c}(C) \geq p S_v. \tag{2}$$

By definition of the bottleneck graph G_i and the clique C, the node v must have degree at least $p-1$ in G_i. Thus v is one of the nodes considered by the test procedure. Let $N(v)$ be the set of $p-1$ nearest neighbors of v in G_i. Then we have

$$\sum_{\substack{w \in N(v) \\ w \neq v}} \delta_c(v, w) \leq S_v, \tag{3}$$

by definition of $N(v)$ as the set of nearest neighbors, $P(v) := N(v) \cup \{v\}$. Let $w \in N(v)$ be arbitrary. Then

$$\begin{aligned}
\sum_{u \in N(v) \cup \{v\} \setminus \{w\}} \delta_c(w, u) &= \delta_c(w, v) + \sum_{u \in N(v) \setminus \{w\}} \delta_c(w, u) \\
&\leq \delta_c(w, v) + \sum_{u \in N(v) \setminus \{w\}} (\delta_c(w, v) + \delta_c(v, u)) \\
&= (p-1)\delta_c(w, v) + \sum_{u \in N(v) \setminus \{w\}} \delta_c(v, u) \\
&= (p-2)\delta_c(v, w) + \sum_{u \in N(v)} \delta_c(v, u) \\
&\overset{(3)}{\leq} (p-2)\delta_c(v, w) + S_v
\end{aligned} \tag{4}$$

Now using (4) and again (3), we obtain

$$\begin{aligned}
S_{\delta_c}(P(v)) &= S_{\delta_c}(N(v) \cup \{v\}) \\
&= \sum_{u \in N(v)} \delta_c(v,u) + \sum_{w \in N(v)} \sum_{u \in N(v) \cup \{v\} \setminus \{w\}} \delta_c(w,u) \\
&\overset{(3)}{\leq} S_v + \sum_{w \in N(v)} \sum_{u \in N(v) \cup \{v\} \setminus \{w\}} \delta_c(w,u) \\
&\overset{(4)}{\leq} S_v + \sum_{w \in N(v)} ((p-2)\delta_c(v,w) + S_v) \\
&= S_v + (p-2)S_v + (p-1)S_v \\
&= (2p-2)S_v \\
&\overset{(2)}{\leq} (2 - 2/p)OPT(I)
\end{aligned}$$

Thus the placement $P(v)$ violates the budget–constraint by a factor of at most $2 - 2/p$. Consequently, as the algorithm chooses the placement with P_{best} with the least constraint-violation, it follows that the test–procedure called with $G_i = \text{bottleneck}(G, \delta_d, OPT(I))$ will not return a "certificate of failure".

The placement P_{best} that is produced by the algorithm turns into a clique in G_i^2. Thus the longest edge in the placement with respect to δ_d is at most $2OPT(I)$. □