Minimizing Average Shortest Path Distances via Shortcut Edge Addition

Adam Meyerson and Brian Tagiku

University of California, Los Angeles. E-mail:{awm,btagiku}@cs.ucla.edu

Abstract. We consider adding k *shortcut edges* (*i.e.* edges of small fixed length $\delta \geq 0$) to a graph so as to minimize the weighted average shortest path distance over all pairs of vertices. We explore several variations of the problem and give $O(1)$-approximations for each. We also improve the best known approximation ratio for metric k-median with penalties, as many of our approximations depend upon this bound. We give a $(1 + 2\frac{(p+1)}{\beta(p+1)-1}, \beta)$-approximation with runtime exponential in p. If we set $\beta = 1$ (to be exact on the number of medians), this matches the best current k-median (without penalties) result.

1 Introduction

Multi-core processors have become popular in modern computer architectures because they provide large gains in performance at relatively low cost. In many of these processors the multiple cores are connected as a Network-on-Chip (NoC) as described in [5]. While each individual core may be slower than a state-of-the-art single-core processor, together they form a processor well-suited for largely parallel applications. Moreover, NoC designs avoid tedious power and heat constraints associated with single-core processor design. Instead, the important concern is how to best connect these multiple cores into a single, efficient network.

NoC designs typically use mesh networks since regular topologies are easier to manufacture. However, many pairs of nodes are far apart in mesh graphs. Thus, it becomes necessary to add several long interconnects to decrease average communication latency. While traditional interconnects become inhibitively slow when too long (see [13]), radio-frequency (RF) interconnects, introduced in [8], exhibit much better performance. Unfortunately, RF interconnects require much more area and cannot completely replace traditional interconnects.

Despite this, Chang *et. al.* show how to reap the benefits of RF interconnects without significantly increasing area. They propose in [7, 9] a hybrid architecture which uses an underlying mesh topology (using traditional interconnects) with an overlay of a small number of RF interconnects, each of which forms a fast point-to-point connection between otherwise distant nodes. Yet, Chang *et. al.* leave open the question of how to best place these RF interconnects given the traffic profile (between pairs of cores) of a specific application.

We formulate this as a general network design problem which we call the Average Shortest Path Distance Minimization (ASPDM) problem: Given a graph with weights on pairs of nodes, find k shortcut edges (of length $\delta \geq 0$) whose addition minimizes the weighted average shortest path distance over all pairs of nodes. We give the following results, where α is the best approximation known for metric k-Median with Penalties:

1. an α-approximation for Single-Source (one-to-all) ASPDM,
2. a 2α-approximation if all pairs have equal weight (Unweighted ASPDM),
3. a $(4\alpha, 2)$-approximation (*i.e.* a 4α-approximation using at most $2k$ edges) for general ASPDM,
4. an α-approximation if paths can use at most one shortcut (1-ASPDM), and
5. an $(\frac{e}{e-1})$-approximation on the improvement in cost for 1-ASPDM.

We show all the above versions to be NP-complete. We also improve the approximation to k-median with penalties by applying local search to $(1+2\frac{p+1}{\beta(p+1)-1}, \beta)$, where an (α, β)-approximation implies that we achieve an α-approximation on cost using at most βk medians. This gives us a smooth tradeoff between allowing additional medians and reducing the cost, and if we require exactly k medians ($\beta = 1$) it gives $\alpha = 3 + \varepsilon$.

Shortcut addition is frequently used in computer networks to obtain small-world topologies. Yet, existing techniques are either heuristic approaches [17, 20] or consider specific graphs [22, 15, 18, 21]. Other related problems are the Buy-at-Bulk [4], Rent-or-Buy [12] and Cost-Distance [19] problems which consider purchasing edges in a network. However, unlike these problems, ASPDM places a hard limit on the number of shortcuts. Our results guarantee constant approximations on general graphs despite this hard constraint.

2 Problem Formulation

Let $G = (V, E)$ be an undirected graph with non-negative edge lengths ℓ_e for each $e \in E$ and non-negative weights w_{uv} on each ordered pair of vertices $u, v \in V$. We use d_{uv} to denote the length of the shortest uv-path for vertices $u, v \in V$. The *weighted one-to-all shortest path sum* $D_u(G)$ from vertex u is defined as

$$D_u(G) = \sum_{v \in V} w_{uv} d_{uv}.$$

We then define the *weighted all-pairs shortest-path sum* $D(G)$ to be

$$D(G) = \sum_{u \in V} D_u(G) = \sum_{u \in V} \sum_{v \in V} w_{uv} d_{uv}.$$

Then the *weighted average shortest path distance* $\bar{D}(G)$ over all pairs of vertices is simply $D(G)$ divided by the sum of all the ordered pair weights. Throughout this paper we will be interested in minimizing $\bar{D}(G)$, but it is easy to see that it is equivalent to minimize $D(G)$.

We can now formally define the Average Shortest Path Distance Minimization via Shortcut Edge Addition problem (ASPDM) as follows:

Problem 1 (ASPDM). *Given an undirected graph $G = (V, E)$ with lengths ℓ_e on the edges $e \in E$, weights w_{uv} for each ordered pair of vertices $u, v \in V$, a shortcut edge length $\delta \geq 0$ and an integer k, find a set $F \subseteq V \times V$ of at most k shortcut edges of length δ such that $\bar{D}(G + F)$ is minimized.*

Of course, $F + G$ may be a multi-graph if $F \cap E \neq \emptyset$. In some cases we can consider *directed* shortcuts, but graph G must remain undirected for reasons stated in Section 4. For simplicity of analysis we assume that $\delta = 0$, but all our results extend to arbitrary $\delta \geq 0$.

We consider several variations of ASPDM. The Single-Source ASPDM problem (SS-ASPDM) is the case where the only non-zero weights are on pairs involving a designated source vertex s. Unweighted ASPDM (U-ASPDM) places equal weight on all pairs (which may be the case for general-application NoC designs where weights are unknown). Finally, the 1-Shortcut Edge Restricted ASPDM (1-ASPDM) restricts that each shortest path uses at most one of the added shortcut edges. 1-ASPDM is a suitable model for NoC design since it reduces the complexity of the routing tables that need to be stored in the design and also reduces congestion along these shortcuts.

3 Preliminaries and Initial Observations

In this section, we review k-median with penalties which we use in many of our results below. We will also analyze an algorithm for SS-ASPDM, which is a useful subroutine for more general results.

3.1 Metric k-Median with Penalties

In k-median with penalties, we are given a set of cities and a set of potential facility locations arranged in a metric space. Each city has a demand that needs to be served by a facility. Each city also has a penalty cost, which we can pay to refuse service to the city. If we choose to serve a city, we must pay the distance between the city and its assigned facility for each unit demand. Our job is to find a set of k facilities to open, a set of cities to be served, and an assignment of cities to open facilities such that our total cost is minimized.

Throughout this paper, we use α to denote the ratio of the best approximation algorithm for k-median with penalties. We use this approximation as a subroutine in many of our algorithms. Because of the inapproximability of asymmetric k-median ([2]), our algorithms only apply to undirected graphs. However, most of our algorithms permit directed shortcuts.

3.2 Single Source ASPDM

In this section we consider SS-ASPDM where only the weights w_{sv} may be nonzero for some designated source s and $v \in V$. Thus, we are simply minimizing $D_s(G)$. This model will become useful in analyzing the complexity of our ASPDM variants as well as for obtaining an approximation for U-ASPDM.

Lemma 1. *For every instance of SS-ASPDM, there exists an optimal set F^* such that each edge $e \in F^*$ is incident on s. Moreover, for every $v \in V$, there exists a shortest sv-path that uses at most one edge in F^*.*

Proof. Let F^* be an optimal set of shortcut edges and consider $e = uv \in F^*$. Suppose p_1 is a shortest sx-path that traverses e in the uv direction and p_2 is a shortest sy-path that traverses e in the vu direction. Then the sy-path p_3 that starts at s, follows p_1 until u then follows p_2 never crosses e and can be no longer than p_2 (otherwise there would exist a sx-path shorter than p_1). Thus, e has an implicit orientation such that it is only ever used in the correct direction.

Since e is only used in one direction (say, u to v), then moving u closer to s only improves our cost. Thus, $F^* - uv + sv$ is at least as good a solution. We can do this for all other edges so that F^* contains only edges incident on s. Notice that now since every shortcut edge is incident on s, there is never any incentive to use more than one shortcut in a shortest path. □

Then we need only find k endpoints for our edges that minimize our cost if for each vertex v we pay either its weighted distance to the nearest endpoint or a penalty $w_{sv} d_{sv}$. This is precisely the k-median with penalties problem, thus we have an α-approximation algorithm for SS-ASPDM.

Theorem 1. *There exists a polynomial-time α-approximation algorithm ALG_{SS} for SS-ASPDM.*

Moreover, this α-approximation holds when adding directed shortcuts (to an undirected graph) since each edge $e \in F^*$ is only ever used in a single orientation.

4 Complexity

Consider unweighted (*i.e.* all non-zero weights are equal) SS-ASPDM. We now show that this problem is NP-Hard via reduction from the well-known Set Cover problem (defined in [11]).

Theorem 2. *Unweighted SS-ASPDM is NP-Hard. Further, for directed graphs, unweighted SS-ASPDM is hard to approximate to better than $\Omega(\log |V|)$.*

Proof. Omitted. Here, we give only the construction: Given an instance of set cover with universe U, subset collection $\mathcal{C}$ and integer k, let G have a vertex v_x for every $x \in U$, a vertex v_S for every $S \in \mathcal{C}$, and a vertex s. There is an edge of length 1 from s to each v_S and an edge of length 1 from v_S to each v_x where $x \in S$. Notice that $D_s(G) = |\mathcal{C}| + 2|U|$. We can now solve set cover by asking if there is a set F of k shortcut edges such that $D_s(G + F) \leq |\mathcal{C}| - k + |U|$. □

Unweighted SS-ASPDM is clearly a restriction of SS-ASPDM and ASPDM. By Lemma 1, SS-ASPDM is also a restriction of 1-ASPDM. The above reduction works for U-ASPDM when we replace s with a sufficiently large clique (connected by length-0 edges). Thus, we immediately get that all these problems are NP-Hard.

Corollary 1. *SS-ASPDM, U-ASPDM, 1-ASPDM, ASPDM are all NP-Hard.*

5 Unweighted ASPDM

In this section, we consider U-ASPDM where all pairs have equal weight. We will give an approximation algorithm which uses our SS-ASPDM algorithm ALG_{SS} as a subroutine. To do this, we must first claim that there exists a vertex x that is sufficiently close to all other vertices.

Lemma 2. *There exists an x such that when used as the source ALG_{SS} returns a 2α-approximation.*

Proof. Let F^* be the optimal solution. The average value of $D_v(G+F^*)$ over all v is $\frac{1}{n}D(G+F^*)$. Thus, some vertex x must not exceed the average. Try adding edge set F so as to minimize $D_x(G+F)$. By Theorem 1, we can do this within α of optimal using ALG_{SS}. Since $D_x(G+F^*)$ is no better than optimal,

$$D_x(G+F) \leq \alpha \cdot D_x(G+F^*) \leq \alpha \cdot \frac{1}{n} D(G+F^*). \tag{1}$$

We can also bound $D(G+F)$ in terms of $D_x(G+F)$. Since d is a metric, for each u, v we have $d_{uv} \leq d_{ux} + d_{xv}$. Summing these inequalities over all pairs gives

$$D(G+F) \leq 2nD_x(G+F). \tag{2}$$

Finally, combining Equations 1 and 2 gives the desired result

$$D(G+F) \leq 2nD_x(G+F) \leq 2n\alpha D_x(G+F^*) \leq 2\alpha D(G+F^*).$$

□

Thus, treating the all-pairs problem as a single-source problem with source vertex x produces a 2α-approximation. However, since finding x requires knowledge of F^*, we must instead try all possible x and take the best solution. We note that while ALG_{SS} works with directed shortcuts, this algorithm does not since edges may need to be used in both directions.

Theorem 3. *There exists a polynomial-time 2α-approximation algorithm for U-ASPDM.*

6 General ASPDM

We now consider the most general version of the problem where each pair can have an arbitrary weight associated with it. For this version, we offer a bicriteria approximation algorithm that breaks the restriction that only k edges be added.

Theorem 4. *There exists a polynomial-time $(4\alpha, 2)$-approximation algorithm for ASPDM. In particular, this algorithm gives at most $2k-1$ edges yielding cost at most 4α-times the optimum k-edge cost.*

Proof. Let F^* be the optimal set of k edges. Notice that these edges involve $j \leq 2k$ endpoints. Let $\hat{F}$ be a set of $j-1 \leq 2k-1$ edges that connect these endpoints as a star. Thus, we can travel between any two endpoints using two shortcuts giving $D(G+\hat{F}) \leq 2D(G+F^*)$.

Since we do not know the set of endpoints used by F^* *a priori*, we try to find a star F over $2k$ points that minimizes $D(G+F)$. We can use $2k$-median with penalties to find this approximate solution F. To do this, we duplicate each vertex u so that the $2k$-median solution can connect u to some vertices and deny connections to others. We duplicate u a total of $2n-2$ times introducing u_{uv} and u_{vu} for each $v \neq u$, having weights w_{uv} and w_{vu} and penalties $\max\{0, w_{uv}(d_{uv} - 2\delta)\}$ and $\max\{0, w_{vu}(d_{vu} - 2\delta)\}$, respectively. Since all the vertices corresponding to u are co-located, we need only choose one representative as a potential facility location.

For each pair u, v the $2k$-median instance pays for "connecting" u and v through these medians and never pays more than $2w_{uv}(d_{uv} - 2\delta)$. Adding the cost due to traversing shortcuts between these medians shows the optimum $2k$-median solution will have cost less than $2D(G+\hat{F})$. Using an α approximation gives us a cost of:

$$D(G+F) \leq 2\alpha D(G+\hat{F}) \leq 4\alpha D(G+F^*).$$

It follows that F gives a 4α-approximation for this problem. $\square$

Notice that when $\delta = 0$ we can actually improve this to a $(2\alpha, 2)$-approximation since we have $D(G+\hat{F}) \leq D(G+F^*)$. In this case, we can also deal with directed shortcuts if we connect the $2k$ endpoints as a directed cycle (thus, using exactly $2k$ shortcuts).

7 1-Shortcut Edge Restricted ASPDM

We consider a restriction that each path must use at most one shortcut edge. This allows us to provide improved approximations (in particular removing the increase over k shortcut edges). For real NoC designs, this kind of restriction ensures no pair monopolizes the RF interconnects and permits simplified routing.

7.1 Approximating Total Cost

We first define a metric over pairs of points $V \times V$.

Theorem 5. *If (V, d) is a metric, then so is the space $(V \times V, \hat{d})$ where*

$$\hat{d}(x_1y_1, x_2y_2) = \min(d(x_1, x_2) + d(y_2, y_1), d(x_1, y_2) + d(x_2, y_1)).$$

Proof. Omitted. $\square$

Note that in this space, we can naturally assign weight w_{uv} and penalty $w_{vu}d_{uv}$ to point uv. Moreover, if we select xy as a shortcut edge, then any 1-shortcut edge restricted shortest uv-path using xy has length $\hat{d}(uv, xy)$. Then adding k shortcut edges is equivalent to picking k medians in this pairs-of-points space. Thus, we can use k-median with penalties to obtain an α approximation.

Corollary 2. *There exists a polynomial-time α-approximation algorithm for 1-ASPDM.*

This works for directed shortcuts if we instead use $\hat{d}(x_1y_1, x_2y_2) = d(x_1, x_2) + d(y_2, y_1)$ which explicitly uses shortcuts in the correct direction.

7.2 Approximating cost improvement

The previous result guarantees a solution cost of at most $\alpha D(G+F^*)$. However, if $D(G+F^*) \geq \frac{1}{\alpha}D(G)$, then this guarantee can exceed $D(G)$, which even a trivial solution could satisfy! In such cases, it is more meaningful to approximate the optimum amount of improvement. We define $\Delta(G, H) = D(G) - D(H)$. Then we want our solution F to satisfy

$$\Delta(G, G + F) \geq \frac{1}{\zeta}\Delta(G, G + F^*)$$

for some $\zeta \geq 1$. We can obtain such an approximation using linear programming.

We first give an ILP formulation for 1-ASPDM. We use binary variables $x_{xy}, f^{st}_{uv}, g^{st}_{uv}, h^{st}_{xy}$ for each $s, t \in V$, $uv \in E$ and shortcut edge xy whose addition we are considering. If $x_{xy} = 1$ then edge $xy \in F$. Each pair (s, t) is given one unit of flow that needs to travel from s to t. Variable f^{st}_{uv} indicates the amount of (s, t)-flow over edge uv allowed to use a shortcut edge. Similarly, g^{st}_{uv} indicates the amount of (s, t)-flow over edge uv that has already used a shortcut edge. Finally, $h^{st}_{xy} \in \{0, 1\}$ indicates the amount of (s, t)-flow over shortcut edge xy.

Our ILP formulation is as follows:

$$\begin{aligned}
\text{minimize} \quad & \sum_{s,t} \left[w_{st} \cdot \sum_{uv \in E} \ell_{uv} \left(f^{st}_{uv} + g^{st}_{uv} \right) \right] && && (3)\\
\text{subject to} \quad & \sum_{x,y} x_{xy} = k && && (4)\\
& h^{st}_{xy} \leq x_{xy} && \forall s, t, x, y && (5)\\
& \sum_{v \in \Gamma(s)} f^{st}_{sv} + \sum_{y} h^{st}_{sy} = 1 && \forall s, t && (6)\\
& \sum_{u \in \Gamma(w)} f^{st}_{uw} = \sum_{v \in \Gamma(w)} f^{st}_{wv} + \sum_{y} h^{st}_{wy} && \forall s, t, \forall w \neq s, t && (7)\\
& \sum_{u \in \Gamma(w)} g^{st}_{uw} + \sum_{x} h^{st}_{xw} = \sum_{v \in \Gamma(w)} g^{st}_{wv} && \forall s, t, \forall w \neq s, t && (8)\\
& x_{xy}, f^{st}_{uv}, g^{st}_{uv}, h^{st}_{xy} \in \{0, 1\} && \forall s, t, u, v, x, y && (9)
\end{aligned}$$

where $\Gamma(v)$ are the neighbors of vertex v in graph G. Equation (4) ensures that exactly k edges are selected and Equation (5) ensures that we only use selected shortcuts. Equation (6) enforces that for each pair (s,t), s adds one unit of (s,t)-flow to the graph. Equation (7) and (8) enforce conservation of flow at each vertex other than s,t (this also stipulates that t sink the one unit of (s,t)-flow). Finally, Equation (9) enforces integrality.

Since solving ILPs is NP-complete in general, we relax the integrality constraints by replacing Equation (9) with

$$0 \leq x_{xy}, f_{uv}^{st}, g_{uv}^{st}, h_{xy}^{st} \leq 1.$$

We can now use the solution to this LP as a guide for our edge selection process.

We build F iteratively using the values assigned to each x_{uv} by the optimal LP solution such that $\Pr[(uv) \in F] = x_{uv}$. Arbitrarily order the edges $e_1, e_2, \ldots, e_m$ and set $\hat{x}_{e_i} = x_{e_i}$ for all i and $F_1 = \emptyset$. In the i-th iteration, we add e_i with probability $\hat{x}_{e_i}$ to get $F_{i+1} = F_i \cup \{e_i\}$ or otherwise set $F_{i+1} = F_i$. After doing this, for each $j > i$ we set

$$\hat{x}_{e_j} \leftarrow \hat{x}_{e_j} \cdot \frac{k - |F_{i+1}|}{k - |F_i| - \hat{x}_{e_i}}.$$

We continue this process to get set $F = F_n$ containing at most k shortcut edges.

Lemma 3. *The above process yields a set F of at most k edges such that for each e_i, $1 \leq i \leq m$, we have $Pr[e_i \in F] = x_{e_i}$. Moreover, for any $S_i \subseteq \{e_1, e_2, ..., e_{i-1}\}$ we have:*

$$Pr[e_i \in F \mid S_i \cap F = \emptyset] \geq x_{e_i}.$$

Proof. Omitted. □

We can decompose the flow and calculate expected cost to get the following:

Theorem 6. *Let F^* be the optimal set of edges and F the set of edges generated by the process above. Then*

$$Ex[\Delta(G, G+F)] \geq \left(\frac{e-1}{e}\right) \Delta(G, G+F^*).$$

Proof. Fix the pair (s,t) and consider its associated flow in the LP solution. Decompose this flow into simple paths using at most one shortcut. Let $p_1, p_2, \ldots, p_\alpha$ be the paths (in order of non-decreasing length) using exactly one shortcut. Let f_i be the flow over p_i and e_i the shortcut edge used by p_i . We can assume that each path uses a distinct shortcut (we can reroute the flow from one path to the other path otherwise). By LP optimality, none of these paths are longer than d_{st}.

Let q_i be the probability that at least one of paths $p_1, \ldots, p_i$ exist in $G + F$. Then notice

$$\begin{aligned} q_i &= 1 - \Pr[\text{none of paths } p_1, \ldots, p_i \text{ exist}] \\ &= 1 - (1 - \Pr[p_1 \text{ exists}]) \cdots (1 - \Pr[p_i \text{ exists} \mid p_1, \ldots, p_{i-1} \text{ don't exist}]) \\ &\geq 1 - (1 - x_{e_1})(1 - x_{e_2}) \cdots (1 - x_{e_i}) \\ &\geq 1 - (1 - f_1)(1 - f_2) \cdots (1 - f_i) \end{aligned}$$

where the first inequality follows from Lemma 3 and the second follows from LP-feasibility. Notice this quantity is minimized when all f_js are equal. Let $S_i = \sum_{j=1}^{i} f_j$ and note that since (s,t) has only one unit of demand we have $S_\alpha = 1$. Then since $(1-x)^{1/x} \leq \frac{1}{e}$ and $0 \leq S_i \leq 1$ we have

$$q_i \geq 1 - (1 - f_1)(1 - f_2) \cdots (1 - f_i) \geq 1 - \left(1 - \frac{S_i}{i}\right)^i \geq 1 - \frac{1}{e^{S_i}} \geq \left(1 - \frac{1}{e}\right) S_i.$$

Thus, our expected cost for the (s,t)-pair is precisely

$$\begin{aligned} \text{Ex}[\text{cost}] &= d_{st} - (\ell_{p_2} - \ell_{p_1})q_1 - (\ell_{p_3} - \ell_{p_2})q_2 - \cdots - (d_{st} - \ell_\alpha)q_\alpha \\ &\leq \frac{1}{e} d_{st} + \left(1 - \frac{1}{e}\right)[\ell_{p_1} S_1 + \ell_{p_2}(S_2 - S_1) + \cdots + \ell_{p_\alpha}(S_\alpha - S_{\alpha-1})] \\ &= \frac{1}{e} d_{st} + \left(1 - \frac{1}{e}\right)[\ell_{p_1} f_1 + \ell_{p_2} f_2 + \cdots + \ell_{p_\alpha} f_\alpha] \end{aligned}$$

Summing this inequality over all (s,t) pairs gives us

$$\text{Ex}[D(G+F)] \leq \left(1 - \frac{1}{e}\right) LP + \left(\frac{1}{e}\right) D(G) \leq \left(1 - \frac{1}{e}\right) D(G+F^*) + \left(\frac{1}{e}\right) D(G)$$

where LP is the cost of the LP solution. Subtituting into our definition of $\Delta(G, G+F)$ finishes the proof. $\square$

This shows we have a $\frac{e}{e-1}$-approximation algorithm on the total amount of improvement. While this algorithm uses randomness to select the shorcut edges, we can easily derandomize the process using conditional expectations. In other words, when considering e_i, we calculate the conditional expected cost given $e_i \notin F$ and given $e_i \in F$. Once this is calculated, we follow the decision that gives us the smallest expected cost.

Corollary 3. *There exists a polynomial-time $\frac{e}{e-1}$-approximation algorithm on the improvement in cost for 1-ASPDM.*

We note that this algorithm works on directed graphs. Additionally, it works if we restrict the possible shortcuts we can add. We also note that the LP used can be rewritten as a much smaller convex program and may be more efficiently solved.

8 Improved k-median with penalties approximation

We now show that for k-median with penalties, we can use βk medians, $\beta \geq 1$, to acheive a cost of at most $1 + 2\frac{p+1}{\beta(p+1)-1}$ times the optimum cost (using k medians). For $\beta = 1$ this improves upon the 4-approximation for k-median with penalties given in [10] and matches the best approximation known for standard k-median given in [3]. This also improves upon the $(1 + \frac{5}{\varepsilon}, 3 + \varepsilon)$-approximation given in [16] for standard k-median. We note that standard k-median is hard to approximate to within $1 + \frac{2}{e}$ as shown in [14]. Our approach extends the local search based approximation algorithm given in [3] by permitting penalties and by creating a smooth bicriteria tradeoff when the algorithm is permitted to use additional medians.

Let C be the set of cities, F the set of potential facility locations and c the metric distance function. City j has demand w_j and penalty cost p_j. Thus, we are searching for a set $S \subseteq F$ of k facilities to open, a set $T \subseteq C$ of cities to serve and an assigment $\sigma : T \to S$ of cities to facilities to minimize cost

$$\mathit{cost}(S) = \mathit{serv}(S) + \mathit{deny}(S) = \sum_{j \in T} w_j c_{j,\sigma(j)} + \sum_{j \in C-T} p_j .$$

We say city j is *served* by facility i if $\sigma(j) = i$. Otherwise, city j is *denied service*. The neighborhood $\mathcal{N}_S(i)$ of facility i in solution S is the set of cities served by i. We abuse notation and write $\mathcal{N}_S(A)$ to denote the neighborhood of a set A of facilities. It will be convenient to refer to the cost due only to a set X of cities. Here we use $\mathit{cost}_X(S), \mathit{serv}_X(S), \mathit{deny}_X(S)$ to denote the total cost, service cost and denial cost (respectively) due to cities in X.

8.1 The Local Search Algorithm

Given a set of facilities S, we can easily calculate the best T and σ to use by greedily choosing to either assign each city to its closest open facility or to deny it service. Thus, we perform a local search only on the set S. Each iteration we consider all sets $A \subseteq S$ and $B \subseteq F - S$ with $|A| = |B| \leq p$ for some fixed parameter $p \geq 1$. We choose A, B such that $\mathit{cost}(S - A + B)$ is minimized and iterate until no move yields a decrease in cost. We denote swapping the sets A and B by $\langle A, B \rangle$.

8.2 Analysis

We now bound the locality gap of our algorithm:

Theorem 7. *The local search algorithm in Section 8.1 has a locality gap of at most* $1 + 2\frac{p+1}{\beta(p+1)-1}$.

Proof. Let (S, T, σ) be our solution using βk medians and (S^*, T^*, σ^*) be the optimum solution using k medians. We assume for simplicity that all weights are multiples of some $\delta > 0$. Replace each city j with $\frac{w_j}{\delta}$ copies each with weight δ and penalty $\frac{p_j \delta}{w_j}$. S and S^* treat all copies of j as they did j. Clearly, it is enough to analyze this unweighted case.

For a subset $A \subseteq S$, we will say A *captures* $o \in S^*$ if A serves at least half the cities served by both o in the optimum solution and by some facility in our solution. We then define $capture(A)$ to be the set of optimum facilities that A captures. Thus,

$$capture(A) = \{o \in S^* : \ |\mathcal{N}_S(A) \cap \mathcal{N}_{S^*}(o)| \geq \frac{1}{2}|\mathcal{N}_{S^*}(o) \cap T|\}.$$

A facility $s \in S$ is *bad* if $|capture(s)| \neq \emptyset$ and is *good* otherwise. Note that if $A, B \subseteq S$ are disjoint then so are $capture(A)$ and $capture(B)$.

Suppose S has $r-1$ bad facilities. Partition S into $A_1, \ldots, A_r$ and S^* into $B_1, \ldots, B_r$ such that for all $i \leq r-1$ we have $|A_i| = |B_i|$, $B_i = capture(A_i)$ and A_i contains exactly one bad facility. We can build this partition by adding a bad facility to each A_i then adding good facilities until $|A_i| = |capture(A_i)|$. Since each $o \in S^*$ is captured by at most one facility and $capture(A_1) \cap capture(S - A_1) = \emptyset$, we never run out of good facilities.

In fact, we only care about the A_i with $|A_i| \leq p$ (excluding A_r). Without loss of generality, we assume these to be sets $A_1, \ldots, A_b$. Let $x = \sum_{i=1}^{b} |A_i| = \sum_{i=1}^{b} |B_i|$ and note that $x \geq b$ since each A_i is non-empty. Then there are at most $k - x \leq k - b$ optimum facilities total among sets $B_{b+1}, \ldots, B_{r-1}$. Since all these sets have cardinality greater than p and there is one bad facility per A_i, we can upper bound the number of bad facilities by $b + \frac{k-b}{p+1} = \frac{k+pb}{p+1}$.

We let G be the good facilities in $A_{b+1}, \ldots, A_r$ and $a = |G|$. Then since we have βk medians total, we have

$$a \geq \beta k - \frac{k+pb}{p+1} = \frac{\beta k p + \beta k - k - pb}{p+1} \tag{10}$$

For each i such that $|A_i| \leq p$, we consider the swap $\langle A_i, B_i \rangle$. We will refer to these swaps as *set swaps*. We also consider all possible single-facility swaps between optimum facilities in B_i and facilities in G. We will call these swaps *bad singleton swaps*. Lastly, we consider all possible single-facility swaps between the remainder of optimum facilities and facilities in G. We will call these swaps *good singleton swaps*. By local optimality, each swap (either set or singleton) $\langle X, Y \rangle$ satisfies

$$cost(S - X + Y) - cost(S) \geq 0. \tag{11}$$

For each facility $o \in S^*$, partition $\mathcal{N}_{S^*}(o)$ into parts $p_X = \mathcal{N}_{S^*}(o) \cap \mathcal{N}_S(X)$ for each considered swap $\langle X, Y \rangle$ above and $p_{deny} = \mathcal{N}_{S^*}(o) - T$. We let $\pi : \mathcal{N}_{S^*}(o) \to \mathcal{N}_{S^*}(o)$ be a bijection such that $p_{deny} = \pi(p_{deny})$ and for each part $p \neq p_{deny}$ having $|p| < \frac{1}{2}|\mathcal{N}_{S^*}(o) \cap T|$ we have $p \cap \pi(p) = \emptyset$. It is easy to check that such a bijection exists.

Now let $\langle X, Y \rangle$ be a set or singleton swap considered above. When we make this swap, we can make sure to assign $\mathcal{N}_{S^*}(Y)$ to Y, but we also need to reassign any other cities served by X. If S^* denies any of these cities, we will also deny them service. Otherwise, we can reassign j to the facility serving $\pi(j)$. Thus, we can bound our change in cost above by:

$$\begin{aligned}
&0 \le cost(S - X + Y) - cost(S) \le \\
&\textstyle\sum_{j\in\mathcal{N}_{S^*}(Y)\cap T}\left[c_{j,\sigma^*(j)} - c_{j,\sigma(j)}\right] + \\
&\textstyle\sum_{j\in\mathcal{N}_{S^*}(Y)-T}\left[c_{j,\sigma^*(j)} - p_j\right] + \\
&\textstyle\sum_{j\in(\mathcal{N}_S(X)-\mathcal{N}_{S^*}(Y))\cap T^*}\left[c_{j,\sigma^*(j)} + c_{\sigma^*(j),\pi(j)} + c_{\pi(j),\sigma(\pi(j))} - c_{j,\sigma(j)}\right] + \\
&\textstyle\sum_{j\in(\mathcal{N}_S(X)-\mathcal{N}_{S^*}(Y))-T^*}\left[p_j - c_{j,\sigma(j)}\right].
\end{aligned} \tag{12}$$

Consider the inequalities corresponding to Equation 12 for each swap considered. We mutiply the inequalities for set, bad singleton and good singleton swaps by $\gamma = \frac{p+1}{\beta(p+1)-1}$, $\frac{1-\gamma}{a}$ and $\frac{1}{a}$ (respectively) then sum the resulting inequalities. Notice that each $o \in B_i$ is involved in swaps of total weight one. Thus the first two terms of Equation 12 sum to $serv(S^*) - cost_{T^*}(S)$.

Each bad facility s is involved with a set swap of weight γ or is never swapped. Each good facility s is involved in x bad singleton swaps and $k-x$ good singleton swaps for a total weight of

$$x\left(\frac{1-\gamma}{a}\right) + \frac{k-x}{a} = \frac{1}{a}\left(k - x\frac{p+1}{\beta(p+1)-1}\right) \le \frac{1}{a}\left(k - b\frac{p+1}{\beta(p+1)-1}\right) \le \gamma$$

Thus, any $j \in T \cap T^*$ is considered in a weighted total of at most γ swaps. Since $\left[c_{j,\sigma^*(j)} + c_{\sigma^*(j),\pi(j)} + c_{\pi(j),\sigma(\pi(j))} - c_{j,\sigma(j)}\right] \ge 0$ by triangle inequality and $\left[p_j - c_{j,\sigma(j)}\right] \ge 0$ we can assume that each j appears exactly γ times (this only increases the right-hand side of Equation 12). Then the third and fourth terms of Equation 12 sum to at most $\gamma\left(2serv_T(S^*) + deny_T(S^*) - serv_{T-T^*}(S)\right)$.

Thus summing Equation 12 over all swaps and rearranging gives

$$serv(S^*) + 2\gamma serv_T(S^*) + \gamma deny_T(S^*) \ge cost_{T^*}(S) + \gamma serv_{T-T^*}(S). \tag{13}$$

Since $deny_{T-T^*}(S) = 0$ as S does not deny service to any member of T, the right-hand side exceeds $cost(S)$. Since $cost(S^*) = serv(S^*) + deny(S^*)$ the left-hand side is no greater than $(1+2\gamma)cost(S^*)$. Thus, we have

$$cost(S) \le \left(1 + 2\frac{p+1}{\beta(p+1)-1}\right) cost(S^*).$$

□

9 Experiments and Future Work

We have run some experiments comparing the result of our local search-based approximation for 1-ASPDM against the heuristics described in [6] and obtained a 4-5% improvement in both latency and power. We are conducting further experiments to determine whether local search is producing optimum results in practice, and whether a more complex model might lead to even more improvement.

From a theoretical standpoint, we have given constant factor approximations for all versions of ASPDM except the most general one. Whether the general ASPDM problem has a (single criterion) constant approximation remains an open problem. The problem is related to a series of works in the theory of network design literature (for example Rent-or-Buy problems) in much the same way that k-median relates to facility location (instead of summing two types of cost, we have a hard constraint on one type and seek to minimize the other). If we permit *restrictions* on the set of available shortcuts, then approximation hardness results follow from the work of Andrews [1] but we are not aware of any such results for the case where any pair of nodes can be connected via a shortcut edge.

References

1. Matthew Andrews. Hardness of buy-at-bulk network design. In *Annual Symposium on Foundations of Computer Science*, 2004.
2. Aaron Archer. Inapproximability of the asymmetric facility location and k-median problems. Unpublished manuscript, 2000.
3. Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local search heuristics for k-median and facility location problems. In *Symposium on Theory of Computing*, 2001.
4. Baruch Awerbuch and Yossi Azar. Buy-at-bulk network design. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science*, page 542, 1997.
5. Luca Benini and Giovanni De Micheli. Networks on chips: A new SoC paradigm. *Computer*, 35(1):70–78, 2002.
6. Mau-Chung Frank Chang, Jason Cong, Adam Kaplan, Chunyue Liu, Mishali Naik, Jagannath Premkumar, Glenn Reinman, Eran Socher, and Sai-Wang Tam. Power reduction of CMP communication networks via RF-interconnects. In *Proceedings of the 41st Annual International Symposium on Microarchitecture*, November 2008.
7. Mau-Chung Frank Chang, Jason Cong, Adam Kaplan, Mishali Naik, Glenn Reinman, Eran Socher, and Sai-Wang Tam. CMP network-on-chip overlaid with multi-band RF-interconnect. In *International Symposium on High Performance Computer Architecture*, February 2008.
8. Mau-Chung Frank Chang, Vwani P. Roychowdhury, Liyang Zhang, Hyunchol Shin, and Yongxi Qian. RF/wireless interconnect for inter- and intra-chip communications. *Proceedings of the IEEE*, 89(4):456–466, 2001.
9. Mau-Chung Frank Chang, Eran Socher, Sai-Wang Tam, Jason Cong, and Glenn Reinman. RF interconnects for communications on-chip. In *Proceedings of the 2008 international symposium on Physical design*, pages 78–83, New York, NY, USA, 2008. ACM.
10. Moses Charikar, Samir Khuller, David M. Mount, and Giri Narasimhan. Algorithms for facility location problems with outliers. In *Symposium on Discrete Algorithms*, pages 642–651, 2001.
11. Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979.

12. Anupam Gupta, Amit Kumar, Martin Pal, and Tim Roughgarden. Approximation via cost-sharing: a simple approximation algorithm for the multicommodity rent-or-buy problem. *Proceedings 44th Annual IEEE Symposium on Foundations of Computer Science*, pages 606–615, Oct. 2003.
13. Ron Ho, Kenneth W. Mai, and Mark A. Horowitz. The future of wires. *Proceedings of the IEEE*, 89(4):490–504, Apr 2001.
14. Kamal Jain, Mohammad Mahdian, and Amin Saberi. A new greedy approach for facility location problems. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 731–740, 2002.
15. Jon Kleinberg. The small-world phenomenon: An algorithmic perspective. In *Proceedings of the 32nd ACM Symposium on Theory of Computing*, pages 163–170, 2000.
16. Madhukar R. Korupolu, Charles G. Plaxton, and Rajmohan Rajaraman. Analysis of a local search heuristic for facility location problems. In *Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms*, 1998.
17. Sabato Manfredi, Mario di Bernardo, and Franco Garofalo. Small world effects in networks: An engineering interpretation. *Proceedings of the 2004 International Symposium on Circuits and Systems*, 4:IV–820–3 Vol.4, May 2004.
18. Chip Martel and Van Nguyen. Analyzing Kleinberg's (and other) small-world models. In *In 23rd ACM Symp. on Principles of Distributed Computing*, pages 179–188. ACM Press, 2004.
19. Adam Meyerson, Kamesh Munagala, and Serge Plotkin. Cost-distance: Two metric network design. *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 624–630, 2000.
20. Umit Y. Ogras and Radu Marculescu. "It's a small world after all": NoC performance optimization via long-range link insertion. *IEEE Transactions on Very Large Scale Integration Systems*, 14(7):693–706, July 2006.
21. Oskar Sandberg and Ian Clarke. The evolution of navigable small-world networks. Technical report, Chalmers University of Technology, 2007.
22. Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small world' networks. *Nature*, 393:440–442, 1998.