Fast and Robust Techniques for the Euclidean p-Median Problem with Uniform Weights

G. J. Lim[*] J. Reese[†] A. Holder[‡]

May 19, 2008

Abstract

We discuss new solution techniques for the p-median problem, with the goal being to improve the solution time and quality of current techniques. In particular, we hybridize the discrete Lloyd algorithm and the vertex substitution heuristic. We also compare three starting point techniques and present a new solution method that provides consistently good results when appropriately initialized.

Keywords: p-median, facility location, Lloyd Algorithm, vector quantization, vertex substitution

1 Introduction

The well studied p-median problem has applications in operations research, facility location, machine learning and graph theory [1, 8, 11, 12]. Hakimi developed the problem in the 1960s by generalizing his concept of the *absolute median* [4], which is a point on a network minimizing the sum of the weighted distances between that point and the vertices. Although such a point could lie anywhere (continuously) along the graph's edges, including vertices, Hakimi proved that there was always an optimal vertex solution [5]. This result established a discrete representation of the original continuous problem. Hakimi generalized the question to that of finding p medians, where the objective remained to minimize the sum of the weighted distances between the vertices and their nearest median [5]. Again, these medians could have been located anywhere along the edges of the graph, including vertices, but as in the case of the absolute median, a solution of p vertices was shown to exist. Hakimi developed these concepts to optimally locate switching centers in a communications network. Since his initial work, the p-median problem has been an integral part of location theory, becoming one of the most common models.

The p-median problem is NP-hard for general p but is polynomial if p is fixed [2, 8]. Because the integer programming formulation is computationally intractable on large problems, much of the literature focuses on (meta-)heuristics, which are cataloged in [12]. We concentrate on vertex substitution (VS), which is a well-studied p-median heuristic, and the discrete Lloyd algorithm (DLA), which is a discrete version of Lloyd's algorithm to design a vector quantizer [7, 9]. VS was introduced by Teitz and Bart [16] and works by iteratively swapping solution with non-solution vertices. VS is robust in terms of solution quality and requires less computation than integer programming approaches. DLA was first applied to the p-median problem in [7]. The technique converges rapidly, but as is often the case with heuristics, the speed often comes at the cost of solution quality. We form new hybrid techniques that combine VS and DLA that provide quality solutions in a timely fashion. Solutions from all methods depend on initialization, and we discuss three initialization schemes: *Random-start*, *Greedy-start*, and *Multi-start.*

The paper is organized as follows. Section 2 provides a mathematical description of the p-median problem, and Section 3 presents our solution techniques. Section 4 presents numerical results, and concluding thoughts are in Section 5.

[*]Department of Industrial Engineering, University of Houston. Corresponding author: ginolim@uh.edu
[†]PROS, Houston, Texas
[‡]Department of Mathematics, Trinity University

2 Problem Description

Let $(\mathbb{V}, \mathbb{E})$ be a connected, undirected network with vertex set $\mathbb{V} = \{v_1, \ldots, v_n\}$ and nonnegative vertex and edge weights w_i and l_{ij}, respectively. Further let $\gamma(v_i, v_j)$ be the length (distance) of the shortest path between vertices v_i and v_j with respect to the edge weights and $d(v_i, v_j) = w_i \gamma(v_i, v_j)$ be the weighted length (distance) of the corresponding shortest path. Notice d is not a metric since w_i is not necessarily w_j. This notation is extended so that for any set of vertices X, we have $\gamma(v_i, X) = \min\{\gamma(v_i, x) : x \in X\}$ and $d(v_i, X) = \min\{d(v_i, x) : x \in X\}$. With this notation, the p-median problem is

$$\min \left\{ \sum_{v_i \in \mathbb{V}} d(v_i, X) : X \subseteq \mathbb{V},\ |X| = p \right\}. \tag{1}$$

Hakimi's original work used a broader definition of d so that a solution could contain vertices and/or points along edges, but one of his main results showed that there was always a solution comprised of vertices. Most solution techniques exploit this fact and restrict the search to vertex solutions, a standard we uphold.

The p-median problem is commonly stated as a binary optimization problem, originally found in [14]. Allowing

$$\xi_{ij} = \begin{cases} 1 & \text{if vertex } v_i \text{ is allocated to vertex } v_j \\ 0 & \text{otherwise,} \end{cases}$$

a binary formulation is

$$\begin{array}{ll} \min & \sum_{ij} d(v_i, v_j)\xi_{ij} \\ \text{subject to} & \sum_j \xi_{ij} = 1, \text{ for } i = 1, \ldots, n, \\ & \sum_j \xi_{jj} = p, \\ & \xi_{jj} \geq \xi_{ij}, \text{ for } i, j = 1, \ldots, n,\ i \neq j, \\ & \xi_{ij} \in \{0, 1\}. \end{array} \tag{2}$$

A lower bound is available by relaxing the binary constraint with $0 \leq \xi_{ij} \leq 1$. This relaxation is often exact since it commonly solves with a binary solution [15].

3 Solution Techniques

In this section we succinctly present the algorithms and starting point methods that we consider. The initial discussion focuses on the established heuristics of VS and DLA, which are then combined into hybrid techniques.

3.1 Vertex Substitution

There are numerous variations of vertex substitution, most being slight modifications of the original Teitz and Bart procedure [16]. We consider implementations based on the work of Resende and Werneck [13], which are based on the ideas of Hansen and Mladenović [6] and Whitaker [17]. The difference in the techniques lies with how many comparisons are made before updating the collection of medians. Whitaker used a first profit strategy, meaning that the first profitable (improving) swap was performed. Hansen and Mladenović instead used a best profit strategy (as did the original Teitz and Bart algorithm), meaning that all possible swaps were evaluated and the best was performed.

Let $\mathcal{S}$ be a candidate set of medians and $\mathcal{N} = \mathbb{V} \backslash \mathcal{S}$ be the remaining vertices. We order $\mathcal{S} \times \mathcal{N}$ and calculate for any pair (v_r, v_i) the gain associated with inserting v_i into $\mathcal{S}$, creating a $p + 1$ candidate set of medians, and the respective loss of removing v_r from $\mathcal{S} \cup \{v_i\}$. Allowing $\mathcal{I}$ to be $\{v \in \mathbb{V} : \gamma(v, \mathcal{S}) = \gamma(v, v_r)\}$, we have that the profit is the difference between this gain and loss,

$$p(v_r, v_i) = \sum_{v \in \mathbb{V}} \max\{0,\ d(v, \mathcal{S}) - d(v, v_i)\} - \sum_{v \in \mathcal{I}} \left(d(v, \mathcal{S} \cup \{v_i\}) - d(v, v_r) \right).$$

We investigate implementations using either a first or best profit swap. In the former we iterate through $\mathcal{S} \times \mathcal{N}$ until $p(v_r, v_i)$ is positive, at which time we insert v_i into the candidate set of medians and remove v_r. In the later, we find a pair (v_r, v_i) that maximizes p and preform the swap with this pair. The worst case complexity for both implementations is $O(n^2)$. Pseudocode for the best profit implementation is presented in Algorithm 1. The differences for the first profit implementation are that in step 5 we would not investigate all pairs (v_r, v_i) to find the maximum but rather would stop once $p(v_r, v_i)$ is positive, and in step 7, (v_r^*, v_r^*) would be this pair. The subscript q on $\mathcal{S}$ indicates the iteration of the algorithm.

Algorithm 1 Pseudocode for Resende and Werneck's implementation of vertex substitution

```
1: procedure VertexSubstitution($\mathcal{S}_0$)
2:     $q \leftarrow 1$
3:     $\mathcal{S}_q \leftarrow \mathcal{S}_0$
4:     repeat
5:         $p_q^* \leftarrow \max\{p(v_r, v_i) : v_r \in \mathcal{S}_q, v_i \notin \mathcal{S}_q\}$
6:         if $p_q^* > 0$ then
7:             $\mathcal{S}_{q+1} \leftarrow (\mathcal{S}_q \cup \{v_i^*\}) \backslash \{v_r^*\}$, where $(v_r^*, v_i^*)$ is a solution to 5
8:             $q \leftarrow q + 1$
9:         end if
10:    until $p_q^* \leq 0$
11: end procedure
```

3.2 Discrete Lloyd Algorithm

DLA [7] is a modified version of the Lloyd algorithm, which is a heuristic addressing the optimal design of a vector quantizer [9]. Vector quantization is a continuous problem in $\mathbb{R}^k$, and we point readers to [3] for further information. DLA is instead applied to the discrete case of at most a countable number of vectors from $\mathbb{R}^k$. Assuming the common Euclidean metric, the authors of [7] showed that DLA applied to the complete graph of a discrete collection of vectors produces the same iterates as the Maranzana heuristic [10] for the p-median problem, provided that $l_{ij} = \|v_i - v_j\|^2$ and $w_i = 1$. Thus the original Lloyd algorithm, which was intended to design an optimal quantizer, had a natural interpretation as a heuristic to the p-median problem as long as the graph embedded in $\mathbb{R}^k$ and the vertex weights were equal. Moreover, DLA uses the special Euclidean properties to achieve rapid convergence, which follows because DLA replaces a $O(n^2)$ calculation in the Maranzana algorithm with a $O(n)$ process that projects a center-of-mass. Starting with an initial collection of possible medians $\mathcal{S}$, DLA iterates between partitioning the vertices into nearest neighbor cells, defined for $i = 1, 2, \ldots, p$ to be

$$\mathbb{V}_i = \{v \in \mathbb{V} : \gamma(v, v_i) \leq \gamma(v, v_k), \ \forall \ v_k \in \mathcal{S}\}, \tag{3}$$

and forming a new candidate set of medians by projecting the center of mass of each $\mathbb{V}_i$ onto a nearest element in $\mathbb{V}$, denoted by μ_i. A vertex v with the property that $|\{v_k : \gamma(v, v_k) = \gamma(v, \mathcal{S})\}| > 1$ is assigned to $\mathbb{V}_i$ with the lowest index. Pseudocode is found in Algorithm 2, where again q indicates the iteration of the algorithm.

The strength of DLA is that its worst case complexity is $O(pn)$, and since we commonly have $p << n$, the iteration count is nearly linear. The drawback is that the solution quality is sensitive to initialization and is often poor. DLA was used in [7] to initialize the best profit vertex substitution algorithm, but this technique proved ineffective. Indeed, with this initialization technique DLA underperformed random starts when comparing the final objective value in over 70% of the experiments.

3.3 Hybrid Approaches

VS and DLA update the candidate set of medians differently. Notice that VS searches $\mathcal{S} \times \mathcal{N}$ to make a single swap, and hence, VS changes a single candidate median per iteration. In contrast, DLA can change all candidate medians per iteration, something it accomplishes with reduced complexity. However, DLA

Algorithm 2 Structure of the original discrete Lloyd algorithm

```
1: procedure DISCRETELLOYDALGORITHM($\mathcal{S}_0$)
2:     $q \leftarrow 1$
3:     $\mathcal{S}_q \leftarrow \mathcal{S}_0$
4:     repeat
5:         Calculate $\mathbb{V}_{(i,q)}, \ \forall i$
6:         $\mathcal{S}_{q+1} \leftarrow \{\mu_{(i,q)} : i = 1, 2, \ldots, p\}$
7:         $q \leftarrow q + 1$
8:     until $\mathcal{S}_q = \mathcal{S}_{q+1}$
9: end procedure
```

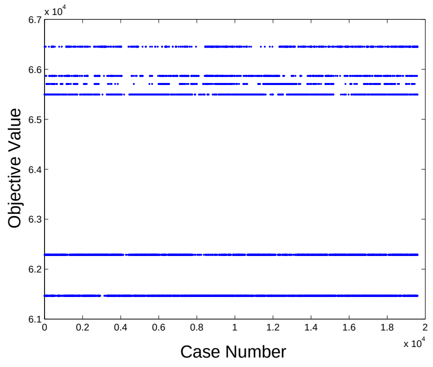

Figure 1: The objective value of VS with best profit swap versus all possible $\binom{50}{3}$ initiations.

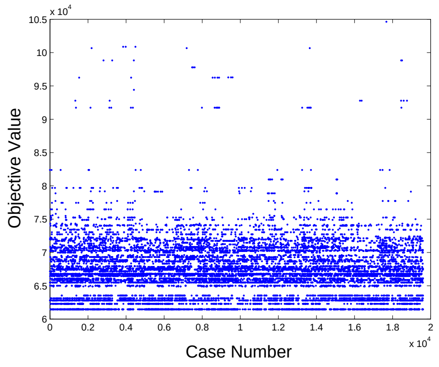

Figure 2: The objective value of DLA verses all possible $\binom{50}{3}$ initiations.

is more sensitive to its initialization than is VS, with DLA often returning poor quality solutions when compared to VS. As an example, we considered the 3-median problem on a uniformly distributed, 50 node network in $[0, 10] \times [0, 10]$ with $w_i = 1, \ \forall i$. We initiated both VS, with best profit swap, and DLA with each of the $\binom{50}{3}$ possible 3 element subsets. The global optimal solution of the problem was obtained by solving (2) to verify optimality. Figures 1 and 2 depict how the terminal objective value adjusts to the different initiations. Notice that VS terminated with only 6 different objective values, all of which were within 8.1% of the global optimum. Compared to DLA, which instead terminated with 105 different solutions within 70% of the global optimum, we see that VS more consistently returned quality solutions. Note that 98% of the DLA solutions are within 20% of the optimality. Figure 3 has another depiction of the frequency with which the techniques terminated with a certain value. In terms of quality, VS would be preferred to DLA, but this quality comes with the cost of increased computational time. Indeed, the average completion time for VS with best profit swap was about 100 times that of the average completion time for DLA.

What should not be overlooked is that just because DLA is more sensitive to initialization does not mean that it rarely returns quality solutions, and in fact, DLA found the global optimum with 8% of the initializations and was within the same range of VS 45.3% of the time. This begs the question of whether or not it might be possible to use the search iterations of VS to seed the expedient steps of DLA to improve the probability of DLA finding a quality solution, or *vice versa*. It is this question that supports the investigation into hybrid techniques that alternate between VS and DLA, and our numerical experiments show that such methods are capable of finding a quality solution faster than VS alone.

3.4 Geometric Stability of DLA

Before discussing the numerical results, it is natural to ask why DLA frequently terminates with a poor quality solution. The answer to this question is that DLA is somewhat geometrically stable, a fact

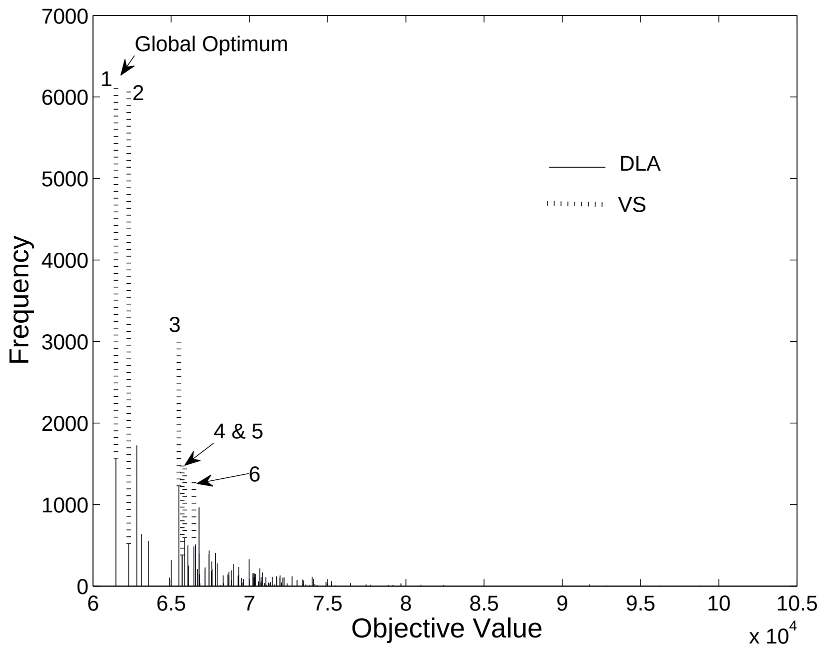

Figure 3: Comparison of terminal objective value for VS with best profit search and DLA.

highlighted by the following calculation. Consider the one dimensional problem where n nodes are i.i.d with respect to the uniform distribution on $[a, b]$. Let

$$\mathcal{S}_0 = \left\{ v_i = \left(a + \frac{b-a}{2p} \right) + \left(\frac{b-a}{p} \right) i : i = 1, 2, \dots p \right\}$$

be the initial collection of candidate medians and further let $q_0 = a$, $q_p = b$ and $q_i = (v_i + v_{i+1})/2$, for $i = 1, 2, \dots, p-1$. With this notation, the cells in (3) are

$$\mathbb{V}_i = \mathbb{V} \cap [q_{i-1}, q_i).$$

The new medians are found by projecting the center of mass of each of these cells onto $\mathbb{V}$, but if we ignore this projection and instead recalculate the cells with the centroids themselves, then the expected update to the boundary point q_i is

$$\left| q_i - \frac{1}{2} \left(\frac{\int_{q_i}^{q_{i+1}} x dx}{\int_{q_i}^{q_{i+1}} dx} + \frac{\int_{q_{i-1}}^{q_i} x dx}{\int_{q_{i-1}}^{q_i} dx} \right) \right| = 0, \tag{4}$$

where $i = 1, 2, \dots, p-1$. This calculation does not perfectly align with DLA since it ignores the projection. However, although the boundary elements q_i, for $i = 1, 2, \dots, p-1$, would be adjusted by this projection, the cells themselves may not since they only include the vertices within $[q_{i-1}, q_i)$. What (4) highlights is that if the distance between the centroids and their projections is small, then we don't expect much, if any, change from one iteration to the next. The geometric interpretation is that if the centroids are close to their projections, then only those vertices near a cell's border are likely to switch to another cell; if any switch at all. This discussion relates to the underlying Voronoi diagram associated with the centroids and essentially states that these diagrams stabilize quickly as DLA moves from one iteration to another. From this geometric perspective, we interpret DLA as a search in which the most significant adjustments occur in the first few iterations when the centroids may not be close to their projections. Figure 4 compares convergence speed between VS and DLA based on a problem instance with 500 nodes and $p = 5$. Both methods are initialized with a random starting point. As expected, DLA converges

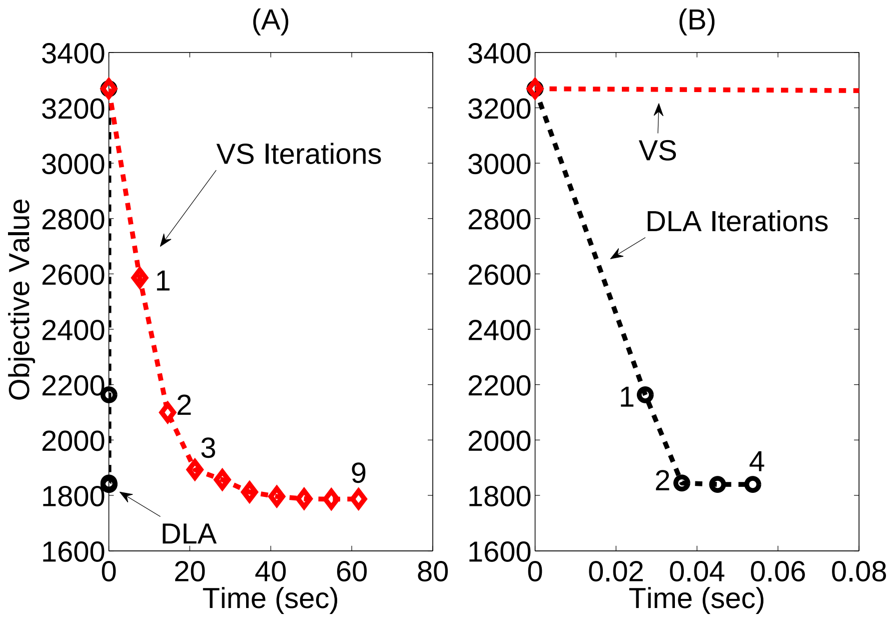

Figure 4: Convergence of DLA and VS for a problem instance with $|\mathcal{N}| = 500, p = 5$, and a random starting point.

to a local solution quickly, with most of the improvements being made in the first two iterations, see Figure 4(B). Alternatively, VS converges slower to a solution with a better objective value: 1,783 for VS and 1,839 for DLA, see Figure 4(A).

Unlike DLA, VS considers all possible swaps (in the best profit case) or depends on the ordering of $\mathcal{S} \times \mathcal{N}$ (in the first profit case), but in either situation, the updated collection of medians is not limited by the geometric stability of DLA. The cost to remove this limitation is that we have to possibly search the entirety of $\mathcal{S} \times \mathcal{N}$, something DLA forgoes. The hope in alternating between VS and DLA is to use the more computationally expensive search of VS to get geometrically "close" to a quality solution and then use DLA to quickly identify it. We continue the iteration consisting of M steps of VS in either first or best profit mode followed by N steps of DLA. Numerical results are presented in the next section.

4 Computational Comparisons

The purpose of this section is to compare performance (both solution speed and quality) between VS, DLA, and the hybrids of the previous section. We begin by discussing how the techniques were initialized.

4.1 Initialization Techniques

Figures 1 and 2 demonstrate that both DLA and VS depend on how they are initialized, and part of our numerical work is to explore different initialization schemes. We compare three starting-point techniques.

1. *Random-start.* This method calculates a random p-element subset of $\mathbb{V}$ and uses it as the initial set of candidate medians.

2. *Greedy-start.* This method calculates for each vertex the number of vertices within a radius r, and the vertex with the most vertices is added to the initial collection of candidate medians (ties are decided by lowest index). Vertices within a slightly smaller neighborhood of the selected vertex are removed from the selection process, and the process repeats until the pool is empty or an initial set is constructed. If the pool becomes empty before identifying p candidate medians, then vertices are added at random until completion.

Data Type	Node Size	Algorithm	Starting Point
Unclustered	250	DLA	Random-start
Clustered-R	500	VS	Greedy-start
Clustered-p	1000	DLA-FP	Multi-start
		DLA-BP	

Table 1: Combinations of Experiments

3. *Multi-start.* We use the expediency of DLA together with a multi-start technique to calculate candidate medians. This method initializes DLA with several randomly generated collections of candidate medians, each of which leads to a terminal collection after DLA ends. The terminal collection with the lowest objective is selected. This method is embarrassingly parallel, and would reduce time in a parallel environment.

The multi-start method is viewed as both a solution heuristic and as an initialization scheme for VS and the hybrids. This differs from how DLA initialized VS in [7], which started DLA with a single, random collection of medians. Again, the insight is that although DLA often gives poor solutions, it also regularly gives quality solutions, and since it is computationally inexpensive, it makes sense to spend a bit of time to search for an improved collection of candidate medians to seed another technique.

4.2 Design of Experiments

Although the general hybrid discussed in Section 3.3 layers an arbitrary number of DLA iterations with an arbitrary number of VS iterations, our experience indicates that representative results are typical with a few VS steps followed by a complete DLA heuristic, meaning that DLA runs until the median set stabilizes. The reason for completing DLA is that it typically takes only a few iterations to converge, and in fact, our numerical work shows that DLA typically converges in $(0.0015n + 5.738)$ steps. Multiple steps of VS are needed since each iteration changes a single node, and hence, to sufficiently alter the candidate medians we use more than a single step of VS before returning to DLA. We consider these four heuristics in our numerical comparisons.

DLA The discrete Lloyd algorithm.

VS Vertex substitution with best profit swap.

DLA-FP A hybrid with a complete DLA sequence and 3 iterations of VS with first profit swaps.

DLA-BP A hybrid with a complete DLA sequence and 2 iterations of VS with best profit swaps.

Each initialization scheme was paired with each heuristic, except that multi-start was not used to initialize DLA since multi-start is DLA with multiple random starts.

Table 1 shows the combinations of our experiments. The numerical comparisons were conducted on 90 random, 2-dimensional networks in $[0, 10] \times [0, 10]$, for which n was 250, 500 and 1000 and $p = 5$. The vertices for the first 30 graphs, referenced as *Unclustered*, were distributed uniformly over $[0, 10] \times [0, 10]$, whereas the vertices for the second 30, referenced by *Clustered-R*, were clustered by aggregating normal distributions, see Figures 5 and 6. Note that the number of clusters in the clustered data is determined randomly and is not necessarily the same as p. We added a third category of 30 graphs called *Clustered-p* that assumes $R = p$.

The heuristics and initialization methods were implemented in MATLAB 7.0. The radius for *Greedy-start* was $r = \sqrt{pM/2\pi}$, where M was the largest vertex coordinate. This value is based on the hypothesis that we want to roughly divide the space defined by the data into equally sized circles. The value of the removal radius was $(0.8)r$, a value that gave increased stability over r itself. The number of p-element, random samples for the multi-start technique was investigated. Figure 7 shows that the average objective value decreases as the number of samples increases, which is not surprising. We decided to use $0.05n$ as the number of random initializations, a value that consistently provided quality and efficiency. Stability

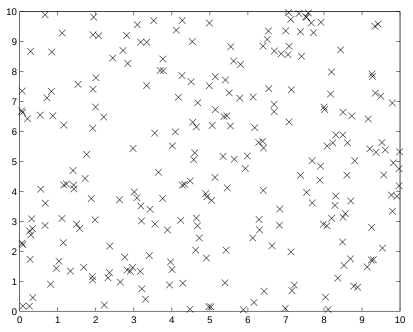
Figure 5: An example of an unclustered data set

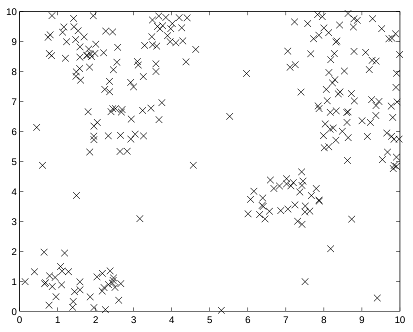
Figure 6: An example of a clustered data set

		DLA		DLA-FP		DLA-BP		VS	
250	LP Obj: 853.68	DLA		DLA-FP		DLA-BP		VS	
	LP Time: 13.10	Time	Obj	Time	Obj	Time	Obj	Time	Obj
	Random-start	0.03	920.82	1.46	907.04	9.29	864.48	14.62	869.65
	Greedy-start	0.02	894.50	1.40	894.50	9.41	858.50	13.37	867.51
	Multi-start	0.39	859.77	2.07	859.56	5.87	855.55	5.84	856.66
500	LP Obj: 1710.50	DLA		DLA-FP		DLA-BP		VS	
	LP Time: 148.67	Time	Obj	Time	Obj	Time	Obj	Time	Obj
	Random-start	0.07	1789.87	4.43	1777.46	39.78	1736.69	52.76	1731.48
	Greedy-start	0.04	1773.35	4.41	1773.35	28.76	1736.17	48.23	1733.53
	Multi-start	1.46	1717.73	5.79	1717.07	14.74	1713.28	16.34	1714.64
1000	LP Obj: 3455.11	DLA		DLA-FP		DLA-BP		VS	
	LP Time: 2173.13	Time	Obj	Time	Obj	Time	Obj	Time	Obj
	Random-start	0.15	3552.21	15.70	3569.39	143.72	3482.04	201.19	3498.22
	Greedy-start	0.09	3557.68	17.13	3556.81	165.05	3483.16	202.59	3492.25
	Multi-start	7.83	3459.77	22.11	3459.52	63.87	3457.43	63.30	3457.33

Table 2: Average values on 30 *Unclustered* data sets, for $n = 250, 500, 1000$, $p = 5$

of the candidate set of medians was used as the termination criteria, where this check was only performed after the VS iterations in DLA-FP and DLA-BP.

Recall that the relaxation of (2) is often exact [15], but even if it is not, it provides a lower bound on the objective. We found that we could solve this relaxation in reasonable time with CPLEX 10.0 for our problems. For each heuristic and initialization scheme, the average objective value was no worse than 1.5% of the relaxed optimal value. This is somewhat surprising with regards to randomly initialized DLA since Figure 2 indicates that DLA often leads to significant duality gaps. However, the end result is that DLA counters its poor solutions with quality solutions, and that on average, the quality solutions outnumber the poor solutions.

4.3 Numerical Results

Tables 2, 3 and 4 display our numerical results for *Unclustered, Clustered-R* and *Clustered-p*, respectively, where each value is the average over the 30 respective problems. Figure 12 shows the average time of each solution technique initialized with multi-start. Likewise, Figure 13 shows the ratio of the average heuristic objective value to the average LP relaxation objective value, again each solution technique was initialized with multi-start.

The results show a clear indication that DLA outperforms the other heuristics with regards to speed (see Figures 8, 10, 14, 16, 20 and 22). Both hybrids (DLA-FP and DLA-BP) run faster than VS in all tests. The solution quality of DLA with *Random-start* or *Greedy-start* is not as consistent as DLA with *Multi-start* as evidenced by Figures 9, 11, 15, 17, 21, and 23. Note that DLA-BP often performs better than VS in both computational time and solution quality. The speed of DLA-FP is attractive, but its

	LP Obj: 447.69	DLA		DLA-FP		DLA-BP		VS	
	LP Time: 9.37	Time	Obj	Time	Obj	Time	Obj	Time	Obj
250	Random-start	0.03	549.36	1.79	494.20	7.23	460.44	12.80	459.51
	Greedy-start	0.03	538.91	1.51	498.04	5.58	460.18	15.34	458.22
	Multi-start	0.41	449.76	1.73	449.76	4.48	449.38	4.71	449.38
	LP Obj: 1149.43	DLA		DLA-FP		DLA-BP		VS	
	LP Time: 99.62	Time	Obj	Time	Obj	Time	Obj	Time	Obj
500	Random-start	0.10	1248.30	5.64	1243.61	24.11	1165.71	52.06	1174.03
	Greedy-start	0.04	1248.30	4.97	1266.12	27.04	1172.48	53.10	1169.53
	Multi-start	1.49	1152.58	7.56	1151.26	16.00	1149.96	17.78	1151.35
	LP Obj: 2496.81	DLA		DLA-FP		DLA-BP		VS	
	LP Time: 1381.28	Time	Obj	Time	Obj	Time	Obj	Time	Obj
1000	Random-start	0.20	2596.92	18.22	2591.73	130.65	2550.47	241.66	2525.87
	Greedy-start	0.17	2592.65	16.48	2592.47	140.31	2508.13	234.39	2511.61
	Multi-start	7.84	2498.35	24.00	2498.35	54.18	2498.35	65.64	2497.77

Table 3: Average values on 30 *Clustered-R* data sets, for $n = 250, 500, 1000$, $p = 5$

	LP Obj: 122.0	DLA		DLA-FP		DLA-BP		VS	
	LP Time: 4.39	Time	Obj	Time	Obj	Time	Obj	Time	Obj
250	Random-start	0.02	277.80	2.40	129.45	4.02	122.00	10.58	122.00
	Greedy-start	0.02	185.38	1.83	128.52	2.99	122.00	10.67	122.00
	Multi-start	0.31	122.00	1.59	122.00	1.94	122.00	3.24	122.00
	LP Obj: 248.52	DLA		DLA-FP		DLA-BP		VS	
	LP Time: 45.62	Time	Obj	Time	Obj	Time	Obj	Time	Obj
500	Random-start	0.13	444.71	7.89	283.78	13.15	248.52	36.08	248.52
	Greedy-start	0.07	347.43	6.69	258.71	9.62	248.52	39.35	248.52
	Multi-start	1.62	248.52	5.51	248.52	6.13	248.52	11.17	248.52
	LP Obj: 497.79	DLA		DLA-FP		DLA-BP		VS	
	LP Time: 445.97	Time	Obj	Time	Obj	Time	Obj	Time	Obj
1000	Random-start	0.12	897.32	30.57	561.68	42.67	497.79	135.37	497.79
	Greedy-start	0.11	843.75	22.94	562.42	37.73	497.79	143.45	497.79
	Multi-start	4.67	497.79	25.08	497.79	22.84	497.79	41.95	497.79

Table 4: Average values on 30 *Clustered-p* data sets, for $n = 250, 500, 1000$, $p = 5$

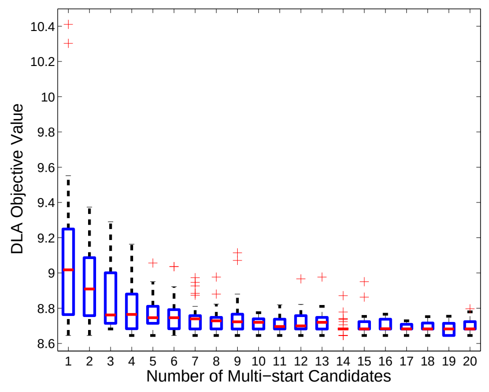

Figure 7: Box plot of objective values resulting multi-start with different number of samples

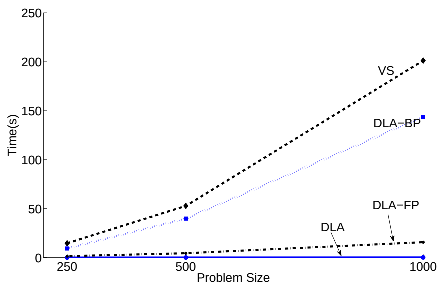

Figure 8: Average time with *Random-start* plotted against problem size for *Unclustered* data

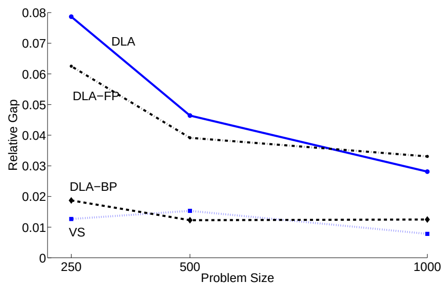

Figure 9: Relative duality gap for the relaxed LP with *Random-start* plotted against problem size for *Unclustered* data

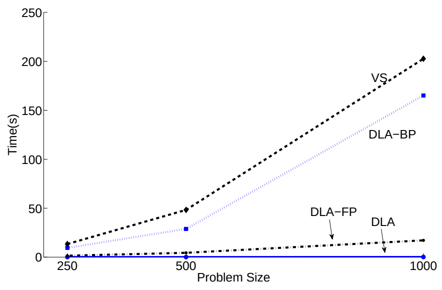

Figure 10: Average time with *Greedy-start* plotted against problem size for *Unclustered* data

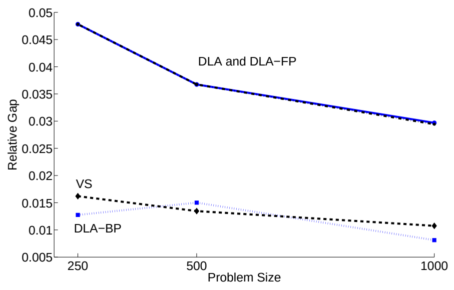

Figure 11: Relative duality gap for the relaxed LP with *Greedy-start* plotted against problem size for *Unclustered* data

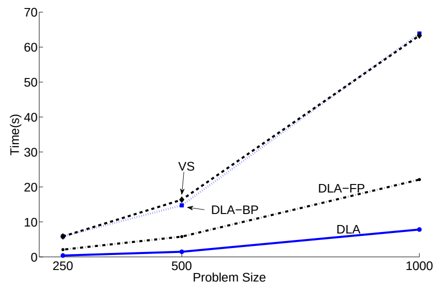

Figure 12: Average time with *Multi-start* plotted against problem size for *Unclustered* data

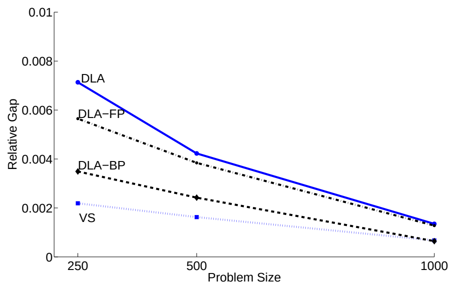

Figure 13: Relative duality gap for the relaxed LP with *Multi-start* plotted against problem size for *Unclustered* data

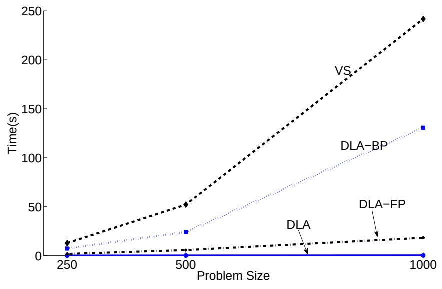

Figure 14: Average time with *Random-start* plotted against problem size for *Clustered-R* data

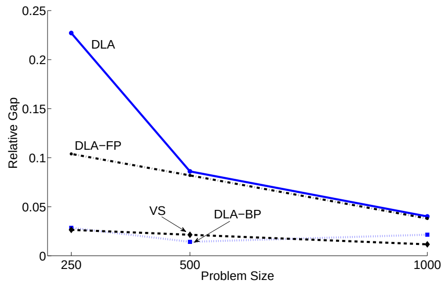

Figure 15: Relative duality gap for the relaxed LP with *Random-start* plotted against problem size for *Clustered-R* data

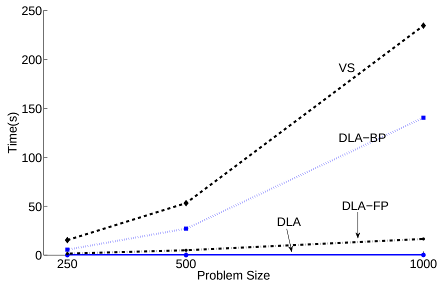

Figure 16: Average time with *Greedy-start* plotted against problem size for *Clustered-R* data

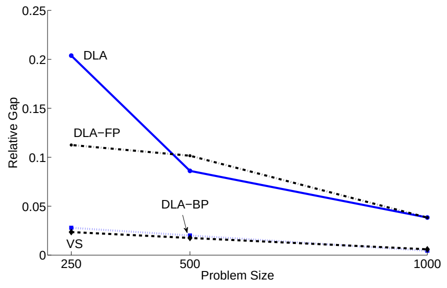

Figure 17: Relative duality gap for the relaxed LP with *Greedy-start* plotted against problem size for *Clustered-R* data

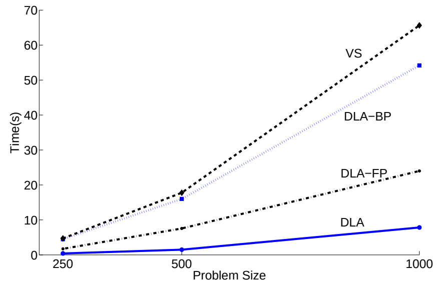

Figure 18: Average time with *Multi-start* plotted against problem size for *Clustered-R* data

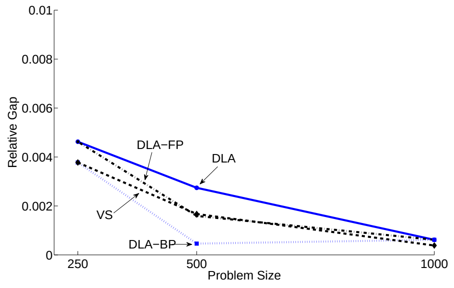

Figure 19: Relative duality gap for the relaxed LP with *Mutli-start* plotted against problem size for *Clustered-R* data

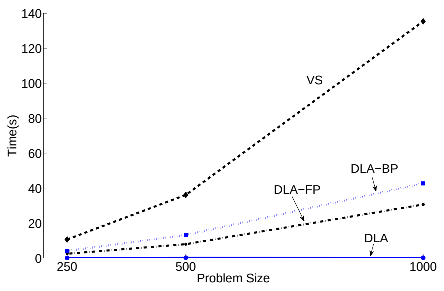

Figure 20: Average time with *Random-start* plotted against problem size for *Clustered-p* data

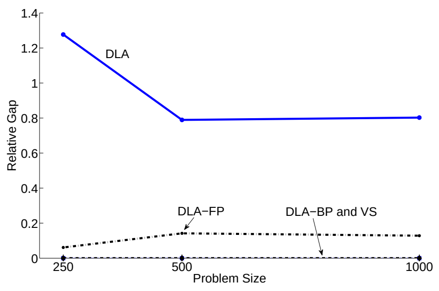

Figure 21: Relative duality gap for the relaxed LP with *Random-start* plotted against problem size for *Clustered-p* data

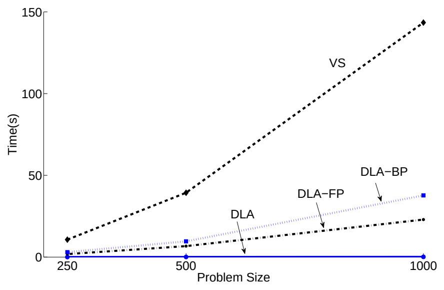

Figure 22: Average time with *Greedy-start* plotted against problem size for *Clustered-p* data

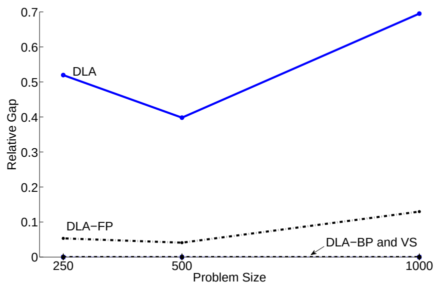

Figure 23: Relative duality gap for the relaxed LP with *Greedy-start* plotted against problem size for *Clustered-p* data

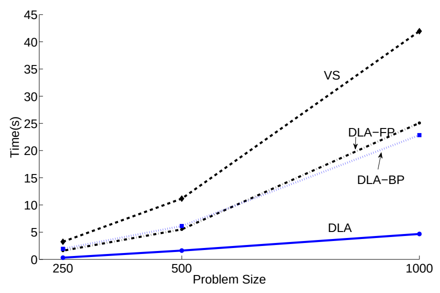

Figure 24: Average time with *Multi-start* plotted against problem size for *Clustered-p* data

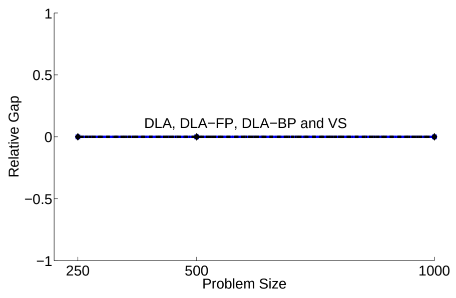

Figure 25: Relative duality gap for the relaxed LP with *Multi-start* plotted against problem size for *Clustered-p* data

solution quality is below that of DLA-BP and VS. The highlight of the numerical work is that *Multi-start* significantly outperformed both *Random-start* and *Greedy-start* with regards to solution quality in all tests. All four algorithms converged to solutions within a 0.7% dual gap, which is almost globally optimal for all three cluster types. *Multi-start* shines when the data is already clustered. For *Clustered-p*, all four algorithms converged to global optimality when initialized with *Multi-start* (see Figure 25).

5 Conclusion

We investigated hybrid solution methods for the p-Median problem that combine elements of DLA and VS. These methods perform well when initialized with *Multi-start*, delivering superior solution quality in an efficient manner. Our experiments show the new hybrids outperform best profit vertex substitution, which is widely regarded as one of the most robust solution techniques. The overriding trend is that the DLA with *Multi-start* provides a robust and efficient solution heuristic. A direction of future research is to use the speed of DLA to address stochastic variants that require numerous solves to evaluate scenarios.

Acknowledgment This research is supported in part by the University of Houston GEAR Grants #91063. We thank Ms. Ayse Durukan for her help with some of the numerical experiments presented in this paper.

References

[1] M. S. Daskin. *Network and Discrete Location: models, algorithms, and applications.* John Wiley & Sons, Inc., New York, 1995.

[2] M. R. Garey and D. S. Johnson. *Computers and Intractibility: A guide to the theory of NP-completeness.* W. H. Freeman and Co., San Francisco, 1979.

[3] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression.* Kluwer International Series in Engineering and Computer Science. Kluwer Academic Publishers, 1992.

[4] S. L. Hakimi. Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations Research*, 12(3):450–459, 1964.

[5] S. L. Hakimi. Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Operations Research*, 13(3):462–475, 1965.

[6] P. Hansen and N. Mladenović. Variable neighborhood search for the p-median. *Location Science*, 5(4):207–226, 1997.

[7] A. Holder, G. Lim, and J. Reese. The relationship between discrete vector quantization and the p-median problem. Technical Report 102, Trinity University, San Antonio, TX, 2007.

[8] O. Kariv and S. L. Hakimi. An algorithmic approach to network location problems. II. The p-medians. *SIAM Journal on Applied Mathematics*, 37(3):539–560, 1979.

[9] S. P. Lloyd. Least squares quantization in PCM. *IEEE Trans. Inform. Theory*, 28:129–137, 1982.

[10] F. E. Maranzana. On the location of supply points to minimize transport costs. *Operational Research Quarterly*, 15(3):261–270, 1964.

[11] P. B. Mirchandani and R. Francis, editors. *Discrete location theory.* Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, Inc., New York, 1990.

[12] J. Reese. Solution methods for the p-median problem: An annotated bibliography. *Networks*, 48(3):125–142, 2006.

[13] M. G. C. Resende and R. F. Werneck. On the implementation of a swap-based local search procedure for the p-median problem. In R. Ladner, editor, *ALENEX '03: Proceedings of the Fifth Workshop on Algorithm Engineering and Experiments*, pages 119–127, Philadelphia, 2003. SIAM.

[14] C. ReVelle and R. Swain. Central facilities location. *Geographical Analysis*, 2:30–42, 1970.

[15] K. E. Rosing, C. S. ReVelle, and H. Rosing-Vogelaar. The p-median and its linear programming relaxation: An approach to large problems. *Journal of the Operational Research Society*, 30(9):815–823, 1979.

[16] M. B. Teitz and P. Bart. Heuristic methods for estimating the generalized vertex median of a weighted graph. *Operations Research*, 16(5):955–961, 1968.

[17] R. A. Whitaker. A fast algorithm for the greedy interchange of large-scale clustering and median location problems. *INFOR*, 21(2):95–108, 1983.