On the nested p-center problem

Christof Brandstetter
Institute of Business Analytics and Technology Transformation/JKU Business School, Johannes Kepler University Linz
Linz, Austria
christof.brandstetter@jku.at

Markus Sinnl
Institute of Business Analytics and Technology Transformation/JKU Business School, Johannes Kepler University Linz
Linz, Austria
markus.sinnl@jku.at

ABSTRACT

In recent years, the concept of nesting gained renewed interest within the location science community. Nesting allows to model temporal aspects in planning, which are highly relevant in practice. In this paper, we introduce the nested p-center problem, which is an extension of the classic p-center problem. In this problem we are given a finite time horizon and at each time period, we are allowed to open a given number of facilities. The sets of open facilities at each time period must fulfill the nesting property, i.e., the open facilities at an earlier time period must be a subset of the open facilities at a later time period. The objective function is the sum of the objective function values of the individual periods and the goal is to minimize this objective function. The objective function value of each period is the maximal distance between a customer and its closest open facility. We present two mixed integer programming formulations for this problem. We provide a computational study on well-known p-center instances from literature to assess the performance of the two formulations and also to analyse the effect of nesting.

1 INTRODUCTION

Consistency is very important in long term planning and in particular in the location of facilities. However, many facility location problems potentially provide inconsistent solutions for varying numbers of open facilities, i.e., for different numbers of allowed open facilities the optimal locations can be vastly different. In practice, this could result in opening and closing of facilities when facing a long term planning project, where initially some facilities are to be built, and at some later time steps additional facilities are to be built. This is of course undesirable for a variety of reasons such as monetary cost or environmental cost.

The first ideas of modeling a facility location problem with such consistency in mind appeared in the 1970s in works by Scott [20] and Roodman and Schwarz [19]. In these works, the authors describe certain *multi-period facility location problems*, where the number of open facilities is changed over time, but the open facilities cannot be relocated. We note that location problems needing to fulfill constraints of this type such as the particular *nesting property* which is considered in our work (see Definition 1 for details) can be categorized as one problem family within the area of the multi-period problems (see Chapter 11 of [15] for a general overview of this area). However, most of the existing recent work regarding such multi-period location problems usually focuses on varying demand, distances, or cost over time, see e.g., [4, 7]. In 2022, McGarvey and Thorsen [16] revisited nesting and applied the concept to the p-median problem. They also posed the question about applying the nesting property to other classical location problems, such as the maximum coverage problem or the p-center problem. In this paper, we follow up on this open question by considering the latter.

The *(discrete) nested p-center problem (n-pCP)* can be defined the following way:

Definition 1. We are given a set $\mathcal{I}$ of customer demand points, a set $\mathcal{J}$ of potential facility locations and distances $d_{ij} \geq 0$ between each $i \in \mathcal{I}$ and $j \in \mathcal{J}$. Additionally, we are given a time horizon $\mathcal{H} = \{1, \ldots, H\}$ and a set of integers $\mathcal{P} = \left\{p^1, \ldots p^H\right\}$ where $p^h \leq p^{h+1}$ for $h = 1, \ldots, H-1$ and $p^H \leq |\mathcal{J}|$.

A feasible solution to the nested p-center problem consists of a set $\mathcal{J}^h \subseteq \mathcal{J}$ with $\left|\mathcal{J}^h\right| = p^h$ for $h \in \mathcal{H}$. Moreover, the *nesting property* must be fulfilled by these sets, i.e., $\mathcal{J}^h \subseteq \mathcal{J}^{h+1}$ must hold for $h = 1, \ldots, H-1$.

For a given time period $h \in \mathcal{H}$ and set $\mathcal{J}^h$, let $d_h(\mathcal{J}^h) = \max_{i \in I} \min_{j \in \mathcal{J}^h} d_{ij}$. The objective function value of a given feasible solution is defined as $\sum_{h=1}^{H} d_h(\mathcal{J}^h)$ and the goal is to find a feasible solution with minimal objective function value.

Observation 1. The objective function of the n-pCP can be viewed as minimizing the sum of the *regrets* over the time periods, where the regret of a given n-pCP-solution for a time period h is defined as the difference between the objective function value of the n-pCP-solution for the time period and the optimal p-center value for $p = p^h$. We note that the minimization of regret is a popular concept when dealing with uncertainty, see, e.g., [21] and was also considered in [16].

Observation 2. For $|\mathcal{P}| = 1$ the problem reduces to the (classical) p-center problem (pCP) which was introduced by Hakimi [12] in 1964. The pCP is NP-hard for $p \geq 2$ [14].

Observation 3. In our definition of the n-pCP we have that the number of facilities to be allowed open is non-decreasing over the time horizon. The optimal solution to this problem is also the optimal solution to the problem variant, where the number of facilities to be allowed open is non-increasing over the time horizon (and the nesting property is accordingly adapted to $\mathcal{J}^{h+1} \subseteq \mathcal{J}^h$), as these problems are equivalent.

Figure 1 shows an instance of the (nested) p-center problem (the `eil51` instance of the `TSPlib`[18]). In this instance, we have that $\mathcal{I} = \mathcal{J}$, i.e., each point (visualized as gray dot) is a demand point

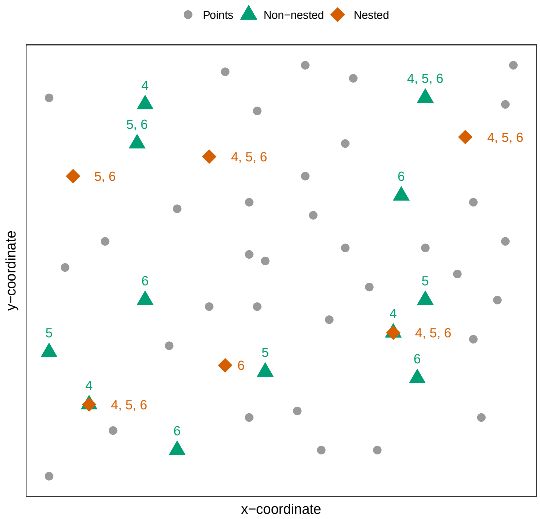

Figure 1: Instance: `eil51` and optimal solutions for n-pCP with $\mathcal{P} = \{4, 5, 6\}$ and pCP for $p = 4, 5, 6$

and can also be used as potential facility location. The distance between two points in this instance is the Euclidean distance. In this figure, next to illustrating an optimal solution of the n-pCP for $H = 3$ with $\mathcal{P} = \{4, 5, 6\}$, we also illustrate optimal solutions for the pCP when solving it for $p = 4, 5, 6$ individually. The nested solution is visualized using green triangles, with a green number above a triangle indicating that the facility is open at this location in the time period where p^h is this number. For example, if the numbers four, five, and six are besides the diamonds, it means that this location is used in all three time periods for the nested solution. The solutions for the pCP are visualized using the orange rectangles, with the orange numbers indicating that a facility is open at the location in the optimal solution where p is this number. Note that the optimal solutions for the pCP open twelve different facilities in total and only one facility which was in the solution for $p = 4$ was also in the solution for $p = 6$. The optimal objective function value for the n-pCP is 61, while the objective function values for the pCP for $p = 4, 5, 6$ are 22, 19 and 17, giving a value of 58 in total. The sum of the regrets is 61-58=3.

1.1 Contribution and outline

In this paper, we introduce the nested p-center problem and provide two mixed-integer linear programming (MILP) formulations for it. Based on these formulations, we conduct a computational study on well-known pCP instances from literature to assess the performance of the two approaches and also identify their limitations and potential areas for their improvement.

The paper is structured as follows: In the remainder of this section, we discuss previous and related work to the pCP and the nesting property. In Section 2 we present our two MILP formulations and Section 3 contains the computational study. Section 4 concludes the paper with an outlook to potential improvements to be considered in future work.

1.2 Literature review

For the pCP there exists considerable amount of work on heuristic and exact solution methods, as well as different adaptations and variants. We focus our literature review on existing exact methods, as our work is about the design of exact solution approaches. For the existing work on heuristic methods and approximation algorithms for the pCP we refer to the recent survey [11]. In 1970 the first exact solution approach for the pCP was developed [17] using the relationship to the set cover problem. Some recent state-of-the-art algorithms for the pCP also use this connection [5, 6]. The classical MILP formulation for the problem can be found in textbooks like in Chapter 5 of [8]. More recently, a compact formulation has been introduced in [9] and further extension of this formulation are presented in [1]. Their formulation has a binary variable y_j for $j \in \mathcal{J}$ to indicate at which point a facility opens and a binary variable u_k for each distinct distance D_k indicating whether the optimal value of pCP is less or equal than D_k. The authors show that their formulation has stronger linear programming (LP) relaxation bounds than the classical formulation. In [3] another compact formulation with the same strength of the LP-relaxation is presented.

In 2022 a new projection based formulation was introduced [10], which can be obtained by applying Benders Decomposition to the classical formulation [8]. Their formulation only uses binary variables y_j for $j \in \mathcal{J}$ indicating open facilities and a continuous variable which measures the distance in the objective function. The authors also present a lifting procedure for the inequalities in their formulation and show that the LP-relaxation bounds of their lifted formulation are the same as the bounds obtained by the formulations of [1, 3, 9].

Location problems, where facilities are opened iteratively over a certain time horizon, has been discussed firstly in the 70s by [20]. The authors compared a dynamic programming system, which takes into account the complete time horizon, with a myopic system, which optimizes the next period without considering any later periods. Their dynamic programming system outperformed the myopic system for larger time horizons. A first MILP formulation for such problems has been introduced in [19], where the authors try to minimize the operational cost of closing facilities iteratively over a certain time horizon and where the first to use a nesting-type constraint which enforces open facilities to be open until the end of the time horizon. Further, they presented a generalization of the formulation where they start with a set of facilities which are open at the beginning and a set of potential facilities which can be opened. The starting facilities can be closed at any point in the time horizon, but once closed must remain closed, while the potential facilities can be opened, but not closed again. This allows for a restricted redistribution of facilities. More recently, the nesting concept was revisited and applied it to the p-median problem for two objective functions, minimizing the sum of the regrets and minimizing maximum regret by [16].

Incremental facility location and network design problems, f.e. [2, 13] are multi-period problems, in which a network is incrementally extended by the means of adding arcs, facilities or nodes to maintain incrementally increasing coverage requirements in each period while optimizing some objective like minimization of total cost or maximizing the cumulative flow over the planning horizon.

2 MIXED INTEGER LINEAR PROGRAMMING FORMULATIONS

In this section, we begin by presenting a first formulation of the nested p-center problem based on the textbook pCP formulation (see, e.g., [8]). Afterward, we present a second formulation based on the pCP formulation presented in Section 2.1 of [1].

2.1 First formulation

Our first formulation (nPC1) for the n-pCP uses two sets of binary variables, denoted as x and y. The variable x_{ij}^h is indicating if customer demand point $i \in \mathcal{I}$ is assigned to potential facility location $j \in \mathcal{J}$ in time period h and the variable y_j^h is indicating if a facility is opened at the potential facility location j in time period h. The continuous variables R^h measure the maximum distance from any customer demand point to its nearest open facility in time period h.

$$
\begin{aligned}
(\texttt{nPC1})\ \min \quad & \sum_{h\in\mathcal{H}} R^h && && \text{(1a)}\\
\text{s.t.} \quad & \sum_{j\in\mathcal{J}} y_j^h = p^h && \forall h \in \mathcal{H} && \text{(1b)}\\
& \sum_{j\in\mathcal{J}} x_{ij}^h = 1 && \forall i \in \mathcal{I}, \forall h \in \mathcal{H} && \text{(1c)}\\
& \sum_{j\in\mathcal{J}} d_{ij} x_{ij}^h \le R^h && \forall i \in \mathcal{I}, \forall h \in \mathcal{H} && \text{(1d)}\\
& x_{ij}^h \le y_j^h && \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, h \in \mathcal{H} && \text{(1e)}\\
& y_j^h \le y_j^{h+1} && \forall j \in \mathcal{J}, \forall h \in \mathcal{H} \setminus \{H\} && \text{(1f)}\\
& x_{ij}^h \in \{0,1\} && \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, h \in \mathcal{H} && \text{(1g)}\\
& y_j^h \in \{0,1\} && \forall j \in \mathcal{J}, h \in \mathcal{H} && \text{(1h)}\\
& R^h \in \mathbb{R}_{\ge 0} && \forall h \in \mathcal{H} && \text{(1i)}
\end{aligned}
$$

The objective function (1a) minimizes the sum over the distances R^h over all time periods. The constraints (1b) ensure that p_h facilities are opened in time period h. Constraints (1c) ensure that each customer is only assigned to one facility in each time period. The constraints (1d) are pushing the decision variables R^h to the largest distance of any assigned customer-facility combination in each time period. Each customer can only be assigned to an open facility, which is ensured by constraints (1e). The nesting constraints (1f) ensure that each facility, which is opened in time period h, is also open in time period $h + 1$, so once a facility is opened in a time period, it cannot be closed in later time periods. Without this constraint, the formulation would just represent the sum over the individual p-center problems for each time period. The remaining constraints (1g) - (1i) are the binary constraints for the variables x_{ij}^h and y_j^h and the non-negativity constraint for variables R^h.

2.2 Second formulation

Our second formulation uses the binary variable y_j^h for $j \in \mathcal{J}$ and $h \in \mathcal{H}$ to indicate the open facilities analogously to the formulation (nPC1). Furthermore, let $\mathcal{D} = \{d_{ij} : i \in \mathcal{I}, j \in \mathcal{J}\}$ denote the set of all possible distances and let $D_1 \le \ldots \le D_K$ be the values contained in $\mathcal{D}$, so $\mathcal{D} = \{D_1, \ldots, D_K\}$. Let $\mathcal{K}$ be the set of indices in $\mathcal{D}$.

For a $k \in \mathcal{K}$ the binary variables u_k^h indicate if the objective function value in time period h (measured by continuous variable R^h) is greater or equal than D_k. For customer $i \in \mathcal{I}$ let the set $\mathcal{S}_i$ be the set of indices $k \in \mathcal{K}$ for which there exists a facility $j \in \mathcal{J}$ with $d_{ij} = D_k$.

$$
\begin{aligned}
(\texttt{nPC2})\ \min \quad & \sum_{h\in H} R^h && && \text{(2a)}\\
\text{s.t.} \quad & \sum_{j\in J} y_j^h = p^h && \forall h \in \mathcal{H} && \text{(2b)}\\
& D_0 + \sum_{k=1}^{K} (D_k - D_{k-1})\ u_k^h \le R^h && \forall h \in \mathcal{H} && \text{(2c)}\\
& u_k^h + \sum_{j: d_{ij} < D_k} y_j^h \ge 1 && \forall i \in \mathcal{I},\ \forall k \in \mathcal{S}_i \cup \{K\}, \forall h \in \mathcal{H} && \text{(2d)}\\
& u_k^h \ge u_{k+1}^h && \forall k \in \mathcal{K} \setminus \{K\}, \forall h \in \mathcal{H} && \text{(2e)}\\
& y_j^h \le y_j^{h-1} && \forall j \in \mathcal{J}, \forall h \in \mathcal{H} && \text{(2f)}\\
& y_j^h \in \{0,1\} && \forall j \in \mathcal{J}, \forall h \in \mathcal{H} && \text{(2g)}\\
& u_k^h \in \{0,1\} && \forall k \in \mathcal{K}, \forall h \in \mathcal{H} && \text{(2h)}\\
& R^h \in \mathbb{R} && \forall h \in \mathcal{H} && \text{(2i)}
\end{aligned}
$$

The objective function (2a) minimizes the sum over the distances R^h over all time periods. The correct value of the R^h-variables is ensured by constraints (2c). The constraints (2b) ensure that no more than p^h facilities are opened in each time period. Constraint (2d) is ensuring that if for any customer i in time period h no facility j with smaller distance than D_k is opened u_k^h has to be one. Since (2d) is not defined for all $k \in \mathcal{K}$ but only for the subsets based on $\mathcal{S}_i \cup \{K\}$, constraints (2e) are necessary in order to ensure that no u_k^h can equal zero if u_{k+1}^h is one (otherwise constraints (2c) would not measure the distance correctly). The inequalities (2f) are for the nesting and are the same as (1f). The remaining constraints are the binary and non-negativity constraints, respectively. This formulation has $O((|\mathcal{I}|+|\mathcal{K}|)\,|\mathcal{H}|)$ variables and $O(\min\{|\mathcal{I}|\,|\mathcal{J}|, |\mathcal{I}|\,|\mathcal{K}|\}\,|\mathcal{H}|)$ constraints instead of $O(|\mathcal{I}|\,|\mathcal{J}|\,|\mathcal{H}|)$ variables and constraints in (nPC1). Depending on $|\mathcal{K}|$ this can be a significant reduction in both variables and constraints.

Observation 4. There exist instances of the n-pCP where the LP relaxation bound of (nPC2) is stronger than the LP relaxation bound of (nPC1). This is a direct consequence of the fact that for instances with $|\mathcal{P}| = 1$ (i.e., the pCP) both formulations reduce to their classical pCP-formulation counterparts and that such a result is known for these pCP-formulations, see, e.g., [1, 9].

3 COMPUTATIONAL RESULTS

The formulations from Section 2 have been implemented in C++ with CPLEX 20.1 as MILP-solver and were run on a single core of an Intel Xeon X5570 machine with 2.93 GHz with all CPLEX settings left on default values. The time limit was set to 3600 seconds and the memory limit to 9 GB.

We used two well-known instance sets in our computational study:

- pmed: This set contains 40 instances and was used for the pCP in e.g., [1, 3, 5, 6, 10]. For all instances, $\mathcal{I} = \mathcal{J} = \mathcal{V}$ holds, so all customer demand points are also potential facility locations. The number of $|\mathcal{V}|$ ranges between 100 and 900, and p is between 5 and 200. The instances in this set are given as graphs, and the distances d_{ij} are the shortest-path distances between $i, j \in \mathcal{V}$ in the graph.
- TSPlib: This is an instance set originally introduced in [18] for the traveling salesperson problem. Subsets of this instance set have been used in many works for the pCP, see, e.g., [3, 5, 6, 9, 10]. The number of potential facilities equals again the number of customer demand points, so $|\mathcal{V}| = |\mathcal{I}| = |\mathcal{J}|$. In the subset we consider for our study $|\mathcal{V}|$ ranges between 51 and 1002. The exact values of $|\mathcal{V}|$ can be found in the name of the instances in Table 2 as part of the instance name. The instances contain the two-dimensional coordinates for each point to calculate the Euclidean distance. Following previous literature, we rounded the distances to the nearest integer.

3.1 Comparison of formulations

In Table 1 we present the runtime (column t[s]), lower and upper bounds (columns lb and ub) and optimality gap (column g[%]) obtained for the pmed instances with $\mathcal{P} = \{p, p+1, p+2\}$, where p is given by the instance. If no optimal solution has been found within the time limit, it is indicated with the abbreviation TL in the runtime-column. The optimality gap is calculated as $\frac{ub-lb}{ub}100$ and the best obtained gap considering both formulations is indicated in bold. The formulation (nPC2) manages to solve eleven instances to optimality within the time limit, whereas formulation (nPC1) does not manage to solve any instance. Moreover, there is no instance where the obtained optimality gap of (nPC1) at the time limit is better than the time limit of (nPC2). We note that for formulation (nPC1) in 19 instances it was not possible to finish building the model for CPLEX to finish solving the initial LP relaxation within the given time limit. These issues in scalability between the two formulations can be explained by the fact, that the distances in the pmed-instance are based on shortest-path distances in a graph, thus there are not so many distinct distances, hence K is small, which is good for (nPC2).

For the instance set TSPlib, we used $\mathcal{P} = \{4, 5, 6\}$. The results (reported in Table 2) paint a similar overall picture, i.e., (nPC2) outperforms (nPC1) most of the time. With formulation (nPC2) 14 out of 50 instances of this set can be solved to optimality within the timelimit, whereas for formulation (nPC1) only 3 out of 50 instances can be solved to optimality. Regarding the optimality gap for instances not solved to optimality, the picture is not as clear as for instance set pmed. However, for nearly all instance for which (nPC1) had a lower gap than (nPC2), the lower bound of (nPC2) is better, but the upper bound is worse. This indicates that CPLEX primal heuristics seem to work better with (nPC1) while the theoretical strength of the LP relaxation of (nPC2) compared to (nPC1) seems to pay off also in practice. We note that for these instances there are much more distinct distances, and thus larger values of K compared to instance set pmed, as in this instance set the distances are based on the Euclidean distance. This can contribute to make them more difficult for formulation (nPC2) and is also reflected in the results, as for some instances of this set, the formulation does not manage to solve the root relaxation within the time limit (indicated by a value of zero in the column lb in the table).

Table 1: Comparison of (nPC1) and (nPC2) for the pmed instances with $\mathcal{P} = \{p, p+1, p+2\}$

			(nPC1)				(nPC2)			
Inst.	p	|V|	t[s]	lb	ub	g[%]	t[s]	lb	ub	g[%]
1	100	5	TL	267	381	29.95	1833.939	356	356	**0.00**
2	100	10	TL	196	345	43.14	313.387	292	292	**0.00**
3	100	10	TL	207	310	33.27	111.594	278	278	**0.00**
4	100	20	TL	139	235	40.85	85.634	220	220	**0.00**
5	100	33	TL	85	139	38.64	61.483	138	138	**0.00**
6	200	5	TL	178	276	35.50	TL	220	252	**12.70**
7	200	10	TL	134	201	33.42	TL	180	188	**4.05**
8	200	20	TL	100	180	44.31	1028.47	161	161	**0.00**
9	200	40	TL	59	488	87.84	1022.54	109	109	**0.00**
10	200	67	TL	29	63	53.23	416.797	58	58	**0.00**
11	300	5	TL	132	297	55.51	TL	160	175	**8.79**
12	300	10	TL	111	303	63.25	TL	140	154	**9.03**
13	300	30	TL	69	349	80.34	2964.903	107	107	**0.00**
14	300	60	TL	42	406	89.73	TL	67	78	**13.49**
15	300	100	TL	24	204	88.24	2756.437	52	52	**0.00**
16	400	5	TL	108	223	51.51	TL	131	140	**6.50**
17	400	10	TL	87	216	59.55	TL	106	117	**9.68**
18	400	40	TL	0	103239	100.00	TL	74	423	**82.44**
19	400	80	TL	30	256	88.34	TL	49	303	**83.92**
20	400	133	TL	18	106	83.04	3488.017	39	39	**0.00**
21	500	5	TL	0	92343	100.00	TL	107	130	**17.39**
22	500	10	TL	0	125994	100.00	TL	101	339	**70.13**
23	500	50	TL	0	95016	100.00	TL	58	282	**79.31**
24	500	100	TL	0	101730	100.00	TL	39	300	**86.89**
25	500	167	TL	14	228	93.64	TL	32	245	**86.98**
26	600	5	TL	0	106278	100.00	TL	101	113	**10.88**
27	600	10	TL	0	120651	100.00	TL	87	96	**9.30**
28	600	60	TL	0	148722	100.00	TL	48	330	**85.57**
29	600	120	TL	0	109215	100.00	TL	32	264	**87.74**
30	600	200	TL	13	220	94.13	TL	24	288	**91.82**
31	700	5	TL	0	96330	100.00	TL	82	195	**58.05**
32	700	10	TL	0	210198	100.00	TL	77	372	**79.25**
33	700	70	TL	0	106851	100.00	TL	41	222	**81.60**
34	700	140	TL	0	149175	100.00	TL	28	294	**90.52**
35	800	5	TL	0	119076	100.00	TL	82	91	**9.50**
36	800	10	TL	0	161628	100.00	TL	75	261	**71.44**
37	800	80	TL	0	139044	100.00	TL	40	234	**83.09**
38	900	5	TL	0	160359	100.00	TL	78	252	**69.16**
39	900	10	TL	0	262383	100.00	TL	62	345	**81.89**
40	900	90	TL	0	127518	100.00	TL	34	207	**83.61**

3.2 Results in context to the pCP

Next, we take a closer look at the performance of (nPC2) and also put the results obtained for the (nPC2) in context with results obtained for the pCP. In Table 3 we report the objective function values at termination (columns ub), the lower bound at the root node (columns root lb) and at termination (columns lb) for the n-pCP with $\mathcal{P} = \{p, p+1, p+2\}$ and also for the pCP with p. The pCP is solved using the formulation of [1], i.e., (nPC2) for $\mathcal{P} = \{p\}$. We denote this formulation by (PC). Table 4 reports the same values for TSPlib.

In Table 3 and Table 4, we can see that the nesting does not seem to have a huge effect on the obtained root lower bounds, as for both (PC)¸ and (nPC2) the root lower bounds are close to the lower bounds at termination and also close to the upper bounds for instances solved to optimality. Thus, also for the (nPC2) this

Table 2: Comparison of (nPC1) and (nPC2) for the TSPlib instances with $\mathcal{P} = \{4, 5, 6\}$ for (nPC2). The number in the instance names indicate the numbers of point in each instance.

	(nPC1)				(nPC2)			
Inst.	t[s]	lb	ub	g[%]	t[s]	lb	ub	g[%]
eil51	869.843	61	61	**0.00**	46.905	61	61	**0.00**
berlin52	2105.984	1215	1215	**0.00**	31.623	1215	1215	**0.00**
st70	2558.426	90	90	**0.00**	99.939	90	90	**0.00**
eil76	TL	50	70	29.20	409.317	64	64	**0.00**
pr76	TL	13060	16850	22.49	1416.761	16330	16330	**0.00**
rat99	TL	98	163	39.91	1399.774	144	144	**0.00**
kroA100	TL	1976	2973	33.54	2465.575	2812	2812	**0.00**
kroB100	TL	1967	3426	42.57	TL	2496	2965	**15.83**
kroC100	TL	1937	3199	39.45	TL	2402	2886	**16.76**
kroD100	TL	1842	3062	39.85	2863.816	2862	2862	**0.00**
kroE100	TL	1939	3097	37.38	2582.52	2893	2893	**0.00**
rd100	TL	669	1208	44.64	TL	911	993	**8.22**
eil101	TL	50	75	33.42	537.34	66	66	**0.00**
lin105	TL	1384	2352	41.14	2419.2	2067	2067	**0.00**
pr107	TL	3258	5238	37.80	1763.96	5170	5170	**0.00**
pr124	TL	5637	7682	26.62	2752.469	7370	7370	**0.00**
bier127	TL	12541	18279	31.39	2791.905	15936	15936	**0.00**
ch130	TL	487	776	37.25	TL	589	715	**17.63**
pr136	TL	7251	10674	32.07	TL	8591	9318	**7.80**
pr144	TL	6392	11866	46.13	TL	7774	12221	**36.39**
ch150	TL	441	873	**49.44**	TL	514	2547	79.81
kroA150	TL	1872	3523	**46.85**	TL	2198	12654	82.63
kroB150	TL	1881	3663	**48.66**	TL	2200	12561	82.49
pr152	TL	6831	15625	56.28	TL	8756	14493	**39.58**
u159	TL	3155	6593	52.14	TL	4073	5119	**20.44**
rat195	TL	132	280	**52.71**	TL	152	859	82.29
d198	TL	1013	4531	**77.65**	TL	1260	12780	90.14
kroA200	TL	1905	4092	**53.43**	TL	2215	12879	82.80
kroB200	TL	1862	3602	**48.32**	TL	2179	12510	82.58
ts225	TL	8548	41811	79.56	TL	9602	44826	**78.58**
tsp225	TL	227	543	**58.23**	TL	271	1557	82.58
pr226	TL	7733	14448	**46.47**	TL	9467	50796	81.36
gil262	TL	132	620	78.70	TL	152	703	**78.33**
pr264	TL	3104	24933	87.55	TL	3755	26069	**85.60**
a280	TL	140	726	**80.76**	TL	159	843	81.09
pr299	TL	2760	18220	**84.85**	TL	3136	20877	84.98
lin318	TL	2395	10980	**78.18**	TL	2774	14598	80.99
linhp318	TL	2395	10980	**78.18**	TL	2774	14598	80.99
rd400	TL	0	898956	100.00	TL	767	4059	**81.10**
fl417	TL	0	0	100.00	TL	1381	7029	**80.36**
pr439	TL	0	0	100.00	TL	7780	38460	**79.77**
pcb442	TL	0	0	100.00	TL	2530	14523	**82.58**
d493	TL	0	0	100.00	TL	1900	12888	**85.26**
u574	TL	0	0	**100.00**	TL	0	10314	**100.00**
rat575	TL	0	493497	**100.00**	TL	0	493497	**100.00**
p654	TL	0	0	**100.00**	TL	0	0	**100.00**
d657	TL	0	0	**100.00**	TL	0	0	**100.00**
u724	TL	0	0	**100.00**	TL	0	0	**100.00**
rat783	TL	0	782220	**100.00**	TL	0	782220	**100.00**
pr1002	TL	0	0	**100.00**	2014.429	0	0	**100.00**

modeling approach seems to give strong lower bounds. However, the heuristics of CPLEX seem to struggle to find good feasible solutions for larger instances of the n-pCP. This can be inferred from the fact that we can construct a valid upper bound solution for the n-pCP-instances by just putting the solution obtained for the pCP also as solution for $p+1$ and $p+2$ (together with one, resp., two random additional open facilities). The objective function value of solutions constructed in such a way for the n-pCP is at most three times the objective function value obtained the pCP. Thus, for example, for the instance *rat195* we would obtain a solution for n-pCP with value at most 216 while using (nPC2) we obtained an upper bound of 859 at termination. The situation is similar for many other instances.

Table 3: Root bound comparison on instance set pmed with $\mathcal{P} = \{p, p+1, p+2\}$. Optimal objective function values are printed in **bold**.

			(PC)			(nPC2)		
Inst.	|V|	p	ub	root lb	lb	ub	root lb	lb
1	100	5	**127**	118	127	**356**	306	356
2	100	10	**98**	98	98	**292**	243	292
3	100	10	**93**	93	93	**278**	237	278
4	100	20	**74**	74	74	**220**	188	220
5	100	33	**48**	43	48	**138**	104	138
6	200	5	**84**	79	84	252	214	220
7	200	10	**64**	61	64	188	167	180
8	200	20	**55**	51	55	**161**	140	161
9	200	40	**37**	36	37	**109**	92	109
10	200	67	**20**	25	20	**58**	47	58
11	300	5	**59**	59	59	175	154	160
12	300	10	**51**	52	51	154	139	140
13	300	30	**36**	32	36	**107**	95	107
14	300	60	**26**	31	26	78	65	67
15	300	100	**18**	14	18	**52**	41	52
16	400	5	**47**	46	47	140	128	131
17	400	10	**39**	36	39	117	105	106
18	400	40	**28**	25	28	423	74	74
19	400	80	**18**	15	18	303	46	49
20	400	133	**13**	11	13	**39**	31	39
21	500	5	**40**	38	40	130	107	107
22	500	10	**38**	35	38	339	101	101
23	500	50	**22**	20	22	282	58	58
24	500	100	**15**	13	15	300	38	39
25	500	167	**11**	9	11	245	26	32
26	600	5	**38**	36	38	113	100	101
27	600	10	**32**	30	32	96	87	87
28	600	60	**18**	16	18	330	48	48
29	600	120	**13**	11	13	264	32	32
30	600	200	**9**	7	9	288	22	24
31	700	5	**30**	31	30	195	82	82
32	700	10	**29**	27	29	372	77	77
33	700	70	**15**	14	15	222	41	41
34	700	140	**11**	9	11	294	28	28
35	800	5	**30**	29	30	91	82	82
36	800	10	28	25	27	261	74	75
37	800	80	59	13	15	234	40	40
38	900	5	**29**	29	29	252	77	78
39	900	10	**23**	21	23	345	62	62
40	900	90	**13**	11	13	207	34	34

4 CONCLUSION AND OUTLOOK

In this work, we introduced the nested p-center problem and presented two mixed-integer linear programming formulations for it, together with a computational study to evaluate the effectiveness

Table 4: Root bound comparison on instance set `TSPlib` with $\mathcal{P} = \{4, 5, 6\}$. `TSPlib`. Optimal objective function values are printed in **bold**.

	(PC)			(nPC2)		
Inst.	ub	root lb	lb	ub	root lb	lb
eil51	**22**	22	22	**61**	52	61
berlin52	**426**	426	426	**1215**	1100	1215
st70	**33**	33	33	**90**	78	90
eil76	**23**	23	23	**64**	52	64
pr76	**6082**	6082	6082	16330	12862	16329
rat99	**51**	46	51	**144**	109	144
kroA100	**1001**	832	1001	**2812**	2198	2812
kroB100	**989**	857	989	2965	2241	2496
kroC100	**977**	839	977	2886	2197	2402
kroD100	**995**	840	995	**2862**	2188	2862
kroE100	**1030**	826	1030	**2893**	2164	2893
rd100	**349**	305	349	993	794	911
eil101	**23**	92	23	**66**	53	66
lin105	**717**	615	717	**2067**	1619	2067
pr107	**1746**	1746	1746	**5170**	3730	5170
pr124	**2588**	2497	2588	**7370**	6592	7370
bier127	**5578**	5051	5578	**15936**	13004	15936
ch130	**237**	228	237	715	544	589
pr136	**3225**	2880	3225	9318	7742	8591
pr144	**3375**	2961	3375	12221	7758	7774
ch150	**225**	196	225	2547	512	514
kroA150	**1024**	822	1024	12654	2192	2198
kroB150	**1042**	830	1042	12561	2190	2200
pr152	**5100**	3732	5100	14493	8730	8756
u159	**1655**	1461	1655	5119	3754	4073
rat195	**72**	57	72	859	152	152
d198	**623**	541	623	12780	1255	1260
kroA200	1011	834	975	12879	2214	2215
kroB200	1008	826	835	12510	2178	2179
ts225	**4243**	3751	4243	44826	9601	9602
tsp225	**124**	104	124	1557	271	271
pr226	**4104**	3704	4104	50796	9438	9467
gil262	**66**	58	66	703	152	152
pr264	**1610**	1537	1610	26069	3754	3755
a280	**79**	60	79	843	159	159
pr299	6959	1192	1194	20877	3136	3136
lin318	4866	1065	1065	14598	2774	2774
linhp318	1331	1065	1065	14598	2774	2774
rd400	441	296	296	4059	767	767
fl417	676	536	537	7029	1381	1381
pr439	12820	12820	2958	38460	7780	7780
pcb442	4280	980	980	14523	2530	2530
d493	3957	749	749	12888	1900	1900
u574	3438	3438	692	10314	0	0
rat575	534	100	113	493497	0	0
p654	6083	1438	1444	7594660	0	0
d657	4771	4771	878	6044890	0	0
u724	3198	3198	614	3927170	0	0
rat783	628	628	117	782220	0	0
pr1002	0	0	0	0	0	0

of the proposed formulations. Based on the findings of the computational study, we currently work on the following topics to obtain an improved solution framework for the problem:

- design of starting and primal heuristics, as CPLEX seems to struggle to find good primal solutions on its own
- transferring the projection-based p-center formulation and constraint lifting ideas of [10] to the nested setting for better overall scaleability

Another interesting avenue for further research could be to combine the nested problem with some form of uncertainty.

ACKNOWLEDGMENTS

This research was funded in whole, or in part, by the Austrian Science Fund (FWF) [P 35160-N]. For the purpose of open access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

REFERENCES

[1] Zacharie Ales and Sourour Elloumi. 2018. Compact MILP Formulations for the p-Center Problem. *Combinatorial Optimization* (2018), 14–25.
[2] Ashwin Arulselvan, Andreas Bley, and Ivana Ljubić. 2019. The incremental connected facility location problem. *Computers & Operations Research* 112 (2019), 104763.
[3] Hatice Calik and Barbaros C Tansel. 2013. Double bound method for solving the p-center location problem. *Computers & Operations Research* 40, 12 (2013), 2991–2999.
[4] Tobia Calogiuri, Gianpaolo Ghiani, Emanuela Guerriero, and Emanuele Manni. 2021. The multi-period p-center problem with time-dependent travel times. *Computers & Operations Research* 136 (2021), 105487.
[5] Doron Chen and Reuven Chen. 2009. New relaxation-based algorithms for the optimal solution of the continuous and discrete p-center problems. *Computers & Operations Research* 36, 5 (2009), 1646–1655.
[6] Claudio Contardo, Manuel Iori, and Raphael Kramer. 2019. A scalable exact algorithm for the vertex p-center problem. *Computers & Operations Research* 103 (2019), 211–220.
[7] Isabel Correia and Teresa Melo. 2016. Multi-period capacitated facility location under delayed demand satisfaction. *European Journal of Operational Research* 255, 3 (2016), 729–746.
[8] Mark S. Daskin. 2013. *Network and Discrete Location: Models, Algorithms, and Applications, Second Edition.* John Wiley & Sons, Ltd.
[9] Sourour Elloumi, Martine Labbé, and Yves Pochet. 2004. A new formulation and resolution method for the p-center problem. *INFORMS Journal on Computing* 16, 1 (2004), 84–94.
[10] Elisabeth Gaar and Markus Sinnl. 2022. A scaleable projection-based branch-and-cut algorithm for the p-center problem. *European Journal of Operational Research* 303, 1 (2022), 78–98.
[11] Jesus Garcia-Diaz, Rolando Menchaca-Mendez, Ricardo Menchaca-Mendez, Saúl Pomares Hernández, Julio César Pérez-Sansalvador, and Noureddine Lakouari. 2019. Approximation algorithms for the vertex k-center problem: Survey and experimental evaluation. *IEEE Access* 7 (2019), 109228–109245.
[12] S Louis Hakimi. 1964. Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations Research* 12, 3 (1964), 450–459.
[13] Thomas Kalinowski, Dmytro Matsypura, and Martin W.P. Savelsbergh. 2015. Incremental network design with maximum flows. *European Journal of Operational Research* 242, 1 (2015), 51–62.
[14] Oded Kariv and S Louis Hakimi. 1979. An algorithmic approach to network location problems. I: The p-centers. *SIAM journal on applied mathematics* 37, 3 (1979), 513–538.
[15] Gilbert Laporte, Stefan Nickel, and Francisco Saldanha-da Gama. 2019. *Introduction to location science.* Springer.
[16] Ronald G McGarvey and Andreas Thorsen. 2022. Nested-solution facility location models. *Optimization letters* 16, 2 (2022), 497–514.
[17] Edward Minieka. 1970. The m-center problem. *Siam Review* 12, 1 (1970), 138–139.
[18] Gerhard Reinelt. 1991. TSPLIB—A traveling salesman problem library. *ORSA Journal on Computing* 3, 4 (1991), 376–384.
[19] Gary M Roodman and Leroy B Schwarz. 1975. Optimal and heuristic facility phase-out strategies. *AIIE transactions* 7, 2 (1975), 177–184.
[20] Allen J Scott. 1971. Dynamic location-allocation systems: some basic planning strategies. *Environment and planning A* 3, 1 (1971), 73–82.
[21] Lawrence V. Snyder. 2006. Facility location under uncertainty: a review. *IIE Transactions* 38, 7 (2006), 547–564.