A NEW LOCAL SEARCH FOR THE P-CENTER PROBLEM BASED ON THE CRITICAL VERTEX CONCEPT

DANIELE FERONE, PAOLA FESTA, ANTONIO NAPOLETANO, AND MAURICIO G.C. RESENDE

Abstract. We propose a new smart local search for the p-center problem, based on the critical vertex concept, and embed it in a GRASP framework. Experimental results attest the robustness of the proposed search procedure and confirm that for benchmark instances it converges to optimal or near/optimal solutions faster than the best known state-of-the-art local search.

1. Introduction

The p-center problem is one of the best-known discrete location problems first introduced in the literature in 1964 by Hakimi [13]. It consists of locating p facilities and assigning clients to them in order to minimize the maximum distance between a client and the facility to which the client is assigned (i.e., the closest facility). Useless to say that this problem arises in many different real-world contexts, whenever one designs a system for public facilities, such as schools or emergency services.

Formally, we are given a complete undirected edge-weighted bipartite graph $G = (V \cup U, E, c)$, where

- $V = \{1, 2, \dots, n\}$ is a set of n potential locations for facilities;
- $U = \{1, 2, \dots, m\}$ is a set of m clients or demand points;
- $E = \{(i, j)|\ i \in V,\ j \in U\}$ is a set of $n \times m$ edges;
- $c : E \mapsto \mathbb{R}^+ \cup \{0\}$ is a function that assigns a nonnegative distance c_{ij} to each edge $(i, j) \in E$.

The p-center problem is to find a subset $P \subseteq V$ of size p such that its weight, defined as

$$\mathbb{C}(P) = \max_{i \in U} \min_{j \in P} c_{ij} \tag{1}$$

is minimized. The minimum value is called the *radius*. Although it is not a restrictive hypothesis, in this paper we consider the special case where $V = U$ is the vertex set of a complete graph $G = (V, E)$, each distance c_{ij} represents the length of a shortest path between vertices i and j ($c_{ii} = 0$), and hence the triangle inequality is satisfied.

Date: June 19, 2017.

Key words and phrases. Local search, GRASP, critical vertex concept, p-center problem.

Parts of this paper were written while the last author was employed at AT&T Labs Research. To appear in *Proceedings of 11th International Conference on Learning and Intelligent Optimization (LION 11)*, Lecture Notes in Computer Science, R. Battiti, D.E. Kvasov, and Y.D. Sergeyev (Eds.), Springer, 2017.

In 1979, Kariv and Hakimi [16] proved that the problem is *NP*-hard, even in the case where the input instance has a simple structure (e.g., a planar graph of maximum vertex degree 3). In 1970, Minieka [20] designed the first exact method for the p-center problem viewed as a series of set covering problems. His algorithm iteratively chooses a threshold r for the radius and checks whether all clients can be covered within distance r using no more than p facilities. If so, the threshold r is decreased; otherwise, it is increased. Inspired by Minieka's idea, in 1995 Daskin [3] proposed a recursive bisection algorithm that systematically reduces the gap between upper and lower bounds on the radius. More recently, in 2010 Salhi and Al-Khedhairi [26] proposed a faster exact approach based on Daskin's algorithm that obtains tighter upper and lower bounds by incorporating information from a three-level heuristic that uses a variable neighborhood strategy in the first two levels and at the third level a perturbation mechanism for diversification purposes.

Recently, several facility location problems similar to the p-center have been used to model scenarios arising in financial markets. The main steps to use such techniques are the following: first, to describe the considered financial market via a correlation matrix of stock prices; second, to model the matrix as a graph, stocks and correlation coefficients between them are represented by nodes and edges, respectively. With this idea, Goldengorin et al. [11] used the p-median problem to analyze stock markets. Another interesting area where these problems arise is the manufacturing system with the aim of lowering production costs [12].

Due to the computational complexity of the p-center problem, several approximation and heuristic algorithms have been proposed for solving it. By exploiting the relationship between the p-center problem and the dominating set problem [15, 18], nice approximation results were proved. With respect to inapproximability results, Hochbaum and Shmoys [15] proposed a 2-approximation algorithm for the problem with triangle inequality, showing that for any $\delta < 2$ the existence of a δ-approximation algorithm would imply that $P = NP$.

Although interesting in theory, approximation algorithms are often outperformed in practice by more straightforward heuristics with no particular performance guarantees. Local search is the main ingredient for most of the heuristic algorithms that have appeared in the literature. In conjunction with various techniques for escaping local optima, these heuristics provide solutions which exceed the theoretical upper bound of approximating the problem and derive from ideas used to solve the p-median problem, a similar *NP*-hard problem [17]. Given a set F of m potential facilities, a set U of n users (or customers), a distance function $d : U \times F \mapsto \mathbb{R}$, and a constant $p \leq m$, the p-median problem is to determine a subset of p facilities to open so as to minimize the sum of the distances from each user to its closest open facility. For the p-median problem, in 2004 Resende and Werneck [25] proposed a multistart heuristic that hybridizes GRASP with Path-Relinking as both, intensification and post-optimization phases. In 1997, Hansen and Mladenović [14] proposed three heuristics: `Greedy`, `Alternate`, and `Interchange` (vertex substitution). To select the first facility, `Greedy` solves a 1-center problem. The remaining $p - 1$ facilities are then iteratively added, one at a time, and at each iteration the location which most reduces the maximum cost is selected. In [5], Dyer and Frieze suggested a variant, where the first center is chosen at random. In the first iteration of `Alternate`, facilities are located at p vertices chosen in V, clients are assigned to the closest facility, and the 1-center problem is solved for each facility's set of clients.

During the subsequent iterations, the process is repeated with the new locations of the facilities until no more changes in assignments occur. As for the `Interchange` procedure, a certain pattern of p facilities is initially given. Then, facilities are moved iteratively, one by one, to vacant sites with the objective of reducing the total (or maximum) cost. This local search process stops when no movement of a single facility decreases the value of the objective function. A multistart version of `Interchange` was also proposed, where the process is repeated a given number of times and the best solution is kept. The combination of `Greedy` and `Interchange` has been most often used for solving the p-median problem. In 2003, Mladenović et al. [21] adapted it to the p-center problem and proposed a Tabu Search (TS) and a Variable Neighborhood Search (VNS), i.e., a heuristic that uses the history of the search in order to construct a new solution and a competitor that is not history sensitive, respectively. The TS is designed by extending `Interchange` to the chain-interchange move, while in the VNS, a perturbed solution is obtained from the incumbent by a k-interchange operation and `Interchange` is used to improve it. If a better solution than the incumbent is found, the search is recentered around it. In 2011, Davidović et al. [4] proposed a Bee Colony algorithm, a random search population-based technique, where an artificial system composed of a number of precisely defined agents, also called individuals or artificial bees.

To the best of our knowledge, most of the research effort devoted towards the development of metaheuristics for this problem has been put into the design of efficient local search procedures. The purpose of this article is propose a new local search and to highlight how its performances are better than best-known local search proposed in literature (Mladenović et al.'s [21] local search based on VNS strategy), both in terms of solutions quality and convergence speed.

The remainder of the paper is organized as follows. In Section 2, a `GRASP` construction procedure is described. In Section 3, we introduce the new concept of *critical vertex* with relative definitions and describe a new local search algorithm. Computational results presented in Section 4 empirically demonstrate that our local search is capable of obtaining better results than the best known local search, and they are validated by a statistical significance test. Concluding remarks are made in Section 5.

2. GRASP Construction Phase

GRASP is a randomized multistart iterative method proposed in Feo and Resende [6, 7] and having two phases: a greedy randomized construction phase and a local search phase. For a comprehensive study of GRASP strategies and their variants, the reader is referred to the survey papers by Festa and Resende [9, 10], as well as to their annotated bibliography [8].

Starting from a partial solution made of $1 \leq randElem \leq p$ facilities randomly selected from V, our GRASP construction procedure iteratively selects the remaining $p - randElem$ facilities in a greedy randomized fashion. The greedy function takes into account the contribution to the objective function achieved by selecting a particular candidate element. In more detail, given a partial solution P, $|P| < p$, for each $i \in V \setminus P$, we compute $w(i) = \mathbb{C}(P \cup \{i\})$. The pure greedy choice would consist in selecting the vertex with the smallest greedy function value. This procedure

```
$P \leftarrow \emptyset$ ;
$randElem := \lfloor \alpha \cdot p \rfloor$ ;
for $k = 1, \ldots, randElem$ do                                  // random component
    $f \leftarrow$ SelectRandom($V \setminus P$);
    $P \leftarrow P \cup \{f\}$ ;
end
while $|P| < p$ do
    $z_{\min} \leftarrow +\infty$ ;
    $z_{\max} \leftarrow -\infty$ ;
    for $i \in V \setminus P$ do
        if $z_{\min} > \mathbb{C}(P \cup \{i\})$ then
            $z_{\min} \leftarrow \mathbb{C}(P \cup \{i\})$ ;
        end
        if $z_{\max} < \mathbb{C}(P \cup \{i\})$ then
            $z_{\max} \leftarrow \mathbb{C}(P \cup \{i\})$ ;
        end
    end
    $\mu \leftarrow z_{\min} + \beta(z_{\max} - z_{\min})$ ;
    $RCL \leftarrow \{i \in V \setminus P \mid \mathbb{C}(P \cup \{i\}) \leq \mu\}$ ;
    $f \leftarrow$ SelectRandom($RCL$) ;
    $P \leftarrow P \cup \{f\}$;
end
return $P$;
```
Function greedy-randomized-build($G = \langle V, E, \mathbb{C} \rangle, p, \alpha, \beta$)

Figure 1. Pseudo-code of the greedy randomized construction.

instead computes the smallest and the largest greedy function values:

$$z_{\min} = \min_{i \in V \setminus P} w(i); \quad z_{\max} = \max_{i \in V \setminus P} w(i).$$

Then, denoting by $\mu = z_{\min} + \beta(z_{\max} - z_{\min})$ the cut-off value, where β is a parameter such that $\beta \in [0, 1]$, a *restricted candidate list* (RCL) is made up of all vertices whose greedy value is less than or equal to μ. The new facility to be added to P is finally randomly selected from the RCL.

The pseudo-code is shown in Figure 1, where $\alpha \in [0, 1]$.

3. Plateau Surfer: a New Local Search Based on the Critical Vertex Concept

Given a feasible solution P, the `Interchange` local search proposed by Hansen and Mladenović [14] consists in swapping a facility $f \in P$ with a facility $\overline{f} \notin P$ which results in a decrease of the current cost function. Especially in the case of instances with many vertices, we have noticed that usually a single swap does not strictly improve the current solution, because there are several facilities whose distance is equal to the radius of the solution. In other words, the objective function is characterized by large plateaus and the `Interchange` local search cannot escape from such regions. To face this type of difficulties, inspired by Variable Formulation

Search [22, 23], we have decided to use a refined way for comparing between valid solutions by introducing the concept of *critical vertex*. Given a solution $P \subseteq V$, let $\delta_P : V \mapsto \mathbb{R}^+ \cup \{0\}$ be a function that assigns to each vertex $i \in V$ the distance between i and its closest facility according to solution P. Clearly, the cost of a solution P can be equivalently written as $\mathbb{C}(P) = \max\{\, \delta_P(i) \colon i \in V \,\}$. We also give the following definition:

Definition 1 (Critical vertex)**.** *Let $P \subseteq V$ be a solution whose cost is $\mathbb{C}(P)$. For each vertex $i \in V$, i is said to be a* critical vertex *for P, if and only if $\delta_P(i) = \mathbb{C}(P)$.*

In the following, we will denote with $\max_{\delta_P} = |\{\, i \in V \colon \delta_P(i) = \mathbb{C}(P) \,\}|$ the number of vertices whose distance from their closest facility results in the objective function value corresponding to solution P. We define also the comparison operator $<_{cv}$, and we will say that $P <_{cv} P'$ if and only if $\max_{\delta_P} < \max_{\delta'_P}$.

Then, a solution P has $p \cdot (n-p)$ neighbor solutions $\bar{P} = P \setminus \{i\} \cup \{j\}$, one for each $i \in P$ and $j \in V \setminus P$. A neighbor solution $\bar{P}$ of P is considered improving if either $\mathbb{C}(\bar{P}) < \mathbb{C}(P)$ or $\mathbb{C}(\bar{P}) = \mathbb{C}(P)$ and $\max_{\delta_{\bar{P}}} < \max_{\delta_P}$. A move is performed to the best improving neighbor.

The central point on which we base our local search is that the new solution $\bar{P}$ hasn't to be strictly better than P, according to the cost function, but the algorithm switch from P to $\bar{P}$ even when the cost is the same, but the number of *critical vertices* in $\bar{P}$ ($\max_{\delta_{\bar{P}}}$) is less than that in P ($\max_{\delta_P}$).

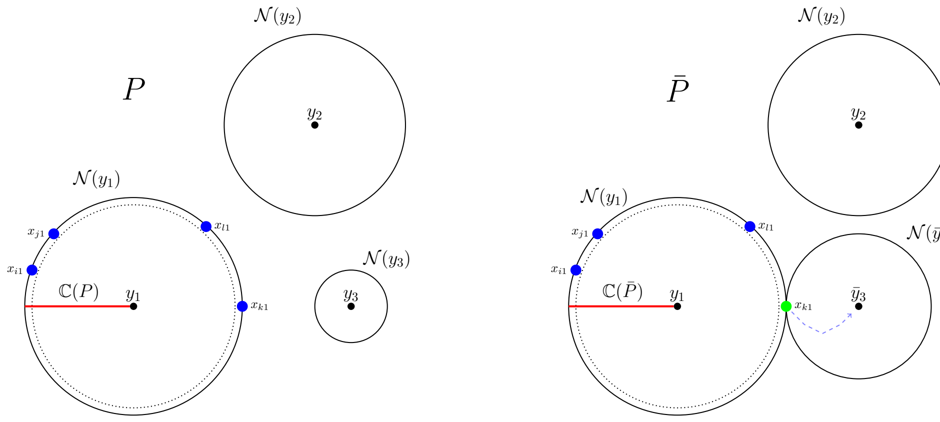


FIGURE 2. An example of how the local search works. In this case, the algorithm switches from solution P to solution $\bar{P}$. In $\bar{P}$, a new facility $\bar{y}_3$ is selected in place of y_3 in P, $\bar{y}_3$ attracts one of the "critical vertices" from the neighborhood of the facility y_1. Although the cost of the two solutions is the same, the algorithm selects the new solution $\bar{P}$ because $\max_{\delta_{\bar{P}}} < \max_{\delta_P}$.

The main idea of our *plateau surfer* local search is to use the concept of critical vertex to escape from plateaus, moving to solutions that have either a better cost than the current solution or equal cost but less critical vertices. Figure 2 shows a simple application of the algorithm, while in Figures 3 and 4, for four benchmark instances, both Mladenović's local search and our local search are applied once taken

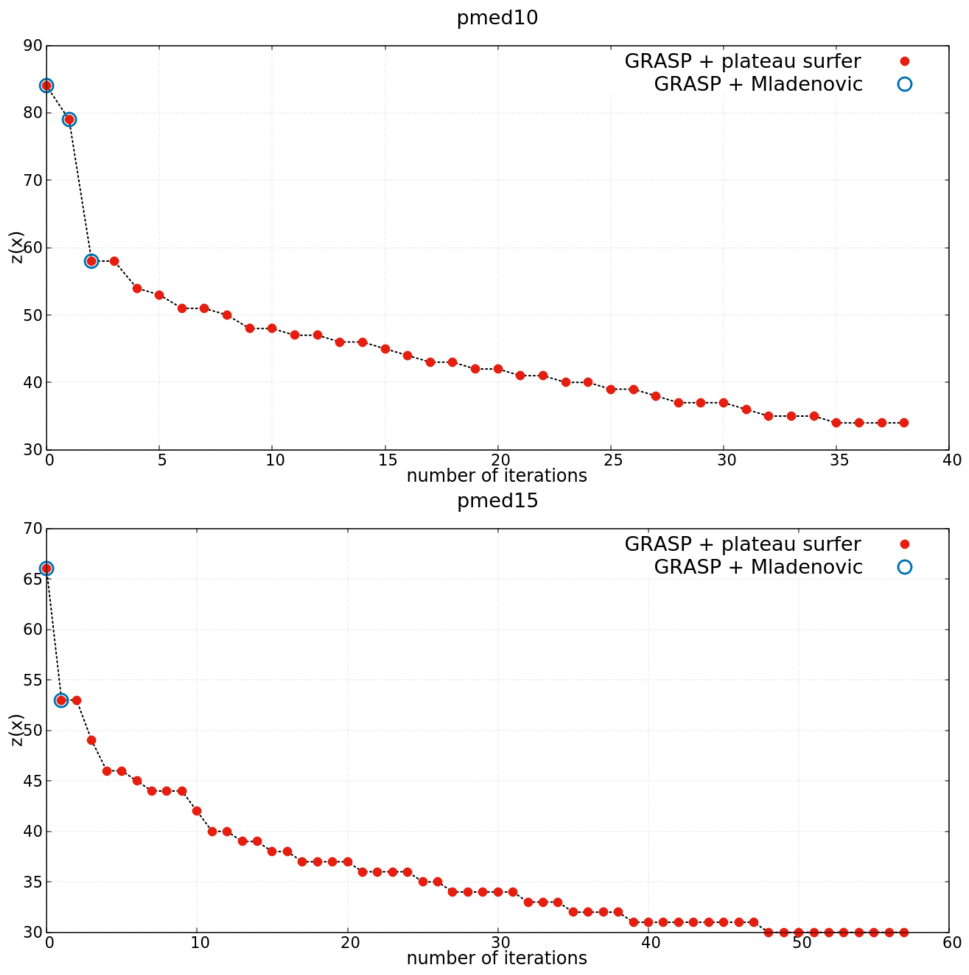
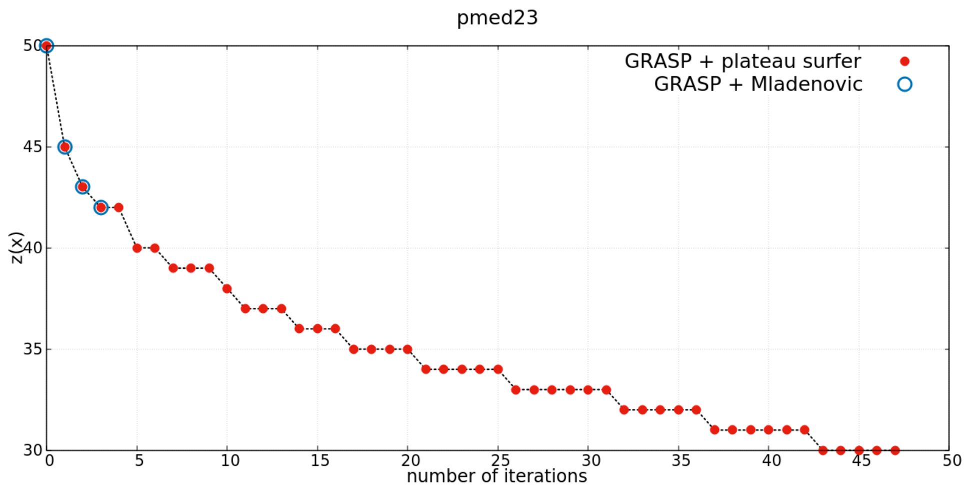


FIGURE 3. Plateau escaping. The behavior of our *plateau surfer* local search (in red) compared with the Mladenović's one (in blue). Both algorithms work on the same instances taking as input the same starting solution. Filled red dots and empty blue circles indicate the solutions found by the two algorithms. Mladenović local search stops as soon as the first plateau is met.

as input the same starting feasible solution. It is evident that both the procedures make the same first moves. However, as soon as a plateau is met, Mladenović's local search ends, while our local search is able to escape from the plateau moving to other solutions with the same cost value, and restarting the procedure from a new solution that can lead to a strict cost function improvement, .

Let us analyze in more detail the behavior of our local search, whose pseudo-code is reported in Figure 5. The main part of the algorithm consists in the portion of the pseudo-code that goes from line 0 to line 0. Starting from an initial solution P, the algorithm tries to improve the solution replacing a vertex $j \notin P$ with a facility $i \in P$. Clearly, this swap is stored as an improving move if the new solution



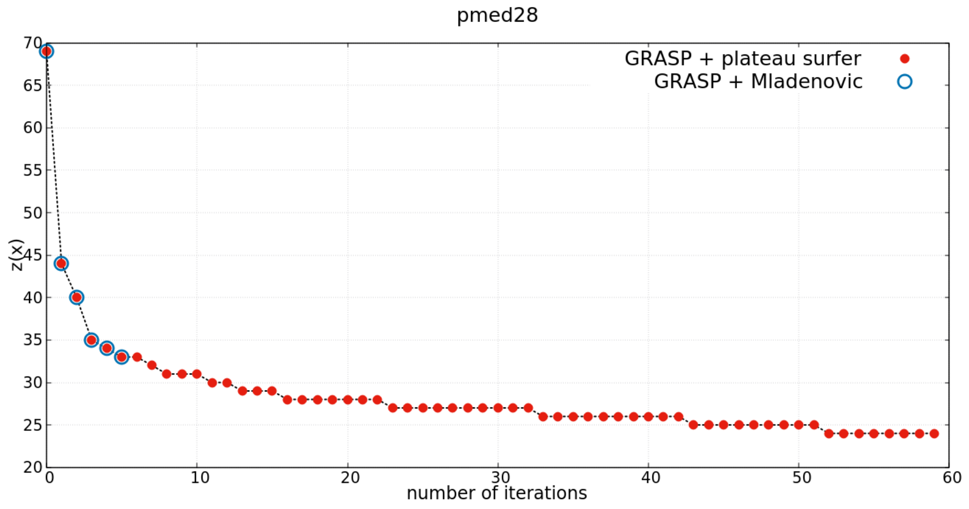


FIGURE 4. Plateau escaping. The behavior of our *plateau surfer* local search (in red) compared with the Mladenović's one (in blue) on other two different instances.

$\bar{P} = P \setminus \{i\} \cup \{j\}$ is strictly better than P according to the cost function $\mathbb{C}$. If $\mathbb{C}(\bar{P})$ is better than the current cost $\mathbb{C}(P)$, then $\bar{P}$ is compared also with the incumbent solution and if it is the best solution found so far, the incumbent is update and the swap that led to this improvement stored (lines 0–0).

Otherwise, the algorithm checks if it is possible to reduce the number of critical vertices. If the new solution $\bar{P}$ is such that $\bar{P} <_{cv} P$, then the algorithm checks if $\bar{P}$ is the best solution found so far (line 0), the value that counts the number of critical vertices in a solution is update (line 0), and the current swap stored as an improving move (line 0).

To study the computational complexity of our local search, let be $n = |V|$ and $p = |P|$, the number of vertices in the graph and the number of facilities in a solution, respectively. The loops at lines 0 and 0 are executed p and n times, respectively.

```
repeat
    modified := false;
    forall the $i \in P$ do
        best_flip := best_cv_flip := NIL;
        bestNewSolValue := $\mathbb{C}(P)$;
        best_cv := $\max_\delta(\bar{P})$;
        forall the $j \in V \setminus P$ do
            $\bar{P} := P \setminus \{i\} \cup \{j\}$;
            if $\mathbb{C}(\bar{P}) <$ bestNewSolValue then
                bestNewSolValue := $\mathbb{C}(\bar{P})$;
                best_flip := $j$;
            end
            else if best_flip = NIL and $\max_\delta(\bar{P}) <$ best_cv then
                best_cv := $\max_\delta(\bar{P})$;
                best_cv_flip := $j$;
            end
        end
        if best_flip $\neq$ NIL then
            $P := P \setminus \{i\} \cup \{$best_flip$\}$;
            modified := true;
        end
        else if best_cv_flip $\neq$ NIL then
            $P := P \setminus \{i\} \cup \{$best_cv_flip$\}$;
            modified := true;
        end
    end
until modified = false;
return $P$;
        Function plateau-surfer-local-search($G = \langle V, A, \mathbb{C} \rangle, P, p$)
```

FIGURE 5. Pseudocode of the plateau surfer local search algorithm based on the critical vertex concept.

The update of the solution takes $\mathcal{O}(n)$. In conclusion, the total complexity is $\mathcal{O}(p \cdot n^2)$.

4. Experimental Results

In this section, we describe computational experience with the local search proposed in this paper. We have compared it with the local search proposed by Mladenović et al. [21], by embedding both in a GRASP framework.

The algorithms were implemented in C++, compiled with `gcc 5.2.1` under Ubuntu with `-std=c++14` flag. The stopping criterion is $maxTime = 0.1 \cdot n + 0.5 \cdot p$. All the tests were run on a cluster of nodes, connected by 10 Gigabit Infiniband technology, each of them with two processors Intel Xeon E5-4610v2@2.30GHz.

Table 1 summarizes the results on a set of `ORLIB` instances, originally introduced in [1]. It consists of 40 graphs with number of vertices ranging from 100 to 900, each with a suggested value of p ranging from 5 to 200. Each vertex is both a user and a potential facility, and distances are given by shortest path lengths. Tables 2 and 3 report the results on the TSP set of instances. They are just sets of points on the plane. Originally proposed for the traveling salesman problem, they are available from the TSPLIB [24]. Each vertex can be either a user or an open facility. We used the *Mersenne Twister* random number generator by Matsumoto and Nishimura [19]. Each algorithm was run with 10 different seeds, and minimum (min), average (E) and variance (σ^2) values are listed in each table. The second to last column lists the %-Gap between average solutions. To deeper investigate the statistical significance of the results obtained by the two local searches, we performed the Wilcoxon test [2, 27].

Generally speaking, the Wilcoxon test is a ranking method that well applies in the case of a number of paired comparisons leading to a series of differences, some of which may be positive and some negative. Its basic idea is to substitute scores $1, 2, 3, \ldots, n$ with the actual numerical data, in order to obtain a rapid approximate idea of the significance of the differences in experiments of this kind.

TABLE 1. Results on ORLIB instances.

	GRASP + mladenovic			GRASP + plateau-surfer				
Instance	min	E	σ^2	min	E	σ^2	%-Gap	p-value
pmed01	127	127	0	127	127	0	0.00	
pmed02	98	98	0	98	98	0	0.00	
pmed03	93	93.14	0.12	93	93.54	0.25	0.43	
pmed04	74	76.21	1.33	74	74.02	0.04	-2.87	1.20E-16
pmed05	48	48.46	0.43	48	48	0	-0.95	
pmed06	84	84	0	84	84	0	0.00	
pmed07	64	64.15	0.27	64	64	0	-0.23	
pmed08	57	59.39	1.36	55	55.54	0.73	-6.48	3.37E-18
pmed09	42	46.87	2.83	37	37.01	0.01	-21.04	2.80E-18
pmed10	29	31.21	0.81	20	20.01	0.01	-35.89	9.38E-19
pmed11	59	59	0	59	59	0	0.00	
pmed12	51	51.89	0.1	51	51.41	0.24	-0.93	
pmed13	42	44.47	0.73	36	36.94	0.06	-16.93	1.04E-18
pmed14	35	38.59	3.24	26	26.85	0.13	-30.42	2.11E-18
pmed15	28	30.23	0.7	18	18	0	-40.46	1.09E-18
pmed16	47	47	0	47	47	0	0.00	
pmed17	39	40.71	0.23	39	39	0	-4.20	8.69E-20
pmed18	36	37.95	0.29	29	29.41	0.24	-22.50	6.37E-19
pmed19	27	29.32	0.42	19	19.13	0.11	-34.75	6.25E-19
pmed20	25	27.05	0.99	14	14	0	-48.24	1.46E-18
pmed21	40	40	0	40	40	0	0.00	
pmed22	39	40.06	0.24	38	38.94	0.06	-2.80	1.30E-18
pmed23	30	32.02	0.44	23	23.21	0.17	-27.51	7.16E-19
pmed24	24	25.38	0.34	16	16	0	-36.96	4.37E-19
pmed25	22	22.62	0.24	11	11.89	0.1	-47.44	2.77E-19
pmed26	38	38	0	38	38	0	0.00	
pmed27	33	33.96	0.06	32	32	0	-5.77	2.15E-22
pmed28	26	26.78	0.17	19	19	0	-29.05	2.20E-20
pmed29	23	23.43	0.31	13	13.68	0.22	-41.61	8.00E-19
pmed30	20	21.18	0.47	10	10	0	-52.79	6.50E-19
pmed31	30	30	0	30	30	0	0.00	
pmed32	30	30.37	0.23	29	29.62	0.24	-2.47	
pmed33	23	23.76	0.2	16	16.28	0.2	-31.48	4.31E-19
pmed34	21	22.42	0.66	11	11.56	0.25	-48.44	1.59E-18
pmed35	30	30.01	0.01	30	30	0	-0.03	
pmed36	28	29.37	0.31	27	27.65	0.23	-5.86	4.52E-18
pmed37	23	24.07	0.37	16	16	0	-33.53	2.74E-19
pmed38	29	29	0	29	29	0	0.00	
pmed39	24	25.08	0.11	23	23.98	0.02	-4.39	4.68E-21
pmed40	20	21.81	0.43	14	14	0	-35.81	5.14E-19
						Average	-16.78	

More formally, let A_1 and A_2 be two algorithms, $I_1, \ldots, I_l$ be l instances of the problem to solve, and let $\delta_{A_i}(I_j)$ be the value of the solution obtained by algorithm A_i ($i = 1, 2$) on instance I_j ($j = 1, \ldots, l$). For each $j = 1, \ldots, l$, the Wilcoxon test computes the differences $\Delta_j = |\delta_{A_1}(I_j) - \delta_{A_2}(I_j)|$ and sorts them in non decreasing order. Accordingly, starting with a smallest rank equal to 1, to

Table 2. Results on TSPLIB instances (1)

		GRASP + mladenovic			GRASP + plateau-surfer				
Instace	p	min	E	σ^2	min	E	σ^2	%-Gap	p-value
pcb3038	50	534.48	608.49	1068.09	355.68	374.66	51.05	-38.43	3.90E-18
	100	399.49	481.75	1285.58	259.67	270.2	17.56	-43.91	3.90E-18
	150	331.62	428.69	1741.11	206.71	215.78	23.73	-49.67	3.90E-18
	200	301.01	386.56	3161.87	177.79	190.88	10.4	-50.62	3.90E-18
	250	292.48	359.59	3323.62	155.03	163.75	19.24	-54.46	3.90E-18
	300	261.28	349.42	2902.71	143.39	151.89	10.04	-56.53	3.90E-18
	350	258.82	336.08	3755.72	123.85	136.22	22.45	-59.47	3.90E-18
	400	249.78	337.14	4033.46	119.07	122.31	1.2	-63.72	3.90E-18
	450	214.97	321.36	3373.23	115	117	0.6	-63.59	3.90E-18
	500	209.35	299.4	3378.98	102	110.38	5.78	-63.13	3.90E-18
pr1002	10	3056.55	3313.49	10132.77	2616.3	2727.45	2260.95	-17.69	3.90E-18
	20	2404.16	2668.29	8244.65	1806.93	1886.89	1516.07	-29.28	3.90E-18
	30	2124.26	2358.07	4432.11	1456.02	1505.55	910.93	-36.15	3.89E-18
	40	1960.23	2172.63	7831.77	1253.99	1302.76	751.62	-40.04	3.90E-18
	50	1755.7	1992.08	5842.66	1097.72	1156.77	815.35	-41.93	3.90E-18
	60	1697.79	1865.5	5872.47	1001.25	1042.82	257.42	-44.1	3.89E-18
	70	1569.24	1736.41	4078.39	900	954.04	307.65	-45.06	3.89E-18
	80	1486.61	1633.87	3278.4	851.47	889.5	407.29	-45.56	3.88E-18
	90	1350.93	1543.17	3922.25	764.85	809.78	382.29	-47.52	3.89E-18
	100	1312.44	1472.47	2616	743.3	767.62	77.4	-47.87	3.89E-18
pr439	10	2575.12	2931.83	38470.59	1971.83	1972.28	19.61	-32.73	3.79E-18
	20	1940.52	2577.03	23638.88	1185.59	1194.12	124.58	-53.66	3.71E-18
	30	1792.34	2510.91	23692.47	886	919.1	442.37	-63.4	3.89E-18
	40	1525.2	2413.33	53876.4	704.45	728.19	39.31	-69.83	3.88E-18
	50	1358.54	2252.46	89633.71	575	595.4	64.21	-73.57	3.82E-18
	60	1386.09	2170.85	110065.93	515.39	537.66	75.43	-75.23	3.89E-18
	70	1370.45	1898.53	116167.77	480.23	499.65	4.93	-73.68	3.73E-18
	80	1140.18	1815.1	118394.68	424.26	440.27	166.06	-75.74	3.89E-18
	90	1191.9	1699.64	91388.99	400	406.17	31.71	-76.1	3.88E-18
	100	1190.85	1679.73	94076.45	375	384.27	98.91	-77.12	3.89E-18
rat575	10	81.32	92.98	9.27	73	74.71	0.79	-19.65	3.90E-18
	20	68.07	73.86	3.7	50.54	53.04	0.63	-28.19	3.90E-18
	30	59.81	64.61	3.67	41.79	43.53	0.47	-32.63	3.90E-18
	40	54.13	58.37	3.43	36.12	37.43	0.29	-35.87	3.90E-18
	50	47.68	53.78	3.56	32.45	33.36	0.17	-37.97	3.90E-18
	60	45.62	50.03	3.21	29.15	30.17	0.19	-39.7	3.90E-18
	70	43.68	46.96	2.97	27	27.78	0.13	-40.84	3.90E-18
	80	39.81	44.2	2.75	25.02	25.99	0.11	-41.2	3.90E-18
	90	38.48	41.98	2.06	23.85	24.4	0.07	-41.88	3.90E-18
	100	37.01	39.93	1.4	22.2	23.01	0.08	-42.37	3.89E-18
rat783	10	102.22	110.93	13.17	83.49	87.82	1.65	-20.83	3.90E-18
	20	80.53	88.56	7.68	59.68	62.8	1.41	-29.09	3.90E-18
	30	69.58	76.92	7.87	49.25	51.48	0.74	-33.07	3.90E-18
	40	62.97	69.63	4.62	42.05	44.27	0.53	-36.42	3.90E-18
	50	59.41	65.26	5.59	38.29	39.6	0.42	-39.32	3.90E-18
	60	54.82	60.35	4.77	34.48	35.92	0.24	-40.48	3.90E-18
	70	49.4	56.56	7.48	32.06	33.11	0.24	-41.46	3.90E-18
	80	48.51	53.76	4.03	29.55	30.94	0.21	-42.45	3.90E-18
	90	46.07	51.82	3.53	28.18	28.85	0.11	-44.33	3.90E-18
	100	43.97	49.5	4.68	26.31	27.49	0.14	-44.46	3.90E-18
							Average	-46.84	

each difference Δ_j, it assigns a non decreasing rank R_j. Ties receive a rank equal to the average of the sorted positions they span. Then, the following quantities are computed

$$W^+ = \sum_{j=1,\dots,l:\ \Delta_j>0} R_j,$$

$$W^- = \sum_{j=1,\dots,l:\ \Delta_j<0} R_j.$$

Under the null hypothesis that $\delta_{A_1}(I_j)$ and $\delta_{A_2}(I_j)$ have the same median value, it should result that $W^+ = W^-$. If the p-value associated to the experiment is less than an a priori fixed significance level α, then the null hypothesis is rejected and the difference between W^+ and W^- is considered significant.

The last column of each table lists the p-values where the %-Gap is significant, all the values are less than $\alpha = 0.01$. This outcome of the Wilcoxon test further confirms that our local search is better performing than the local search proposed by Mladenović et al.

Table 3. Results on TSPLIB instances (2)

		GRASP + mladenovic			GRASP + plateau-surfer				
Instace	p	min	E	σ^2	min	E	σ^2	%-Gap	p-value
rl1323	10	3810.84	4185.89	24655.46	3110.57	3241.79	3290.56	-22.55	3.90E-18
	20	2996.4	3348.31	23183.21	2090.87	2236.28	2798.56	-33.21	3.90E-18
	30	2689.44	2979.79	14205.75	1730.78	1808.94	1544.85	-39.29	3.90E-18
	40	2337.92	2712.93	14193.05	1479.24	1576.25	1710.4	-41.9	3.90E-18
	50	2195.91	2462.95	9835.09	1300	1363.88	950.66	-44.62	3.90E-18
	60	2021.87	2278.94	16400.27	1181.3	1244.03	657.55	-45.41	3.90E-18
	70	1900.77	2128.45	11883.58	1076.2	1127.98	475.13	-47	3.90E-18
	80	1866.8	2033.24	4501.73	988.87	1048.87	438.82	-48.41	3.89E-18
	90	1634.37	1966.13	4643.42	935.02	978.6	289.18	-50.23	3.89E-18
	100	1631.5	1909.56	8483.55	886.85	914	238.2	-52.14	3.89E-18
u1060	10	3110.65	3373.87	7541.61	2301.7	2440	599.42	-27.68	3.86E-18
	20	2652.6	2818.37	5787.51	1650.34	1749.15	2814.03	-37.94	3.90E-18
	30	2501.72	2684.87	3811.23	1302.94	1373.21	912.92	-48.85	3.90E-18
	40	2442.07	2616.15	5267.85	1118.59	1176.14	593.6	-55.04	3.90E-18
	50	2378.36	2591.96	7266.77	950.66	1021.55	418.91	-60.59	3.90E-18
	60	2301.83	2602.13	13579.82	860.49	919.97	374.54	-64.65	3.90E-18
	70	2378.36	2606.64	10944.09	790.13	828.16	441.03	-68.23	3.90E-18
	80	2351.82	2622.32	12980.39	720.94	753.64	306.94	-71.26	3.90E-18
	90	2248.61	2562.01	10260.36	667.55	708.04	107.79	-72.36	3.90E-18
	100	2060.29	2494.08	11025.91	632.11	653.15	110.65	-73.81	3.90E-18
	110	2049.18	2444.22	10385.95	570.49	613.02	148.7	-74.92	3.90E-18
	120	2122.97	2406.19	9191.4	570	579.93	96.23	-75.9	3.90E-18
	130	1839.55	2390.82	12029.95	538.82	561.62	78.78	-76.51	3.90E-18
	140	1924.48	2316.25	12982.87	500.39	527.66	172.51	-77.22	3.90E-18
	150	1942.27	2300.45	13245.06	499.65	503.26	20.49	-78.12	3.90E-18
u1817	10	592.97	646.89	325	466.96	485.44	104.33	-24.96	3.90E-18
	20	462.3	564.44	560.9	330.2	348.15	53.96	-38.32	3.90E-18
	30	418.91	530.34	1018.29	265.19	283.4	58.43	-46.56	3.90E-18
	40	407.19	526.44	956.01	232.25	245.78	43.42	-53.31	3.90E-18
	50	330.21	507.52	2889.76	204.79	217.05	26.96	-57.23	3.90E-18
	60	352.88	497.35	3539.09	184.91	197.26	21.79	-60.34	3.90E-18
	70	321.27	477.43	4139.93	170.39	181.53	13.67	-61.98	3.90E-18
	80	289.61	445.35	4866.81	154.5	166.46	22.68	-62.62	3.90E-18
	90	283.99	422.34	3828.2	148.11	153.5	13.72	-63.65	3.90E-18
	100	283.99	416.69	2660.21	136.79	146.67	7.4	-64.8	3.90E-18
							Average	-54.9	

5. Concluding Remarks

In this paper, we presented a new local search heuristic for the p-center problem, whose potential applications appear in telecommunications, in transportation logistics, and whenever one must to design a system to organize some sort of public facilities, such as, for example, schools or emergency services.

The computational experiments show that the proposed local search is capable to reduce the number of local optimum solutions using the concept of *critical vertex*, and it improves the results of the best local search for the problem.

Future lines of work will be focused on a deepeer investigation of the robustness of our proposal by applying it on further instances coming from financial markets and manifacturing systems.

Acknowledgements

This work has been realized thanks to the use of the S.Co.P.E. computing infrastructure at the University of Napoli FEDERICO II.

References

[1] J.E. Beasley. A note on solving large p-median problems. *European Journal of Operational Research*, 21:270–273, 1985.

[2] M. Coffin and M.J. Saltzman. Statistical analysis of computational tests of algorithms and heuristics. *INFORMS Journal on Computing*, 12(1):24–44, 2000.

[3] M. Daskin. *Network and Discrete Location: Models, Algorithms, and Applications*. Wiley, New York, 1995.

[4] T. Davidović, D. Ramljak, M. Šelmić, and D. Teodorović. Bee colony optimization for the p-center problem. *Computers and Operations Research*, 38(10):1367–1376, 2011.

[5] M.E. Dyer and A.M. Frieze. A simple heuristic for the p-centre problem. *Operations Research Letters*, 3(6):285–288, 1985.

[6] T.A. Feo and M.G.C. Resende. A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, 8:67–71, 1989.

[7] T.A. Feo and M.G.C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133, 1995.

[8] P. Festa and M.G.C. Resende. GRASP: An annotated bibliography. In C.C. Ribeiro and P. Hansen, editors, *Essays and Surveys on Metaheuristics*, pages 325–367. Kluwer Academic Publishers, 2002.

[9] P. Festa and M.G.C. Resende. An annotated bibliography of GRASP – Part I: Algorithms. *International Transactions in Operational Research*, 16(1):1–24, 2009.

[10] P. Festa and M.G.C. Resende. An annotated bibliography of GRASP – Part II: Applications. *International Transactions in Operational Research*, 16(2):131–172, 2009.

[11] Boris Goldengorin, Anton Kocheturov, and Panos M. Pardalos. A pseudo-boolean approach to the market graph analysis by means of the p-median model. In Fuad Aleskerov, Boris Goldengorin, and Panos M. Pardalos, editors, *Clusters, Orders, and Trees: Methods and Applications: In Honor of Boris Mirkin's 70th Birthday*, pages 77–89. Springer, New York, 2014.

[12] Boris Goldengorin, Dmitry Krushinsky, and Panos M. Pardalos. *Application of the PMP to Cell Formation in Group Technology*, pages 75–99. Springer New York, 2013.

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

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

[15] D. Hochbaum and D. Shmoys. A best possible heuristic for the k-Center problem. *Mathematics of Operations Research*, 10(2):180–184, 1985.

[16] O. Kariv and S.L. Hakimi. An Algorithmic Approach to Network Location Problems. Part I: The p-Centers. *SIAM Journal on Applied Mathematics*, 37(3):513–538, 1979.

[17] O. Kariv and S.L. Hakimi. An Algorithmic Approach to Network Location Problems. Part II: The p-Medians. *SIAM Journal on Applied Mathematics*, 37(3):539–560, 1979.

[18] J. S. Martinich. A vertex-closing approach to the p-center problem. *Naval Research Logistics*, 35(2):185?201, 1988.

[19] Makoto Matsumoto and Takuji Nishimura. Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.*, 8(1):3–30, January 1998.

[20] E. Minieka. The m-Center Problem. *SIAM Rev.*, 12(1):138–139, 1970.

[21] N. Mladenović, M. Labbé, and P. Hansen. Solving the p-Center Problem with Tabu Search and Variable Neighborhood Search. *Networks*, 42(April):48–64, 2003.

[22] Nenad Mladenović, Dragan Urosevic, Dionisio Pérez-Brito, and Carlos G. García-González. Variable neighbourhood search for bandwidth reduction. *European Journal of Operational Research*, 200(1):14 – 27, 2010.

[23] Eduardo G. Pardo, Nenad Mladenović, Juan J. Pantrigo, and Abraham Duarte. Variable formulation search for the cutwidth minimization problem. *Applied Soft Computing*, 13(5):2242 – 2252, 2013.

[24] Gerhard Reinelt. TSPLIB – A traveling salesman problem library. *ORSA Journal on Computing*, 3(4):376–384, 1991.

[25] M.G.C. Resende and R.F. Werneck. A Hybrid Heuristic for the p-Median Problem. *Journal of Heuristics*, 10(1):59–88, 2004.

[26] S. Salhi and A. Al-Khedhairi. Integrating heuristic information into exact methods: The case of the vertex p-centre problem. *Journal of the Operational Research Society*, 61(11):1619–1631, 2010.

[27] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

(Daniele Ferone) Department of Mathematics and Applications, University of Napoli Federico II, Compl. MSA, Via Cintia, 80126, Napoli, Italy.
E-mail address: `daniele.ferone@unina.it`

(Paola Festa) Department of Mathematics and Applications, University of Napoli Federico II, Compl. MSA, Via Cintia, 80126, Napoli, Italy.
E-mail address: `paola.festa@unina.it`

(Antonio Napoletano) Department of Mathematics and Applications, University of Napoli Federico II, Compl. MSA, Via Cintia, 80126, Napoli, Italy.
E-mail address: `antonio.napoletano2@unina.it`

(Mauricio G.C. Resende) Amazon.com and Industrial & Systems Engineering, University of Washington, Seattle, WA USA.
E-mail address: `mgcr@uw.edu`