Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A New Meta-Heuristic for Improving General Multi-Start Procedures

Zvi Drezner

zdrezner@fullerton.edu

California State University, Fullerton

Jack Brimberg

Royal Military College of Canada

Research Article

Keywords: Meta-heuristic, Multi-start procedure, p-median

Posted Date: November 11th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11461178/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

A New Meta-Heuristic for Improving General Multi-Start Procedures

Zvi Drezner
College of Business and Economics
California State University-Fullerton
Fullerton, CA 92834.
e-mail: zdrezner@fullerton.edu

Jack Brimberg
Dept. of Mathematics & Computer Science
Royal Military College of Canada
Kingston, ON Canada.
e-mail: Jack.Brimberg@rmc.ca

Abstract

In this paper we propose a new and effective meta-heuristic, named POST, that improves the performance of multi-start improvement algorithms. It is a general follow-up (or post-optimizing) procedure that attempts to find a better final solution than the best one found by any multi-start approach. The additional required effort and run time is negligible.

The meta-heuristic is tested on the planar p-median multi-facility location problem which is known to have many (perhaps millions for the instances tested in this paper) local optima. Forty-eight well researched instances were tested, and the meta-heuristic improved the best known solution for all of them, in about the same run time required by the multi-start approach.

Key Words: Meta-heuristic; Multi-start procedure; p-median.

1 Introduction

We propose a meta-heuristic, termed POST, that attempts to find a better solution for any mathematical model that is solved by multi-start approaches (for example, György and Kocsis, 2011; Martí et al., 2025, 2010, 2013). A multi-start approach is based on repeating an improvement algorithm (IMP) many times, and selecting the best found outcome.

Examples of improvement algorithms include: a gradient-based algorithm (Chen, 1983; Murtagh and Niwattisyawong, 1982), Simulated annealing (Kirkpatrick et al., 1983; Moran, 2003), Tabu search (Glover and Laguna, 1997; Moreno-Perez et al., 1994), Variable neighborhood search (Drezner et al., 2015a; Hansen et al., 2009; Hansen and Mladenović, 1997; Hansen and Mladenović, 2001).

One component of the POST procedure is generating offspring from pairs of "parents" (solutions found by IMP), which is an essential component of designing genetic algorithms (Goldberg, 2006; Holland, 1975; Katoch et al., 2021). Genetic algorithms are also improvement algorithms that can

be used in a multi-start approach, but usually with fewer starting solutions because each application of a genetic algorithm requires relatively long computational time.

The paper is organized as follows. Section 2 presents the POST procedure, and provide its details. Section 3 details the implementation of POST for solving heuristically the p-median location problem. Computational results on solving 48 large instances of the planar p-median problem are presented and discussed in Section 4. Section 5 provides conclusions and some suggestions for further research.

2 The POST Procedure

Multi-start heuristic algorithms generate N, for example $N = 100,000$, starting solutions either randomly or by design, and apply an improvement algorithm (IMP) on each. The best found solution is the result of the multi-start approach. An equivalent approach is to repeat the process $L \geq 1$ times, each time form $\frac{N}{L}$ starting solutions, getting L "best found" solutions, and selecting the best one. The outcome is the same as the best result of all N starting solutions. This structure may be useful if the procedure is executed in parallel on L computers or different users.

In the proposed POST meta-heuristic it can be advantageous to apply the multi-start algorithm $L > 1$ times from $I \approx \frac{N}{L}$ starting solutions. As reported in the computational experiments in Section 4, $L > 1$ provided better results, on average, than applying POST on the standard approach ($L = 1$), and the performance tends to improve as L increases.

2.1 List of POST Parameters

N: The total number of times that the improvement algorithm is executed in the multi-start approach. The average run time for a single exacution of the improvement algorithm (IMP) is estimated, and a goal of the total run time can be used to determine N.

$L \geq 1$: The total number of N executions of repeating IMP, is divided into L subsets of repeating IMP in each.

I: The number of executions of IMP in each of the L subsets. $I \approx \frac{N}{L}$.

$K \geq 2$: The number of parents selected from I final solutions of IMP in a subset. In our experiments we selected the $K = 5$ best different solutions. The number of pairs of solutions (pairs of parents) is $\frac{K(K-1)}{2}$.

E: The elite set is defined as the best K different solutions out of the I IMP solutions.

$T \geq 1$: The number of offspring generated from each pair of parents. In some procedures of generating offspring we may be able to generate only $T = 1$ offspring per pair of parents.

P: The total number of offspring generated for each subset of I IMP executions. $P = \frac{K(K-1)}{2}T$.

2.2 The POST Algorithm

1. Set the counter $\ell = 1$, and keep track on the best found solution.

2. A multi-start improvement algorithm (a heuristic) is repeated from I different starting solutions, and the best K non-repeating solutions are selected to form an elite set E.

3. $T \geq 1$ offspring for each of the $\frac{K(K-1)}{2}$ pairs of solutions from E are created for a total of $P = \frac{K(K-1)}{2}T$ offspring.

4. The improvement algorithm is applied on each of the P offspring and the best K non-repeating solutions are selected and saved.

5. If the best found solution has been improved, update the best found solution, and return to Step 3, applying the newly selected K solutions, (i.e., the current E).

6. Otherwise, set $\ell = \ell + 1$. If $\ell \leq L$ return to Step 2; else stop.

An optional extra step: Throughout the POST algorithm, the best K results of the total number $IL \approx N$ IMP executions (before applying steps 3 to 5) are retained, and steps 3 to 5 applied on them. In our computational experiments, the optional extra step rarely found a better solution, and thus is considered optional. However, it requires a negligible extra run time and hence we applied it in the experiments.

2.3 Discussion

The execution time of the improvement algorithm on an offspring starting solution (in the elite set E), is usually much faster than applying it on random starting solutions because the starting solutions in E are of higher quality and require fewer iterations of the improvement algorithm. The number of extra applications of the improvement algorithm is about PL which is a fraction $\frac{PL}{N} \approx \frac{P}{I}$ of the multi-start applications, which is usually small. In the compuational experiments we applied $I \geq 1,000$, so we opted for $\frac{P}{L}$ not to exceed 20%, and selected $P = 200$ by $K = 5$ and $T = 10$. The additional execution time for improving the offspring is relatively short, and the POST algorithm does not require a significant longer run time unless L is large. So, for $L = 80$ we ran POST for only $L = 70$ subsets.

Note that Path Relinking (Glover et al., 2000) is another procedure that employs an "elite set" of solutions. However, this procedure is quite different from the crossover operation (performed in GA) that we have adapted in POST. Path Relinking constructs a path of intermediate solutions joining pairs of solutions in the elite set in a deterministic way. Only attributes that are different between the selected pair are changed one at a time until the origin node in the pair is transformed at the other end of the path to the second node (guide node) in the pair. The intention is to intensify the search in neighborhoods close to the selected pair. POST, on the other hand, joins pairs of solutions in the elite set in a random fashion to produce several offspring. The improvement algorithm is applied to all of them. If improved solutions are found, new elite sets are formed and more generations of offspring are created. The whole process is repeated L times resulting in a highly diversified and extensive search through promising regions of the feasible set.

If the multi-start heuristic finds the optimal solution, then of course POST cannot improve it. If the POST does not improve the multi-start heuristic result it is possible (but definitely not certain) that the multi-start procedure identified the optimal solution or a solution very close to optimality.

We tested the POST algorithm on the planar p-median problem (see Sections 3 and 4 for details). It can be applied for solving models that are usually solved by a multi-start approach such as the minimum sum of squares clustering problem (for example: Aloise, 2009; Aloise et al., 2012;

Bagirov et al., 2015; Costa et al., 2017; Gribel and Vidal, 2019; Kalczynski et al., 2022; Ordin and Bagirov, 2015; Pereira et al., 2018; Zhou et al., 2020, 2021), and the clique partitioning problem (for example, Brimberg et al., 2017; Brusco and Köhn, 2009; Grötschel and Wakabayashi, 1989; Kalczynski et al., 2023).

We anticipate that researchers who apply multi-start heuristic algorithms to solve any optimization model, will recognize the effectiveness of the POST meta-heuristic, and will experiment with POST in future research.

3 Applying POST for Solving the Planar p-Median Location Model

The planar p-median problem (Dantrakul et al., 2014; Drezner and Salhi, 2017), also known as the multi-source Weber problem (Brimberg et al., 2000; Hansen et al., 1998; Kuenne and Soland, 1972), or continuous location-allocation problem (Brimberg et al., 2008; Love et al., 1988), is an extension of the single facility Weber problem (Church et al., 2025; Weber, 1909). It is defined as follows. A given set of n demand points are located at $A_i = (a_i, b_i)$, with associated weights $w_i > 0$, for $i = 1, \ldots n$. We need to find the locations of p facilities, $X_j = (x_j, y_j)$ for $j = 1, \ldots, p$, that solve:

$$\min_{X} \left\{ f(X) = \sum_{i=1}^{n} w_i \min_{1 \le j \le p} \{d(X_j, A_i)\} \right\} , \tag{1}$$

where $X = (X_1, \ldots, X_p)$ denotes the vector of $2p$ decision variables. The function $d(X_j, A_i)$ is the distance, typically Euclidean, between X_j and A_i. For Euclidean distances, investigated in this paper: $d(X_j, A_i) = \sqrt{(x_j - a_i)^2 + (y_j - b_i)^2}$.

There are two main variants of the p-median problem. In the discrete version it is assumed that there is a given finite set of possible locations of the facilities (Daskin, 1995; Daskin and Maass, 2015). A special discrete case is when demand points are located at the vertices of a network with the distance measured along the links of the network. In the continuous version, tested in this paper, it is assumed that facilities can be located anywhere in the plane, and thus there are infinite possible locations for the facilities. In this paper, the term p-median refers to the continuous version, with Euclidean distances applied in equation (1).

This planar p-median was originally proposed by Cooper (1963, 1964) who proposed the ALT

algorithm as a heuristic solution approach. The problem was shown to be NP-hard by Megiddo and Supowit (1984). Krau (1997) designed an algorithm for finding the optimal solution (within an accuracy of $\epsilon > 0$) for relatively small problems. The p-median problem is extremely non-convex. Brimberg et al. (2010) and Brimberg and Drezner (2013) tested an instance with 50 demand points introduced in Eilon et al. (1971). Using 10,000 random starts of the ALT algorithm, that is known to terminate at a local minimum, they found for $p = 10, 15, 20$, and 25, the respective number of different local minima of 4,701, 8,727, 9,164, 9,035. The actual number of local minima may be much higher. Also note that a high variance in solution quality was observed. The problem instances tested in this paper (with up to 10,000 demand points and 100 facilities) may have millions of local optima. For further reviews of the p-median model see Brimberg et al. (2000); Brimberg and Hodgson (2011); Brimberg and Salhi (2019); Drezner et al. (2015b, 2016, 2024).

We applied two recent improvement algorithms that performed very well for solving the p-median problem. Drezner and Brimberg (2026a) proposed an effective procedure (SPL for split) to generate "good" starting solutions. A new improvement algorithm, CLUST (Drezner and Brimberg, 2026b, also described in the Appendix for convenience), is applied on the SPL starting solution. It produced better results than the ALT improvement algorithm and its extensions (Brimberg and Drezner, 2013, 2021).

3.1 Generating Offspring for Solving the Planar p-Median Model

Two configurations (parents) are selected. An offspring is created by selecting several facility locations from one parent and the rest from the other one. We propose, similar to the H2M algorithm (Drezner and Brimberg, 2026a), to draw lines through the center of gravity of the combined configuration (of $2p$ facilities), and join the facilities from the first parent that are located on one side of the line with facilities of the second parent on the other side. A similar idea to create offspring in a genetic algorithm, with only one line, is proposed in Drezner et al. (2015b), and for a different model (selecting one-way and two-way links in designing a network) in Drezner and Salhi (2002).

The two parents define $2p$ locations for facilities (some of them may coincide). Let (x_c, y_c) be the center of gravity of the $2p$ facility locations. For an angle θ, we adjust the standard definition

of the line passing through (x_c, y_c), $y = y_c + (x - x_c)\tan\theta$ by multiplying it by $\cos\theta$ to avoid an infinity issue for $\theta = \frac{\pi}{2}, \frac{3\pi}{2}$, and also to simplify Property 1 below. The equivalent line formula is:

$$(y - y_c)\cos\theta - (x - x_c)\sin\theta \equiv d(x, y) = 0 \ . \tag{2}$$

The left hand side of (2) is defined as $d(x, y)$ whose absolute value is the distance from the line as shown in Property 1. Points with $d(x, y) = 0$ are on the line, points with $d(x, y) > 0$ are on one side of the line, and points with $d(x, y) < 0$ are on the other side of the line. The following property is a simplified version of a well known and widely used formula.

Property 1: *The distance of point* (x, y) *from the line is* $|d(x, y)|$.

Proof: The suggested distance formula (for example in Ballantine and Jerbert, 1952) is simplified because $\sin^2\theta + \cos^2\theta = 1$. □

The values of $d(x, y)$ from one parent and $-d(x, y)$ from the second parent (a total of $2p$ values) are calculated. All the points with a positive value are either on one side of the line from one parent or on the other side of the line from the second parent. It is expected to have about p positive values and p negative values. The p points with the largest values are selected for the offspring. If there is a point common to both parents, it will be selected only once because one will be positive and one negative. It is possible, but very unlikely, to have duplicate locations when there are fewer than p positive values. However, the improvement algorithm will remove one of the facilities in the first iteration because the value of the objective function does not change if two facilities are co-located, and one of them is removed. The other one can be located at a different location attracting a subset of demand points thus improving the value of the objective function.

It is suggested to select $\theta_j = \frac{j-1}{T}2\pi$ for $j = 1, \ldots T$, and create T (a parameter) offspring. Note that $\theta_j + \pi$ generates the same line, but reverses which parent is on a specific side of the line. These two values of θ are selected if T is an even number but the offspring are different. Therefore, there is no need to consider a pair j_2, j_1 when pair j_1, j_2 is selected.

Table 1: Results by POST for 24 uniform instances

		Best	I =1,000, $L = 70$ (POST1)			I =10,000, $L = 8$ (POST10)			I =80,000, $L = 1$ (POST80)		
n	p	Known*	Value+	†	‡	Value+	†	‡	Value+	†	‡
1000	50	71.8130	**71.8014**	**-0.016%**	84.19	71.8171	0.006%	83.52	71.8130	0.000%	81.28
1000	60	63.9879	63.9751	**-0.020%**	99.89	**63.9736**	**-0.022%**	98.98	63.9879	0.000%	96.53
1000	70	57.6391	**57.5821**	**-0.099%**	104.95	57.5969	**-0.073%**	105.73	57.6375	**-0.003%**	105.65
1000	80	52.8462	**52.7813**	**-0.123%**	110.46	52.8448	**-0.003%**	111.00	52.8236	**-0.043%**	110.66
1000	90	49.0128	48.9435	**-0.141%**	117.93	**48.9341**	**-0.161%**	119.56	49.0128	0.000%	117.77
1000	100	45.7834	45.6541	**-0.282%**	135.28	**45.6437**	**-0.305%**	131.11	45.7096	**-0.161%**	128.92
2000	50	150.7560	**150.7339**	**-0.015%**	213.45	150.7550	**-0.001%**	207.38	150.7560	0.000%	204.57
2000	60	135.8410	**135.7747**	**-0.049%**	246.13	135.7864	**-0.040%**	240.96	135.8410	0.000%	236.98
2000	70	123.8238	**123.8204**	**-0.003%**	268.77	123.8234	**0.000%**	265.46	123.8238	0.000%	260.94
2000	80	115.0461	**114.9689**	**-0.067%**	309.02	115.0027	**-0.038%**	299.79	115.0348	**-0.010%**	295.01
2000	90	107.6620	**107.5079**	**-0.143%**	349.61	107.5305	**-0.122%**	336.16	107.5451	**-0.109%**	330.92
2000	100	101.2800	101.1083	**-0.170%**	370.25	**101.1014**	**-0.176%**	355.55	101.1546	**-0.124%**	346.14
5000	50	391.6220	**391.5846**	**-0.010%**	659.39	391.5883	**-0.009%**	614.46	391.5921	**-0.008%**	600.10
5000	60	355.3980	**355.3818**	**-0.005%**	764.24	355.3876	**-0.003%**	732.49	355.3980	0.000%	721.33
5000	70	327.4487	**327.3036**	**-0.044%**	902.68	327.3438	**-0.032%**	858.74	327.3542	**-0.029%**	842.82
5000	80	304.4473	304.4081	**-0.013%**	978.07	**304.3666**	**-0.027%**	971.34	304.3823	**-0.021%**	958.57
5000	90	285.0280	284.8145	**-0.075%**	1111.19	284.8265	**-0.071%**	1038.70	**284.7887**	**-0.084%**	1021.56
5000	100	269.1664	**268.7861**	**-0.141%**	1249.82	268.8725	**-0.109%**	1179.29	269.0321	**-0.050%**	1145.84
10000	50	798.6799	798.6849	0.001%	1272.44	798.6839	0.001%	1192.12	**798.6774**	**-0.000%**	1167.11
10000	60	725.1684	725.1450	**-0.003%**	1361.03	**725.1194**	**-0.007%**	1311.06	725.2119	0.006%	1286.88
10000	70	669.9168	**669.8753**	**-0.006%**	1683.55	669.9191	0.000%	1560.92	669.8918	**-0.004%**	1548.95
10000	80	624.7508	624.6261	**-0.020%**	1891.94	624.6285	**-0.020%**	1800.11	**624.6056**	**-0.023%**	1783.51
10000	90	586.7665	586.7562	**-0.002%**	2188.11	**586.6984**	**-0.012%**	2084.30	586.7665	0.000%	2056.22
10000	100	554.7646	**554.5849**	**-0.032%**	2507.24	554.7152	**-0.009%**	2348.57	554.6667	**-0.018%**	2316.84
Average:				**-0.062%**	790.82		**-0.051%**	751.97		**-0.028%**	740.21

*Best known results from published papers.
+Negative values are the best improved best known results.
† Percent above the best known results. Negative percentages in boldface.
‡ Run times in minutes.

4 Computational Experiments

In this section we use a recent CLUST algorithm (short for Cluster; Drezner and Brimberg, 2026b, see also the Appendix) which is the best heuristic algorithm so far for solving large p-median instances. CLUST uses high quality starting solutions obtained by a constructive heuristic SPL (Drezner and Brimberg, 2026a) with random components.

We tested the POST procedure on the 24 large uniform instances proposed in Drezner et al. (2023), and the 24 non-uniformly generated instances created in Drezner and Brimberg (2026a). We observed that new improved best known solutions are found for all 48 instances.

Table 2: Results for 24 Uniform instances

n	p	Best*	(1)	(2)	SPL$^{\$}$	CLUST†	POST‡	(1)	(2)	SPL$^{\$}$	CLUST†	POST‡
1000	50	71.8014	71.8715	71.8283	71.8263	71.8130	71.8014	0.098%	0.037%	0.035%	0.016%	0%
1000	60	63.9736	64.0382	64.0545	64.0664	63.9879	63.9751	0.101%	0.126%	0.145%	0.022%	0.002%
1000	70	57.5821	57.7542	57.7450	57.6551	57.6391	57.5821	0.299%	0.283%	0.127%	0.099%	0%
1000	80	52.7813	52.8959	53.0803	52.9381	52.8462	52.7813	0.217%	0.566%	0.297%	0.123%	0%
1000	90	48.9341	49.2098	49.3712	49.1478	49.0128	48.9435	0.563%	0.893%	0.437%	0.161%	0.019%
1000	100	45.6437	45.9906	45.9887	45.7834	45.7952	45.6541	0.760%	0.756%	0.306%	0.332%	0.023%
2000	50	150.7339	150.8145	150.7649	150.8344	150.7560	150.7339	0.053%	0.021%	0.067%	0.015%	0%
2000	60	135.7747	135.9403	136.0076	135.9773	135.8410	135.7747	0.122%	0.172%	0.149%	0.049%	0%
2000	70	123.8204	124.0276	124.1456	123.9656	123.8238	123.8204	0.167%	0.263%	0.117%	0.003%	0%
2000	80	114.9689	115.2226	115.2252	115.1419	115.0461	114.9689	0.221%	0.223%	0.150%	0.067%	0%
2000	90	107.5079	107.7817	108.0196	107.7556	107.6620	107.5079	0.255%	0.476%	0.230%	0.143%	0%
2000	100	101.1014	101.4669	101.6735	101.4110	101.2800	101.1083	0.362%	0.566%	0.306%	0.177%	0.007%
5000	50	391.5846	391.7090	391.6951	391.7029	391.6220	391.5846	0.032%	0.028%	0.030%	0.010%	0%
5000	60	355.3818	355.5684	355.6066	355.5968	355.3980	355.3818	0.053%	0.063%	0.060%	0.005%	0%
5000	70	327.3036	327.4698	327.4828	327.5939	327.4487	327.3036	0.051%	0.055%	0.089%	0.044%	0%
5000	80	304.3666	304.7394	304.6706	304.5045	304.4473	304.4081	0.122%	0.100%	0.045%	0.027%	0.014%
5000	90	284.7887	285.1334	285.6168	285.3026	285.0280	284.8145	0.121%	0.291%	0.180%	0.084%	0.009%
5000	100	268.7861	269.3923	269.6615	269.5282	269.1664	268.7861	0.226%	0.326%	0.276%	0.141%	0%
10000	50	798.6774	798.7444	798.6799	798.7432	798.6997	798.6849	0.008%	0.000%	0.008%	0.003%	0.001%
10000	60	725.1194	725.3001	725.1684	725.3322	725.2186	725.1450	0.025%	0.007%	0.029%	0.014%	0.004%
10000	70	669.8753	669.9331	669.9188	670.1747	669.9168	669.8753	0.009%	0.006%	0.045%	0.006%	0%
10000	80	624.6056	625.1549	625.2278	625.2775	624.7508	624.6261	0.088%	0.100%	0.108%	0.023%	0.003%
10000	90	586.6984	586.9920	587.3671	587.4666	586.7665	586.7562	0.050%	0.114%	0.131%	0.012%	0.010%
10000	100	554.5849	554.8952	555.1873	554.9741	554.7646	554.5849	0.056%	0.109%	0.070%	0.032%	0%
Average:								0.169%	0.233%	0.143%	0.067%	0.004%
One tail p-value compared with POST column:								6.7×10^{-5}	6.8×10^{-5}	1.0×10^{-6}	2.2×10^{-4}	

* The best known solution in all papers including the best results found in this paper.
(1) Drezner et al. (2024) best of 42 parameter sets; total run time 145 days.
(2) Brimberg and Drezner (2025) best of 20 parameter sets; total run time 64 days.
$^{\$}$ best SPL (Drezner and Brimberg, 2026a) with one parameter set; total run time 13 days.
† CLUST (Drezner and Brimberg, 2026b) results with one parameter set; total run time 13 days.
‡ POST results from Table 1 for $I = 1,000$, $L = 70$ (POST1); total run time 13 days.

4.1 Testing Uniform Instances

All 24 uniform solutions were improved in Drezner et al. (2024), nine of the 24 were further improved in Brimberg and Drezner (2025), and 21 more instances were further improved by CLUST (Drezner and Brimberg, 2026b). CLUST (Drezner and Brimberg, 2026b) improved 20 out of the 24 solutions of the non-uniform instances found by the SPL algorithm (Drezner and Brimberg, 2026a).

In the following experiments, CLUST was repeated from $N = 80,000$ starting solutions as was tested in Drezner and Brimberg (2026b). We tested I=1,000; 10,000, and 80,000, and $L = \frac{N}{I}$. However, I=1,000 required a slightly longer run time. Therefore, to obtain a comparable run time, we applied $L = 70$ for $I = 1,000$.

The results by POST are reported in Table 1. The number after "POST" refers to the value of I in thousands. The results for $I = 1,000, L = 70$ (POST1) are best with the average deviation

from the last known best solution of -0.062%, compared with -0.051% for $I = 10,000, L = 8$ (POST10), and -0.028% for $I = N = 80,000$ and $L = 1$ (POST80). Comparing each pair by the paired t-test gave the following p-values: 0.041 between $I = 1,000$ and $I = 10,000$; 0.0005 between $I = 1,000$ and $N = 80.000$; and 0.012 between $I = 10,000$ and $I = 80.000$. All differences are statistically significant at the commonly used $p = 0.05$ level. The improvement between $I = 1,000$ and $I = 80,000$ is the most significant. Total run time for $I = 1,000$ was less than 7% higher, at about 13.2 days.

The results from all papers that tested the uniform instances are summarized in Table 2. Since the results in Drezner et al. (2023) where inferior to all the results in published papers, we did not include them in the table. Since POST is repeated for 3 sets of parameters, we selected, for a fair comparison, the best one $I = 1,000, L = 70$ (POST1), as a representative for POST so that the run times are comparable, and are not a combined 39 days if all three versions are considered. The results in the POST column are significantly better than each of the other methods with p-values listed in the last row of the table.

In Table 1 the best values for each variant are boldfaced if they are the revised best known solution. POST1 found the best known solution in all papers, including the three versions of POST, in 14 of the 24 instances, POST10 found it in 7 instances, and POST80 found it in only 3 cases. The other papers combined obtained the revised best known solution zero times. Note that if both POST1 and POST10 results are combined (it will require about 26 days to complete), the new best known solution is found in 21 of the 24 instances. The results of the three POST parameters combined found a new best known solution for all 24 instances.

Run times for POST1 are approximately $0.0006904n^{1.223}p^{0.8356}$ minutes, with the regression p-value of $6.9 \cdot 10^{-23}$. These numbers mean that when n increased 10 times (for example, from 1,000 to 10,000) run times increased about 17 times. However, when p is doubled, it increases less than twice at a ratio of about 1.8.

Table 3: Results by POST for 24 non-uniform instances

		Best	I =1,000, $L = 70$ (POST1)			I =10,000, $L = 8$ (POST10)			I =80,000, $L = 1$ (POST80)		
n	p	Known*	Value+	†	‡	Value+	†	‡	Value+	†	‡
1000	50	63.3601	63.3058	**-0.086%**	82.37	**63.3041**	**-0.088%**	78.44	63.3412	**-0.030%**	76.93
1000	60	56.5553	**56.5234**	**-0.056%**	89.40	56.5370	**-0.032%**	86.95	56.5616	0.011%	85.36
1000	70	51.2058	**51.1354**	**-0.137%**	102.38	51.1409	**-0.127%**	97.99	51.2058	0.000%	96.81
1000	80	46.6700	46.5965	**-0.157%**	106.20	**46.5920**	**-0.167%**	105.95	46.5985	**-0.153%**	101.48
1000	90	43.1613	**43.1038**	**-0.133%**	116.81	43.1434	**-0.041%**	114.79	43.1610	**-0.001%**	109.46
1000	100	40.2536	40.1609	**-0.230%**	128.27	**40.1414**	**-0.279%**	124.76	40.2069	**-0.116%**	118.81
2000	50	132.2109	132.2102	**-0.001%**	196.97	**132.2074**	**-0.003%**	181.80	132.2109	0.000%	178.28
2000	60	119.3157	**119.3135**	**-0.002%**	207.39	119.3416	0.022%	205.10	119.3157	0.000%	200.87
2000	70	108.8951	108.8599	**-0.032%**	242.30	**108.8420**	**-0.049%**	236.68	108.8932	**-0.002%**	232.17
2000	80	100.6412	100.5925	**-0.048%**	246.73	**100.4824**	**-0.158%**	240.12	100.4830	**-0.157%**	234.94
2000	90	93.4773	**93.3405**	**-0.146%**	280.92	93.4242	**-0.057%**	272.26	93.4471	**-0.032%**	268.42
2000	100	87.5557	**87.4042**	**-0.173%**	305.94	87.4368	**-0.136%**	289.70	87.4587	**-0.111%**	284.11
5000	50	344.0196	**343.9968**	**-0.007%**	490.21	344.0096	**-0.003%**	451.84	344.0284	0.003%	439.31
5000	60	312.6244	**312.6025**	**-0.007%**	637.31	312.6107	**-0.004%**	618.21	312.6104	**-0.004%**	604.71
5000	70	287.1591	**286.9828**	**-0.061%**	747.20	287.0791	**-0.028%**	741.68	286.9840	**-0.061%**	729.40
5000	80	266.5877	**266.4461**	**-0.053%**	898.52	266.4726	**-0.043%**	844.66	266.4533	**-0.050%**	825.97
5000	90	249.8066	249.6422	**-0.066%**	1003.85	**249.5525**	**-0.102%**	950.06	249.5732	**-0.093%**	934.33
5000	100	235.2333	**235.0865**	**-0.062%**	1077.92	235.1862	**-0.020%**	1027.02	235.2333	0.000%	1014.26
10000	50	697.0866	697.0745	**-0.002%**	1074.84	**697.0701**	**-0.002%**	988.27	697.0753	**-0.002%**	975.70
10000	60	635.0829	635.0622	**-0.003%**	1177.34	**635.0125**	**-0.011%**	1146.35	635.0126	**-0.011%**	1130.25
10000	70	585.0021	584.9781	**-0.004%**	1366.35	**584.9599**	**-0.007%**	1335.82	584.9806	**-0.004%**	1322.49
10000	80	544.2333	**543.9566**	**-0.051%**	1600.14	544.0633	**-0.031%**	1571.26	544.0368	**-0.036%**	1477.58
10000	90	511.4076	511.3568	**-0.010%**	1902.86	**511.3355**	**-0.014%**	1820.10	511.4076	0.000%	1734.81
10000	100	482.6944	**482.2426**	**-0.094%**	2131.30	482.5196	**-0.036%**	2000.87	482.5624	**-0.027%**	1906.53
Average:				**-0.068%**	675.56		**-0.059%**	647.11		**-0.037%**	628.46

*Best known results from published papers.
+Negative values are the best improved best known results.
† Percent above the best known results. Negative percentages in boldface.
‡ Run times in minutes.

4.2 Testing Non-Uniform Instances

We repeated the experiments on the 24 non-uniform instances to assess the performance differences between these two types of problems. These instances were tested in two papers (Drezner and Brimberg, 2026a,b).

The results by POST are reported in Table 3. The results for POST1 are best with the average deviation from the last known best solution of -0.068%, compared with -0.059% for POST10, and -0.037% for POST80.

The results from all papers that tested the non-uniform instances, and the three variants of POST (termed POST80, POST10 and POST1) are summarized in Table 4.

Table 4: Results for 24 Non-uniform instances

n	p	Best*	SPL	CLUST	POST80	POST10	POST1	SPL	CLUST	POST80	POST10	POST1
1000	50	63.3041	63.4197	63.3601	63.3412	63.3041	63.3058	0.183%	0.088%	0.059%	0%	0.003%
1000	60	56.5234	56.5553	56.5626	56.5616	56.5370	56.5234	0.056%	0.069%	0.068%	0.024%	0%
1000	70	51.1354	51.2116	51.2058	51.2058	51.1409	51.1354	0.149%	0.138%	0.138%	0.011%	0%
1000	80	46.5920	46.6700	46.6862	46.5985	46.5920	46.5965	0.167%	0.202%	0.014%	0%	0.010%
1000	90	43.1038	43.2668	43.1613	43.1610	43.1434	43.1038	0.378%	0.133%	0.133%	0.092%	0%
1000	100	40.1414	40.2536	40.2620	40.2069	40.1414	40.1609	0.280%	0.300%	0.163%	0%	0.049%
2000	50	132.2074	132.2899	132.2109	132.2109	132.2074	132.2102	0.062%	0.003%	0.003%	0%	0.002%
2000	60	119.3135	119.4298	119.3157	119.3157	119.3416	119.3135	0.097%	0.002%	0.002%	0.024%	0%
2000	70	108.8420	109.0226	108.8951	108.8932	108.8420	108.8599	0.166%	0.049%	0.047%	0%	0.016%
2000	80	100.4824	100.6802	100.6412	100.4830	100.4824	100.5925	0.197%	0.158%	0.001%	0%	0.110%
2000	90	93.3405	93.5995	93.4773	93.4471	93.4242	93.3405	0.277%	0.147%	0.114%	0.090%	0%
2000	100	87.4042	87.6522	87.5557	87.4587	87.4368	87.4042	0.284%	0.173%	0.062%	0.037%	0%
5000	50	343.9968	344.0196	344.0305	344.0284	344.0096	343.9968	0.007%	0.010%	0.009%	0.004%	0%
5000	60	312.6025	313.0611	312.6244	312.6104	312.6107	312.6025	0.147%	0.007%	0.003%	0.003%	0%
5000	70	286.9828	287.2732	287.1591	286.9840	287.0791	286.9828	0.101%	0.061%	0.000%	0.034%	0%
5000	80	266.4461	266.8620	266.5877	266.4533	266.4726	266.4461	0.156%	0.053%	0.003%	0.010%	0%
5000	90	249.5525	250.0819	249.8066	249.5732	249.5525	249.6422	0.212%	0.102%	0.008%	0%	0.036%
5000	100	235.0865	235.5951	235.2333	235.2333	235.1862	235.0865	0.216%	0.062%	0.062%	0.042%	0%
10000	50	697.0701	697.1170	697.0866	697.0753	697.0701	697.0745	0.007%	0.002%	0.001%	0%	0.001%
10000	60	635.0125	635.3547	635.0829	635.0126	635.0125	635.0622	0.054%	0.011%	0.000%	0%	0.008%
10000	70	584.9599	585.3349	585.0021	584.9806	584.9599	584.9781	0.064%	0.007%	0.004%	0%	0.003%
10000	80	543.9566	544.5050	544.2333	544.0368	544.0633	543.9566	0.101%	0.051%	0.015%	0.020%	0%
10000	90	511.3355	511.9304	511.4076	511.4076	511.3355	511.3568	0.116%	0.014%	0.014%	0%	0.004%
10000	100	482.2426	482.8746	482.6944	482.5624	482.5196	482.2426	0.131%	0.094%	0.066%	0.057%	0%
Average:								0.150%	0.081%	0.041%	0.019%	0.010%

*The best known solution in all papers including the best results found in this paper.

The results of POST1 are better than each of the other methods. The best known results in all experiments were found 11 times out of 24 by POST10, and 13 times by POST1. SPL, CLUST, and POST80 found it zero times. Note that if both $I = 1,000$ and $I = 10,000$ results are combined (it will require approximately 21 days to complete), new best known solutions are found in all 24 instances.

Run times for POST1 are approximately $0.0007797n^{1.177}p^{0.8622}$ minutes, with the regression p-value of $2.1 \cdot 10^{-23}$. These numbers mean that when n increases 10 times (for example, from 1,000 to 10,000) run time increases about 15 times. However, when p is doubled, it increases less than twice at a ratio of about 1.8. These numbers are similar to the uniform instances.

Table 5: Summary of Results

	SPL	CLUST	POST80	POST10	POST1
Average of 24 uniform instances:	0.143%	0.067%	0.037%	0.014%	0.004%
One tail p-value compared with POST1 column:	$1.0 \cdot 10^{-6}$	$2.2 \cdot 10^{-4}$	$4.8 \cdot 10^{-4}$	0.041	
Average of 24 non-uniform instances:	0.150%	0.081%	0.041%	0.019%	0.010%
One tail p-value compared with POST1 column:	$4.8 \cdot 10^{-8}$	$2.4 \cdot 10^{-5}$	0.0064	0.163	
Average of all 48 instances:	0.147%	0.074%	0.039%	0.016%	0.007%
One tail p-value compared with POST1 column:	$2.9 \cdot 10^{-13}$	$2.8 \cdot 10^{-8}$	$2.3 \cdot 10^{-5}$	0.035	

4.3 Discussion of the Results

A summary of all the results are reported in Table 5. The results for 24 uniform instances, 24 non-uniform instances, and the combined results of all 48 instances show a similar pattern. The percentages of the average results above the new updated best found solution decline from left to right. The least significant difference is the decline from POST10 to POST1. In the non-uniform instances it is the only non-significant difference at the $p = 0.05$ level.

The results confirm the "Less is More" principle (Costa et al., 2017; Mladenović et al., 2022; Silva et al., 2018). In POST80, we select the five best found solutions from all $N = 80,000$ runs. In POST with $I << N$, the best found solutions from I starting solutions are found. These solutions are not as good as the solutions selected after N starting solutions, but eventually they find much better solutions by repeating the process L times, while the execution time remains about the same.

5 Conclusions and Suggestions for Future Research

This paper presents a new meta-heuristic algorithm (called POST) for improving the performance of any general multi-start heuristic algorithm. Improvement algorithms typically begin with a random feasible solution and move through a series of feasible improving solutions until a local optimum is reached. The multi-start heuristic repeats this process from a given number (N) random starting solutions. See Section 2 for details.

The POST procedure was tested on the planar p-median problem on 48 large instances with up to 10,000 demand points and 100 facilities. Experimental tests were conducted for various parameter settings. A new improvement algorithm CLUST (Drezner and Brimberg, 2026b, also

detailed in the appendix) which performed best among improvement algorithms was applied. To maintain reasonable execution times, the number of repetitions of CLUST in the initial multi-start phase was set to $N = 80{,}000$. POST improved the best known solution reported in the literature for all 48 tested instances.

Directions for future research include further investigations of POST with different parameter settings. For example, it may be advantageous to increase the size K (detailed in Section 2) of the elite sets. This, of course, comes at a cost of increased computing time. Different ways of choosing the elite subset of solutions should also be investigated. It may be worthwhile to choose the K solutions based on more than one criterion, for example not only the quality of the solutions but also the distances between them. More dispersion in the elite solutions may be advantageous. Applying POST to other models that are solved heuristically by multi-start procedures may result in better solutions found for such models.

Disclosure of Interest: There are no interests to declare.

Funding: Jack Brimberg was supported by a Natural Sciences and Engineering Research Council of Canada Discovery Grant (RGPIN-2020-04846).

Data availability: Data can be easily generated, and is also available in existing papers.

References

Aloise, D. (2009). *Exact algorithms for minimum sum-of-squares clustering*. PhD thesis, Ecole Polytechnique, Montreal, Canada. ISBN:978-0-494-53792-3.

Aloise, D., Hansen, P., and Liberti, L. (2012). An improved column generation algorithm for minimum sum-of-squares clustering. *Mathematical Programming*, 131:195–220.

Bagirov, A. M., Ordin, B., Ozturk, G., and Xavier, A. E. (2015). An incremental clustering algorithm based on hyperbolic smoothing. *Computational Optimization and Applications*, 61:219–241.

Ballantine, J. P. and Jerbert, A. R. (1952). Distance from a line, or plane, to a point. *The American Mathematical Monthly*, 59:242–243.

Brimberg, J. and Drezner, Z. (2013). A new heuristic for solving the p-median problem in the plane. *Computers and Operations Research*, 40:427–437.

Brimberg, J. and Drezner, Z. (2021). Improved starting solutions for the planar p-median problem. *Yugoslav Journal of Operations Research*, 31:45–64.

Brimberg, J. and Drezner, Z. (2025). Less is more: Adjusting convergence of Cooper's algorithm in the continuous location-allocation problem. *Journal of the Operational Research Society*, 76:1599–1611.

Brimberg, J., Hansen, P., and Mladenović, N. (2010). Attraction probabilities in variable neighborhood search. *4OR - Quarterly Journal of Operations Research*, 8:181–194.

Brimberg, J., Hansen, P., Mladenović, N., and Salhi, S. (2008). A survey of solution methods for the continuous location allocation problem. *International Journal of Operations Research*, 5:1–12.

Brimberg, J., Hansen, P., Mladenović, N., and Taillard, E. (2000). Improvements and comparison of heuristics for solving the uncapacitated multisource Weber problem. *Operations Research*, 48:444–460.

Brimberg, J. and Hodgson, M. J. (2011). Heuristics for location models. In Eiselt, H. A. and Marianov, V., editors, *Foundations of Location Analysis: International Series in Operations Research and Management Science, Vol. 155*, pages 335–355. Springer, New York, NY.

Brimberg, J., Janićijević, S., Mladenović, N., and Urošević, D. (2017). Solving the clique partitioning problem as a maximally diverse grouping problem. *Optimization Letters*, 11:1123–1135.

Brimberg, J. and Salhi, S. (2019). A general framework for local search applied to the continuous p-median problem. In *Contributions to Location Analysis - In Honor of Zvi Drezner's 75th Birthday*, pages 89–108. Springer.

Brusco, M. J. and Köhn, H.-F. (2009). Clustering qualitative data based on binary equivalence relations: neighborhood search heuristics for the clique partitioning problem. *Psychometrika*, 74:685.

Chen, R. (1983). Solution of minisum and minimax location-allocation problems with Euclidean distances. *Naval Research Logistics Quarterly*, 30:449–459.

Church, R., Drezner, Z., Plastria, F., and Tamir, A. (2025). Reviewing extensions and solution methods of the planar Weber single facility location problem. *Computers and Operations Research*, 173:106825. DOI:10.1016/j.cor.2024.106825.

Cooper, L. (1963). Location-allocation problems. *Operations Research*, 11:331–343.

Cooper, L. (1964). Heuristic methods for location-allocation problems. *SIAM Review*, 6:37–53.

Costa, L. R., Aloise, D., and Mladenović, N. (2017). Less is more: basic variable neighborhood search heuristic for balanced minimum sum-of-squares clustering. *Information Sciences*, 415:247–253.

Dantrakul, S., Likasiri, C., and Pongvuthithum, R. (2014). Applied p-median and p-center algorithms for facility location problems. *Expert Systems with Applications*, 41:3596–3604.

Daskin, M. S. (1995). *Network and Discrete Location: Models, Algorithms, and Applications*. John Wiley and Sons, New York.

Daskin, M. S. and Maass, K. L. (2015). The p-median problem. In Laporte, G., Nickel, S., and da Gama, F. S., editors, *Location Science*, pages 21–45. Springer.

Drezner, Z. and Brimberg, J. (2026a). Generating appropriate starting configurations for continuous multi-facility location models. *IMA Journal of Management Mathematics*. DOI: 10.1093/imaman/dpag038.

Drezner, Z. and Brimberg, J. (2026b). A new improvement algorithm for solving location-allocation problems: the case of the planar p-median problem. In review. Also to be presented at the Annual INFORMS Conference, November 2026, San Francisco.

Drezner, Z., Brimberg, J., Mladenović, N., and Salhi, S. (2015a). Solving the planar p-median problem by variable neighborhood and concentric searches. *Journal of Global Optimization*, 63:501–514.

Drezner, Z., Brimberg, J., Salhi, S., and Mladenović, N. (2015b). New heuristic algorithms for solving the planar *p*-median problem. *Computers and Operations Research*, 62:296–304.

Drezner, Z., Brimberg, J., Salhi, S., and Mladenović, N. (2016). New local searches for solving the multi-source Weber problem. *Annals of Operations Research*, 246:181–203.

Drezner, Z., Brimberg, J., and Schöbel, A. (2023). A trajectory based heuristic for the planar *p*-median problem. *Computers and Operations Research*, 158:106296. DOI:10.1016/j.cor.2023.106296.

Drezner, Z., Brimberg, J., and Schöbel, A. (2024). Dispersed starting solutions in facility location: The case of the planar p-median problem. *Computers and Operations Research*, 169:106726.

Drezner, Z. and Salhi, S. (2002). Using hybrid metaheuristics for the one-way and two-way network design problem. *Naval Research Logistics*, 49:449–463.

Drezner, Z. and Salhi, S. (2017). Incorporating neighborhood reduction for the solution of the planar *p*-median problem. *Annals of Operations Research*, 258:639–654.

Eilon, S., Watson-Gandy, C. D. T., and Christofides, N. (1971). *Distribution Management*. Hafner, New York.

Glover, F. and Laguna, M. (1997). *Tabu Search*. Kluwer Academic Publishers, Boston.

Glover, F., Laguna, M., and Martí, R. (2000). Fundamentals of scatter search and path relinking. *Control and cybernetics*, 29:653–684.

Goldberg, D. E. (2006). *Genetic algorithms*. Pearson Education, Delhi, India.

Gribel, D. and Vidal, T. (2019). HG-means: A scalable hybrid genetic algorithm for minimum sum-of-squares clustering. *Pattern Recognition*, 88:569–583.

Grötschel, M. and Wakabayashi, Y. (1989). A cutting plane algorithm for a clustering problem. *Mathematical Programming*, 45:59–96.

György, A. and Kocsis, L. (2011). Efficient multi-start strategies for local search algorithms. *Journal of Artificial Intelligence Research*, 41:407–444.

Hansen, P., Brimberg, J., Urošević, D., and Mladenović, N. (2009). Solving large p-median clustering problems by primal–dual variable neighborhood search. *Data Mining and Knowledge Discovery*, 19:351–375.

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

Hansen, P. and Mladenović, N. (2001). Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, 130:449 – 467.

Hansen, P., Mladenović, N., and Taillard, É. (1998). Heuristic solution of the multisource Weber problem as a p-median problem. *Operations Research Letters*, 22:55–62.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI.

Kalczynski, P., Brimberg, J., and Drezner, Z. (2022). Less is more: Simple algorithms for the minimum sum of squares clustering problem. *IMA Journal of Management Mathematics*, 33:531–548.

Kalczynski, P., Goldstein, Z., and Drezner, Z. (2023). An efficient heuristic for the k-partitioning problem. *Operations Research Forum*, 4:70. DOI: doi.org/10.1007/s43069-023-00249-w.

Katoch, S., Chauhan, S. S., and Kumar, V. (2021). A review on genetic algorithm: past, present, and future. *Multimedia tools and applications*, 80:8091–8126.

Kirkpatrick, S., Gelat, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220:671–680.

Krau, S. (1997). *Extensions du problème de Weber*. PhD thesis, École Polytechnique de Montréal.

Kuenne, R. E. and Soland, R. M. (1972). Exact and approximate solutions to the multisource Weber problem. *Mathematical Programming*, 3:193–209.

Love, R. F., Morris, J. G., and Wesolowsky, G. O. (1988). *Facilities Location: Models and Methods*. North Holland, New York, NY.

Martí, R., Lozano, J. A., Mendiburu, A., and Hernando, L. (2025). Multi-start methods. In *Handbook of heuristics*, pages 211–230. Springer.

Martí, R., Moreno-Vega, J. M., and Duarte, A. (2010). Advanced multi-start methods. In *Handbook of metaheuristics*, pages 265–281. Springer.

Martí, R., Resende, M. G., and Ribeiro, C. C. (2013). Multi-start methods for combinatorial optimization. *European Journal of Operational Research*, 226:1–8.

Megiddo, N. and Supowit, K. J. (1984). On the complexity of some common geometric location problems. *SIAM Journal on Computing*, 13:182–196.

Mladenović, N., Drezner, Z., Brimberg, J., and Urošević, D. (2022). Less is more approach in heuristic optimization. In Salhi, S. and Boylan, J. E., editors, *The Palgrave Handbook of Operations Research*, pages 469–499. Palgrave, London.

Moran, M. D. (2003). Arguments for rejecting the sequential Bonferroni in ecological studies. *Oikos*, 100:403–405.

Moreno-Perez, J., Moreno-Vega, J., and Mladenović, N. (1994). Tabu search and simulated annealing in p-median problem. In *Proceedings of the Canadian Operational Research Society Conference, Montreal.*

Murtagh, B. A. and Niwattisyawong, S. R. (1982). An efficient method for the multi-depot location-allocation problem. *Journal of the Operational Research Society*, 33:629–634.

Ordin, B. and Bagirov, A. M. (2015). A heuristic algorithm for solving the minimum sum-of-squares clustering problems. *Journal of Global Optimization*, 61:341–361.

Pereira, T., Aloise, D., Brimberg, J., and Mladenović, N. (2018). Review of basic local searches for solving the minimum sum-of-squares clustering problem. In *Open Problems in Optimization and Data Analysis*, pages 249–270. Springer.

Silva, K. G., Aloise, D., Xavier-de Souza, S., and Mladenovic, N. (2018). Less is more: simplified Nelder-Mead method for large unconstrained optimization. *Yugoslav Journal of Operations Research*, 28:153–169.

Weber, A. (1909). *Über den Standort der Industrien, 1. Teil: Reine Theorie des Standortes. English Translation: on the Location of Industries*. University of Chicago Press, Chicago, IL. Translation published in 1929.

Zhou, Q., Benlic, U., and Wu, Q. (2020). A memetic algorithm based on reformulation local search for minimum sum-of-squares clustering in networks. *Information Sciences*, 541:271–296.

Zhou, Q., Hao, J.-K., and Wu, Q. (2021). Responsive threshold search based memetic algorithm for balanced minimum sum-of-squares clustering. *Information Sciences*, 569:184–204.

Appendix: The CLUST Algorithm (Drezner and Brimberg, 2026b)

There are two values caclulated for each subset $1 \leq j \leq p$:

Δ_j^+ The increase in the value of the objective function by removing the facility serving subset j.

Δ_j^- The decrease in the value of the objective function by replacing the facility serving subset j by two facilities found by solving heuristicaly the 2-median problem in the subset. The heuristic H2M algorithm (Drezner and Brimberg, 2026a) is applied for getting a "good" solution to the 2-median problem.

The Algorithm

1. Select a starting solution X_j for $j = 1, \ldots, p$ by the SPL algorithm (Drezner and Brimberg, 2026a), followed by ALTR (ALT followed by RATIO, Brimberg and Drezner, 2021).
2. Store the values of X_j for $j = 1, \ldots, p$, and the value of the objective function F.
3. Calculate Δ_j^+ and Δ_j^- for $1 \leq j \leq p$.
4. Create two lists of K (we applied $K = 5$) items each. The K smallest Δ_j^+ (no facility location needed), and the K largest Δ_j^- with the locations of the added two facilities.
5. Repeat the following for up to K^2 pairs between the two lists. Index j_1 is used for the Δ_j^+ list, and j_2 for the Δ_j^- list. The order of the selected pairs is determined by the order of $\Delta_{j_1}^+ - \Delta_{j_2}^-$ (which is the change in the value of the objective function) from the smallest (possibly negative) to the largest.
   (a) If $j_1 = j_2$ skip this combination.
   (b) Remove X_{j_1} and X_{j_2} from the list of facilities and add the two new facilities of j_2 to the list.
   (c) Apply ALTR on the modified list of facilities.
   (d) If the objective function is better than F, replace the "best known solution" with the new set of locations of the p facilities, update F, and go to Step 3.
   (e) Otherwise, select the next pair of j_1, j_2 (up to the 10 best pairs to save computer time) and go to step 5a.
6. If no improvement is found for the first 10 pairs in Step 5, stop.