Capacitated p-hub approach for park-and-ride facility location problem under nested logit demand function: polyhedral approaches

Sang Hyun Kim[1], Feng Qiu[2], Sangho Shim[3*]

[1]School of Air Transport, Transportation, and Logistics, Korea Aerospace University, Goyang-si, 10540, Gyeonggi-do, Republic of Korea.
[2]Energy Systems Division, Argonne National Laboratory, Lemont, 60439, Illinois, USA.
[3*]School of Engineering, Mathematics and Science, Robert Morris University, Moon Twp, 15108, Pennsylvania, USA.

*Corresponding author(s). E-mail(s): shim@rmu.edu;
Contributing authors: sanghyun@kau.ac.kr; fqiu@anl.gov;

Abstract

By generalizing the unconstrained $\boldsymbol{p}$-hub approach for the park-and-ride (P&R) facility location problem under the multinomial logit demand function, the capacitated $\boldsymbol{p}$-hub approach for the problem under the nested logit demand function captures a broader range of real-world cases. To solve this problem optimally, we introduce a mixed-integer linear program and accelerate it by enhancing the branch-and-cut procedure. To address the problem at a large scale, we introduce two other polyhedral approaches: variable neighborhood search (VNS) and adaptive randomized rounding (ARR). Downtown areas in Seoul have a high modal share of public transportation and congested road traffic, yet P&R has not been widely implemented. Therefore, we apply the ARR procedure to solve a real-world problem using traffic and geographic data from the Seoul metropolitan area. ARR performs better than VNS and addresses real-world cases. The solutions obtained by ARR present a phased expansion plan that encourages policymakers to start installing a small number of P&Rs immediately.

Keywords: choice model, capacitated p-hub approach, multinomial logit, nested logit, park-and-ride facility location problem

1 Introduction

To meet climate policy objectives and alleviate road congestion, park-and-ride (P&R) sites must be located in areas of maximal demand. This study developed a randomized rounding procedure and introduces three polyhedral approaches for locating P&R facilities that factor in their capacities, *that is*, both capacity and location are key decision-making parameters.

The study considered commuters who generally have two alternatives: driving directly to their destinations by private car or driving to a P&R site by private car and then using public transport to reach their destinations. Choosing to use a P&R site or drive directly to a destination is a decision that can be represented using a discrete-choice model. In addition to commuter preferences, sites have different capacities. If the commuter arrives at a full P&R site, we assume that the commuter then abandons using public transport and drives from the site to the destination.

A p-hub approach for the park-and-ride facility location problem (P&R FLP) selects p sites from among the location candidates while maximizing the expected number of commuters who use the selected P&R sites. This is the demand function. The capacity of the site also limits the demand function at the P&R site, and capacity constraints are added to the p-hub approach for the P&R FLP. Capacity constraints ensure that the expected number of users at a P&R site cannot exceed the capacity of the site and that arrivees at full sites continue driving on to their destinations.

Conventional P&R FLPs (*e.g.*, [1]) assume the independence of irrelevant alternatives (IIA), a restrictive property. These models only enforce the constraint of selecting p sites, while maximizing the multinomial logit (MNL) demand function. To capture a broader range of real-world scenarios and improve accuracy, we generalize the MNL demand function to the nested logit (NL) demand function, [15]. The available options can be divided into two modes, public transportation and private transportation, and the best model for these situations is the nested logit, in which the two modes are nests. To solve the capacitated p-hub approach for P&R FLP under the NL demand function, we introduce three polyhedral approaches:

- Mixed Integer Linear Programming (MILP)
- Variable Neighborhood Search (VNS)
- Adaptive Randomized Rounding (ARR)

To address large-scale P&R FLP in a large-scale instance, this study focuses on the ARR procedure. MILP is an exact solution method, which in medium-scale instances optimally solves the capacitated p-hub approach for P&R FLP under the NL demand function. The two heuristic algorithms, VNS and ARR, quickly reach optimal solutions for the same medium-scale instances. Computational experiments verified that the ARR was superior to VNS. We addressed a real-world problem in the Seoul metropolitan area by simultaneously running multiple iterations of the ARR procedure.

Section 2 reviews relevant literature. In Section 3, the unconstrained p-hub approach for P&R FLP under the MNL demand function is formulated as a mixed-integer linear program (MILP)[1], denoted as UMILP, which rewrites the formulations introduced by [1, 9]. By adding the capacity constraints of the locations, UMILP is reformulated as another MILP, denoted as CMILP. We then generalize the MNL demand function to an NL demand function. The outer approximation cuts [5, 20, 22] allow us to develop a MILP of the capacitated p-hub approach for P&R FLP under the NL demand function denoted by CMILP-NL. In Section 4, we generalize the constraints introduced by Krohn et al. [16] to bound the continuous variables of MNL choice probabilities and enhance the CMILP-NL by adding the proposed constraints bounding those of NL choice probabilities. We then randomize the VNS method into an ARR procedure to address a large-scale P&R FLP in a capacitated p-hub approach under an NL demand function. In Section 5, we present the numerical experiments performed on our proposed procedure, which verify that the ARR procedure rapidly reaches near-optimal solutions. Section 6 demonstrates the use of the ARR procedure for addressing real-world P&R FLP in the Seoul metropolitan area. Finally, in Section 7, we discuss our conclusions and plans for further research.

Figure 1 illustrates a generalization of the conventional p-hub approach for P&R FLP in two dimensions.

- Unconstrained → Capacitated
- MNL demand → NL demand

This study introduces three polyhedral approaches: CMILP-NL, VNS, and ARR. Throughout this paper, we adopt the informative notation used by Krohn et al. [16] and Train [35].

2 Literature review

Benati [2] stated that the unconstrained p-hub approach under an MNL demand function is NP-hard. Subsequently, Haase [9] linearized the p-hub approach for a facility location problem. After omitting the restrictions, Haase and Müller [10] proposed three reformulations of the unconstrained MNL models by Benati and Hansen [3], Haase [9], and Zhang et al. [41]. Aros-Vera et al. [1] introduced an MILP for the unconstrained p-hub approach for P&R FLP, which our UMILP rewrites. Kitthamkesorn et al. [13] introduced an MILP for P&R FLP by incorporating a paired combinatorial weibit model.

Additional constraints must be added to capture real-world problems. In particular, the capacity constraint is a crucial factor in tactical problem-solving, ensuring that demand is optimally satisfied. In the park-and-ride facility location problem, the parking site's capacity is the number of spaces in the parking lot. Employing auxiliary variables, we added capacity constraints to ensure different capacities for different sites. We integrated a discrete choice model of the P&R FLP with capacity constraints. Lamontagne et. al. [19] introduced a discrete choice model for electric

[1] MILP stands for mixed-integer linear programming and mixed-integer linear program (*i.e.*, mixed-integer linear programming formulation) interchangeably

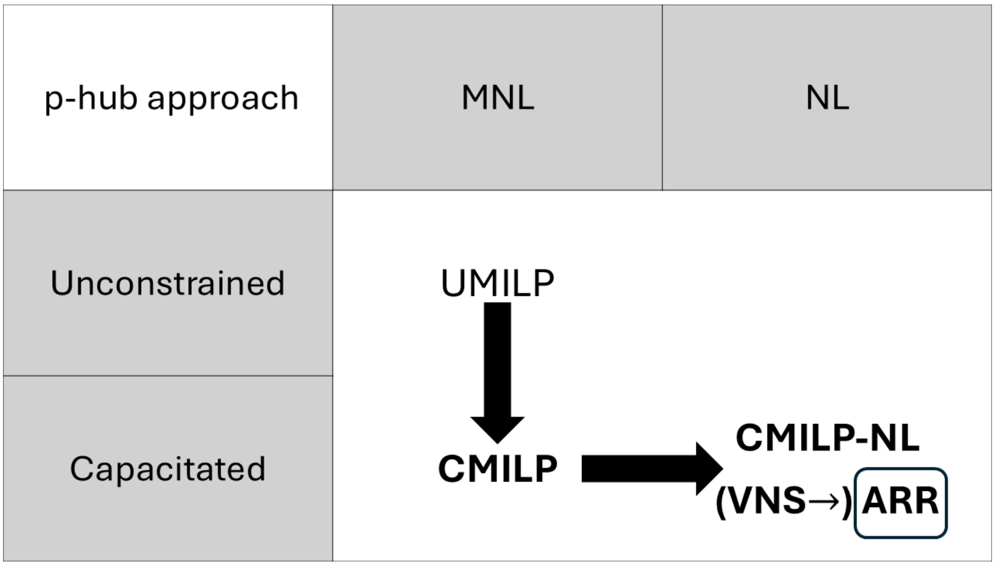

Fig. 1 Flow of this paper

vehicle charging station placement without considering the station's capacity. Without integrating the discrete choice model, Parent, Carvalho, Anjos, and Atallah [27] considered the station's capacity omitted from Lamontagne et al. [19]. The discrete choice model [19] simulates the user's choices and utility maximization behavior to avoid the nonlinearities of the model, as described by Paneque et.al. [28].

To capture real-world problems more closely, we generalize the MNL demand function to a nested logit (NL) demand function because NL models have fewer restrictive assumptions and better fit actual customer choice data than MNL models [15]. The concavity of demand function provides efficient and exact methods for the unconstrained p-hub approach under an NL demand function. Ljubíc and Moreno [20] derived an outer-approximation-based linear formulation of an MNL model and proposed a Branch and Cut (B&C) algorithm that adds outer-approximation cuts when an integer solution is found during the solving process. Méndez-Vogel, Marianov and Lüer-Villagra [22] generalized the B&C algorithm to exact methods for an NL model. In exact solution methods for the unconstrained p-hub approach, the objective function (demand function) maximizes the continuous variables that indicate the choice probabilities. Dam, Ta, and Mai [5] solved the problem of generalized extreme-value models, including NL models, using heuristics. Mai and Lodi [21] solved large-scale problems by using the B&C approach.

To solve the capacitated p-hub approach for the P&R FLP under the NL demand function, we randomize a VNS method into ARR. The VNS method was developed by Mladenovi and Hansen [23]. The VNS method entails a systematic change in the neighborhood within a local search algorithm that performs a sequence of local changes in an initial solution, by which the value of the objective function is improved each time until a local optimum is found. In polyhedral geometry, the ARR procedure finds a better solution in the approximate neighborhood of the current best solution.

By relaxing the integrality of some integer variables, traditional randomized rounding first solves the relaxation of the formulation and determines integer solutions with the probability distribution given by the fractional optimal solution, where the fractional value of a $\{0,1\}$-variable is the probability that the variable takes on 1. For details on traditional randomized rounding, the readers may refer to Motwani and Raghavan [24] and Vazirani [39]. Without solving the relaxation of a formulation, our ARR procedure starts with a uniform distribution (*i.e.*, the $\{0,1\}$-variables take on 1 with equal probabilities), and it adaptively changes the fractional values to the incumbent solution, finding the integer solutions near the incumbent solution. Our randomized rounding also relaxes the principle of probability distribution for integer solutions: The fractional value of a $\{0,1\}$-variable is not the precise probability at which the variable takes 1, but a higher value implies a greater chance of the variable taking 1.

The full procedure of ARR is similar to the framework of the famous greedy randomized adaptive search procedure (GRASP). In the literature on facility location, GRASP has proven useful for adding locations with the highest potential for increasing market share. Resende [30] described a GRASP for the maximum covering problem (MCP) that finds approximate facility placement configurations. GRASP (Feo and Resende [7]) was applied to similar problems [6, 14, 29].

3 Problem formulations

Each commuter is associated with a pair comprising an origin and destination, denoted by (O, D). A segment is a set of commuters having the same origin and destination pairs. An *individual* $i \in \mathcal{I}$ is a (representative) commuter in a segment or in the segment itself, and $|\mathcal{I}|$ segments exist. We identify individuals $i = (O, D) \in \mathcal{I}$ with their origin-destination pairs. Individuals are decision-makers who decide either to drive themselves, denoted by alternative $j = 0$, or use a park-and-ride facility, denoted by alternative $j \in \mathcal{J}$, where $\mathcal{J}$ denotes the set of all candidate locations for P&R facilities.

A p-hub approach for the P&R FLP selects the optimal p P&R facility locations from among the candidates, thus maximizing the expected number of P&R users. Section 3.1 introduces a nonlinear program for the unconstrained p-hub approach for P&R FLP under the MNL demand function. The restrictive condition of the MNL model, named the IIA, allows us to linearize the nonlinear program into MILP which can be optimally solved using a MILP solver. Then, we add a capacity constraint to the unconstrained p-hub approach. Section 3.2 generalizes the MNL demand function to an NL demand function.

3.1 MNL demand function

This section introduces the mixed-integer non-linear and linear programs (MINLPs and MILPs) of the unconstrained and capacitated p-hub approaches for the P&R

FLP. The mathematical program of the choice model employs continuous variables ($P_{ij} \in [0,1] : i \in \mathcal{I}, j \in \mathcal{J}$) indicating choice probabilities, and binary variables ($Y_j \in \{0,1\} : j \in \mathcal{J}$) indicating installed P&Rs. For individual $i \in \mathcal{I}$ and potential P&R $j \in \mathcal{J}$, the choice probability P_{ij} is the probability that individual i chooses P&R j. In particular, $P_{ij} = 0$ for all $i \in \mathcal{I}$ if $Y_j = 0$ (*i.e.*, if j is not installed). Let w_i be the number of commuters with the same origin and destination pair as individual $i \in \mathcal{I}$. The P&R FLP maximizes the expected number of P&R users:

$$\sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} w_i P_{ij}. \tag{1}$$

The p-hub approach selects p facilities from $J = |\mathcal{J}|$ candidate locations. Each facility j to be established is indicated by the binary variable $Y_j \in \{0,1\}$ satisfying

$$\sum_{j\in\mathcal{J}} Y_j = p. \tag{2}$$

3.1.1 Nonlinear program under the MNL demand function

The base case for individual i, denoted by alternative $j = 0$, drives a car with utility V_{i0}. The odds of individual i selecting alternative $j \in \mathcal{J}$ are

$$\text{odds}(i,j) = \frac{p_{ij}}{1 - p_{ij}} \,,$$

where p_{ij} is the probability of individual i choosing alternative $j \in \mathcal{J}$ when j is the only available alternative, *that is*, $j \in \mathcal{J}' = \{j\}$. The independence of irrelevant alternatives (IIA) ensures that the ratio the probability that the two alternatives are chosen is independent of the existence or characteristics of the other alternatives. The IIA property implies that the odds of an alternative are not changed by other alternatives and are represented by the ratio of the preference weights of $j \in \mathcal{J}$ to the base case $j = 0$: *that is*,

$$\text{odds}(i,j) = \frac{e^{V_{ij}}}{e^{V_{i0}}} \,.$$

For any subset $\mathcal{J}' \subseteq \mathcal{J}$ of $p = |\mathcal{J}'|$ that establishes P&Rs, the IIA property ensures that

$$\frac{P_{ij}}{P_{i0}} = \text{odds}(i,j) = \frac{e^{V_{ij}}}{e^{V_{i0}}} \text{ for } i \in \mathcal{I}, j \in \mathcal{J}'.$$

Equivalently,

$$\frac{P_{ij}}{P_{i0}} = \frac{e^{V_{ij}} Y_j}{e^{V_{i0}}} \text{ for } i \in \mathcal{I}, j \in \mathcal{J}, \tag{3}$$

where $Y_j = 1$ indicates $j \in \mathcal{J}'$. This implies that

$$P_{ij} = \frac{e^{V_{ij}} Y_j}{e^{V_{i0}} + \sum_{k\in\mathcal{J}} e^{V_{ik}} Y_k} \text{ for } i \in \mathcal{I}, j \in \mathcal{J}, \tag{4}$$

which is equivalent to

$$P_{ij} = \frac{\text{odds}(i,j)Y_j}{1+\sum_{k\in\mathcal{J}}\text{odds}(i,k)Y_k} \text{ for } i \in \mathcal{I}, j \in \mathcal{J}.$$

With non-negative continuous variables P_{ij} for $i \in \mathcal{I}$ and $j \in \{0\} \cup \mathcal{J}$ and binary variables Y_j for $j \in \mathcal{J}$, a nonlinear program of the unconstrained p-hub approach for the P&R FLP is

$$\max\{(1) : (2) \text{ and } (4)\}. \tag{5}$$

3.1.2 MILPs under the MNL demand function

This section describes integrating the MILP introduced by Haase [9] and Aros-Vera et al. [1] with capacity constraints. The sum of the choice probabilities for each individual i is:

$$P_{i0} + \sum_{j\in\mathcal{J}} P_{ij} = 1 \text{ for } i \in \mathcal{I}. \tag{6}$$

If j is not established, no individual can use P&R at location j, *that is*,

$$P_{ij} \leq Y_j \text{ for } i \in \mathcal{I}, j \in \mathcal{J}. \tag{7}$$

Along with (7), the IIA properties (3) are ensured by the linear constraints

$$P_{ij} \leq \frac{e^{V_{ij}}}{e^{V_{i0}}} P_{i0} \text{ for } i \in \mathcal{I}, j \in \mathcal{J} \tag{8}$$

$$P_{i0} \leq \frac{e^{V_{i0}}}{e^{V_{ij}}} P_{ij} + (1 - Y_j) \text{ for } i \in \mathcal{I}, j \in \mathcal{J}. \tag{9}$$

With non-negative continuous variables P_{ij} for $i \in \mathcal{I}$ and $j \in \{0\} \cup \mathcal{J}$ and binary variables Y_j for $j \in \mathcal{J}$, the MILP under the MNL demand function is

$$\text{(UMILP)} \quad \max\{(1) : (2) \text{ and } (6)\text{-}(9)\}. \tag{10}$$

For the unconstrained p-hub approach for the P&R FLP under the MNL demand function, constraint (9) in MILP (10) is redundant.

Proposition 1. *For the unconstrained p-hub approach for the P&R FLP, the formulation defined by constraints (2) and (6)-(8) is valid, even without (9).*

Proof. Given an integer solution $Y^{\text{INT}} = \left(Y_j^{\text{INT}} \in \{0,1\} : j \in \mathcal{J}\right)$ that satisfies $\sum_{j\in\mathcal{J}} Y_j^{\text{INT}} = p$, the unconstrained P&R FLP is reduced to a subproblem

$$\max \sum_{i\in\mathcal{I}} w_i \sum_{j\in\mathcal{J}} P_{ij} \tag{11}$$

$$\text{subject to} \quad P_{i0} + \sum_{j \in \mathcal{J}} P_{ij} = 1 \text{ for } i \in \mathcal{I} \tag{12}$$

$$P_{ij} \leq Y_j^{\mathrm{INT}} \text{ for } i \in \mathcal{I}, j \in \mathcal{J} \tag{13}$$

$$P_{ij} \leq \frac{e^{V_{ij}}}{e^{V_{i0}}} P_{i0} \text{ for } i \in \mathcal{I},\ j \in \mathcal{J}, \tag{14}$$

where $P = (P_{ij} : i \in \mathcal{I}, j \in \{0\} \cup \mathcal{J})$ are nonnegative continuous variables. For every integer solution $Y^{\mathrm{INT}} = \left(Y_j^{\mathrm{INT}} \in \{0,1\} : j \in \mathcal{J}\right)$ satisfying $\sum_{j \in \mathcal{J}} Y_j^{\mathrm{INT}} = p$, we show that P_{ij} are the choice probabilities in the optimal solution to the subproblem.

Let $\mathcal{J}^{\mathrm{INT}} \subseteq \mathcal{J}$ be a subset of the established facilities $j \in \mathcal{J}$ indicated by $Y_j^{\mathrm{INT}} = 1$. If P_{i0}variables are substituted into (12), the reduced formulation can be rewritten as follows:

$$\max \quad \sum_{i \in \mathcal{I}} w_i \sum_{j \in \mathcal{J}^{\mathrm{INT}}} P_{ij} \tag{15}$$

$$\text{subject to} \quad P_{i0} + \sum_{j \in \mathcal{J}^{\mathrm{INT}}} P_{ij} = 1 \text{ for } i \in \mathcal{I} \tag{16}$$

$$P_{ij} = 0 \text{ for } i \in \mathcal{I}, j \notin \mathcal{J}^{\mathrm{INT}} \tag{17}$$

$$P_{ij} \leq \frac{e^{V_{ij}}}{e^{V_{i0}}} P_{i0} \text{ for } i \in \mathcal{I},\ j \in \mathcal{J}^{\mathrm{INT}} \tag{18}$$

The sum of Constraints (18) over $j \in \mathcal{J}^{\mathrm{INT}}$ implies

$$\sum_{j \in \mathcal{J}^{\mathrm{INT}}} P_{ij} \leq \left(\frac{\sum_{j \in \mathcal{J}^{\mathrm{INT}}} e^{V_{ij}}}{e^{V_{i0}}} \right) \left(1 - \sum_{j \in \mathcal{J}^{\mathrm{INT}}} P_{ij} \right).$$

Equivalently,

$$\sum_{j \in \mathcal{J}^{\mathrm{INT}}} P_{ij} \leq \left(\frac{\sum_{j \in \mathcal{J}^{\mathrm{INT}}} e^{V_{ij}}}{e^{V_{i0}} + \sum_{j \in \mathcal{J}^{\mathrm{INT}}} e^{V_{ij}}} \right). \tag{19}$$

As the left-hand side is maximized in (15) for every $i \in \mathcal{I}$, (19) is satisfied as an equality constraint for every $i \in \mathcal{I}$. For $i \in \mathcal{I}$,

$$P_{i0} = 1 - \sum_{j \in \mathcal{J}^{\mathrm{INT}}} P_{ij} = 1 - \left(\frac{\sum_{j \in \mathcal{J}^{\mathrm{INT}}} e^{V_{ij}}}{e^{V_{i0}} + \sum_{j \in \mathcal{J}^{\mathrm{INT}}} e^{V_{ij}}} \right) = \frac{e^{V_{i0}}}{e^{V_{i0}} + \sum_{j \in \mathcal{J}^{\mathrm{INT}}} e^{V_{ij}}},$$

and constraints (18) are equivalent to

$$P_{ij} \leq \frac{e^{V_{ij}}}{e^{V_{i0}} + \sum_{k \in \mathcal{J}^{\mathrm{INT}}} e^{V_{ik}}} \text{ for } j \in \mathcal{J}^{\mathrm{INT}}. \tag{20}$$

The sum of (20) over $j \in \mathcal{J}^{\text{INT}}$ is given by (19). As (19) is satisfied as an equality, (20) is satisfied as an equality constraint (4) with $Y = Y^{\text{INT}}$, thus completing the proof of the theorem. □

In practice, sites exhibit varying capacities. Let C_j be the capacity of P&R $j \in \mathcal{J}$. We employ the additional variable W_j which denotes the expected number of commuters who use P&R $j \in \mathcal{J}$. Then, the capacity constraint is

$$W_j = \min\left\{\sum_{i\in\mathcal{I}} w_i P_{ij}, C_j\right\} \text{ for } j \in \mathcal{J}. \tag{21}$$

In addition to the objective function $\max \sum_{j\in\mathcal{J}} W_j$, the MILP adds the following two constraints:

$$W_j \le \sum_{i\in\mathcal{I}} w_i P_{ij} \text{ for } j \in \mathcal{J}, \tag{22}$$

$$W_j \le C_j \text{ for } j \in \mathcal{J}. \tag{23}$$

The MILP of the capacitated p-hub approach is:

$$\text{(CMILP)} \quad \max\left\{\sum_{j\in\mathcal{J}} W_j : (2), (6)\text{-}(9) \text{ and } (22)\text{-}(23)\right\}. \tag{24}$$

The constraint (9) in MILP (24) of the capacitated p-hub approach is necessary. This contrasts with the unconstrained p-hub approach (10), where the same constraint is redundant (Theorem 1).

3.2 Nested logit demand function

3.2.1 Non-linear program under the NL demand function

MNL is a special type of NL demand function. The NL model partitions the alternatives into multiple nests. In this study, we consider two nests $\mathcal{N}$: a nest for driving a car (*i.e.*, $\mathcal{N} = \{j = 0\}$) and another nest for P&Rs (*i.e.*, $\mathcal{N} = \mathcal{J}$). In the NL model, the probability of selecting an alternative $j \in \{0\} \cup \mathcal{J}$ is expressed as the product of the probability of selecting the related nest $\mathcal{N}$ and the conditional probability of selecting alternative $j \in \mathcal{N}$ in the nest.

$$P[i \text{ selects } j] = P[i \text{ selects } \mathcal{N}] \times P[i \text{ selects } j | i \text{ selects } \mathcal{N}].$$

They are defined by

$$P_{i0} = \frac{e^{V_{i0}}}{e^{V_{i0}} + e^{\lambda\Gamma_i}}, \tag{25}$$

$$P_{ij} = \frac{e^{\lambda \Gamma_i}}{e^{V_{i0}} + e^{\lambda \Gamma_i}} \frac{e^{V_{ij}/\lambda} Y_j}{\sum_{k \in \mathcal{J}} e^{V_{ik}/\lambda} Y_k}, \tag{26}$$

where $\lambda > 0$ is the nest coefficient related to $\mathcal{N} = \mathcal{J}$ and

$$\Gamma_i = \ln \sum_{j \in \mathcal{J}} e^{V_{ij}/\lambda} Y_j \text{ for } i \in \mathcal{I}. \tag{27}$$

Williams and Ortúzar [40] have shown that if λ is not between zero and one, then the model may provide unacceptable representations of behavior (see [11]), and we assume $0 < \lambda \leq 1$. For the generalization of λ, readers may refer to Davis, Gallego, and Topaloglu [4].

The formulation of the unconstrained p-hub approach for the P&R FLP under the NL demand function is

$$\max\{(1) : (2) \text{ and } (26)\text{-}(27)\}. \tag{28}$$

Note that Γ_i are functions $\Gamma_i(Y)$ of $Y = (Y_j : j \in \mathcal{J})$ (Equation 27), and P_{ij} are also functions $P_{ij}(Y)$ of Y (Equation 26) for $i \in \mathcal{I}$ and $j \in \mathcal{J}$. Thus, objective function (1) is a function of Y, denoted by $\text{DEMAND}_\lambda(Y)$. The formulation of the unconstrained p-hub approach can then be expressed as:

$$\max\left\{\text{DEMAND}_\lambda(Y) : \sum_{j \in \mathcal{J}} Y_j = p\right\}. \tag{29}$$

Note that the NL model with $\lambda = 1$ is also the MNL model.

However, the capacitated p-hub approach for P&R FLP under an NL demand function is formulated as:

$$\max\left\{\sum_{j \in \mathcal{J}} W_j : (2), (21) \text{ and } (26)\text{-}(27)\right\}. \tag{30}$$

The choice probabilities $P_{ij} = P_{ij}(Y)$ are functions of $Y \in \{0,1\}^{\mathcal{J}}$ defined by (26)–(27), whereas W_j are also functions of Y for $j \in \mathcal{J}$ defined by (21); *i.e.*,

$$W_j = W_j(Y) = \min\left\{\sum_{i \in \mathcal{I}} w_i P_{ij}(Y), C_j\right\} \text{ for } j \in \mathcal{J}.$$

Thus, the objective function is a function of Y and is denoted as

$$\text{DEMAND}_\lambda^C(Y) = \sum_{j \in \mathcal{J}} W_j(Y). \tag{31}$$

The mixed-integer nonlinear program (30) of the capacitated p-hub approach can then be expressed as

$$\text{(CMINLP-NL)} \quad \max\left\{\text{DEMAND}_{\lambda}^{C}(Y) : \sum_{j\in\mathcal{J}} Y_j = p\right\}. \tag{32}$$

If $\lambda = 1$, then (26) is the MNL choice probability (4).

3.2.2 MILP under the NL demand function

Next, we linearize the nonlinear model using the NL demand function. Let $0 < \lambda < 1$ and let $\hat{w} : \mathbb{R}_{+} \to \mathbb{R}$ be a differentiable function defined by

$$\hat{w}(z) = \frac{z^{\lambda}}{1+z^{\lambda}}. \tag{33}$$

Then, its derivative is

$$\frac{d}{dz}\hat{w}(z) = \frac{\lambda z^{\lambda-1}}{\left(1+z^{\lambda}\right)^{2}}. \tag{34}$$

Clearly, $\hat{w}$ is an increasing concave function. For $i \in \mathcal{I}$, we define

$$z_i(Y) = \sum_{j\in\mathcal{J}} e^{V_{ij}/\lambda} Y_j. \tag{35}$$

Let $\mathrm{INT}(p)$ denote the set of integer solutions Y^{INT} satisfying $\sum_{j\in\mathcal{J}} Y_j^{\mathrm{INT}} = p$, which are integer solutions for the unconstrained p-hub approach. Because $\hat{w}$ is a concave function, the value of $\hat{w}(z_i(Y))$ is bounded from above by its first-order approximation of $Y^{\mathrm{INT}} \in \mathrm{INT}(p) \subset \{0,1\}^{\mathcal{J}}$; *i.e.*,

$$\begin{aligned}
&\sum_{j\in\mathcal{J}} P_{ij} = \hat{w}(z_i(Y)) \\
&\leq \hat{w}\left(z_i\left(Y^{\mathrm{INT}}\right)\right) + \sum_{j\in\mathcal{J}} \frac{\partial \hat{w}}{\partial Y_j}\left(z_i\left(Y^{\mathrm{INT}}\right)\right)\cdot\left(Y_j - Y_j^{\mathrm{INT}}\right) \\
&= \hat{w}\left(z_i\left(Y^{\mathrm{INT}}\right)\right) + \sum_{j\in\mathcal{J}} \hat{w}'\left(z_i\left(Y^{\mathrm{INT}}\right)\right)\cdot e^{V_{ij}/\lambda}\cdot\left(Y_j - Y_j^{\mathrm{INT}}\right)
\end{aligned}$$

for $i \in \mathcal{I}$ and $Y^{\mathrm{INT}} \in \mathrm{INT}(p)$. Similarly, the valid inequality can be rewritten as

$$\begin{aligned}
&\sum_{j\in\mathcal{J}} P_{ij} - \sum_{j\in\mathcal{J}} \hat{w}'\left(z_i\left(Y^{\mathrm{INT}}\right)\right)\cdot e^{V_{ij}/\lambda}\cdot Y_j \\
&\leq \hat{w}\left(z_i\left(Y^{\mathrm{INT}}\right)\right) - \sum_{j\in\mathcal{J}} \hat{w}'\left(z_i\left(Y^{\mathrm{INT}}\right)\right)\cdot e^{V_{ij}/\lambda}\cdot Y_j^{\mathrm{INT}},
\end{aligned} \tag{36}$$

which we refer to as the *outer approximation cuts.*

For $i \in \mathcal{I}$ and $Y^{\text{INT}} \in \text{INT}(p)$, The lower bound is given by

$$\sum_{j\in\mathcal{J}} P_{ij} \geq \hat{w}\left(z_i\left(Y^{\text{INT}}\right)\right) - \sum_{Y_j^{\text{INT}}=1} (1 - Y_j) - \sum_{Y_j^{\text{INT}}=0} Y_j. \tag{37}$$

Whenever $Y = (Y_j : j \in \mathcal{J}) \neq Y^{\text{INT}} \in \text{INT}(p)$, the inequality (37) is redundant because $\hat{w}(z) < 1$ and the right-hand side of (37) is negative. Along with the outer approximation cuts (36), the lower bounding constraints (37) ensure that the total probability of accessing a facility satisfies $\sum_{j\in\mathcal{J}} P_{ij} = 1 - P_{i0}$ or matches the base case probability P_{i0} in (25). Subsequently, (26) is ensured by

$$P_{ij} \leq \frac{e^{V_{ij}/\lambda}}{e^{V_{ik}/\lambda}} P_{ik} + (1 - Y_k) \tag{38}$$

for $i \in \mathcal{I}$ and $j \neq k \in \mathcal{J}$ across alternatives.

In (24), a MILP under the NL demand replaces (8)-(9) with (36)-(38).

$$\text{(CMILP-NL)} \quad \max \left\{ \sum_{j\in\mathcal{J}} W_j : (2),\ (6)\text{-}(7),\ (22)\text{-}(23) \text{ and } (36)\text{-}(38) \right\}. \tag{39}$$

A MILP under the NL demand has exponentially many constraints (36)-(37) because $|\text{INT}(p)|$ is exponentially large. Given $Y^{\text{INT}} \in \text{INT}(p)$, the number of constraints (36)-(37) is a polynomial of $|\mathcal{I}|$ and $|\mathcal{J}|$. A branch-and-cut procedure may efficiently solve a MILP of this type.

4 Three Polyhedral Approaches

We introduce three polyhedral approaches to address the capacitated p-hub problem under the NL demand function. First, we enhance the proposed MILP and optimally solve the problem for medium-scale instances. To address a large-scale instance, we introduce two heuristic algorithms: VNS and ARR.

VNS and ARR find good integer solutions $Y^{\text{INT}} = \left(Y_j^{\text{INT}} \in \{0, 1\} : j \in \mathcal{J}\right)$ satisfying $\sum_{j\in\mathcal{J}} Y_j^{\text{INT}} = p$ in the Euclidean space $\mathbb{R}^{\mathcal{J}}$. Let $\text{INT}(p)$ denote the set of integer solutions Y^{INT} satisfying $\sum_{j\in\mathcal{J}} Y_j^{\text{INT}} = p$, which are integer solutions for the unconstrained p-hub approach. The relaxation of $\text{INT}(p)$ is given by

$$\text{REL}(p) = \left\{ \tilde{Y} \in [0, 1]^{\mathcal{J}} : \sum_{j\in\mathcal{J}} \tilde{Y}_j = p \right\},$$

where $[0, 1]$ denotes the interval $\left\{\tilde{Y}_j \in \mathbb{R} : 0 \leq \tilde{Y}_j \leq 1\right\}$ between 0 and 1 for each $j \in \mathcal{J}$.

This indicates that $\text{INT}(p)$ is the set of vertices of the polyhedron $\text{REL}(p)$. It can also be observed that the two vertices $Y^{\text{INT}(1)}$ and $Y^{\text{INT}(2)}$ are adjacent (*i.e.*, connected

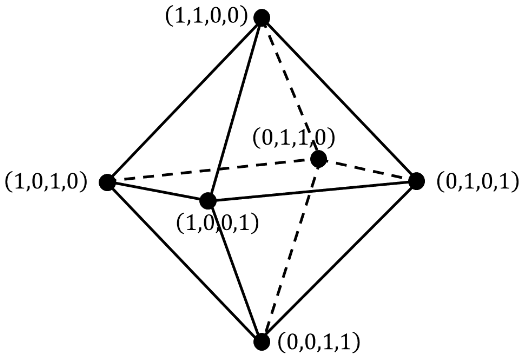

Fig. 2 Adjacency graph of REL(p) selecting $p = 2$ locations among $|\mathcal{J}| = 4$ candidates

by an edge of the polyhedron REL(p)) if their 1-norm distance is 2. When the two vertices $Y^{\text{INT}(1)}$ and $Y^{\text{INT}(2)}$ are adjacent, they are said to be *neighbors*. Each vertex has $p \times (|\mathcal{J}| - p)$ neighbors, where p is the number of 1s and $|\mathcal{J}| - p$ is the number of zero components of the vector of the vertex. For example, Figure 2 shows an adjacency graph of the vertices and edges of the smallest polyhedron REL(2) containing integer solutions INT(2) in the Euclidean space $\mathbb{R}^{\mathcal{J}}$ with $|\mathcal{J}| = 4$. The integer solutions are the vertices of polyhedron REL(2).

To develop ARR, which was the focus of this study, we randomize VNS. In Section 5, ARR was verified to be superior to VNS. In Section 6, ARR tackles a real-world case.

4.1 Mixed-integer linear programming

In this section, we enhance CMILP-NL (39) by imposing additional constraints on the continuous variables of the choice probabilities, further tightening (7). Given an individual and alternative $(i, j) \in \mathcal{I} \times \mathcal{J}$, we define

$$z_i^{\max(j)} = \max\left\{ z_i(Y) : Y_j = 1 \text{ and } \sum_{k \in \mathcal{J}} Y_k = p \right\} \tag{40}$$

$$z_i^{\min(j)} = \min\left\{ z_i(Y) : Y_j = 1 \text{ and } \sum_{k \in \mathcal{J}} Y_k = p \right\} \tag{41}$$

where $z_i(Y) = \sum_{k \in \mathcal{J}} e^{V_{ik}/\lambda} Y_k$ is defined by (35). The values can be easily computed by sorting the utilities $(V_{ik} : k \in \mathcal{J})$ for each individual $i \in \mathcal{I}$ in decreasing and increasing order, respectively.

The choice probabilities are determined by the installed P&R sites, as indicated by Y. We rewrite Equation (26) as

$$P_{ij} = \hat{w}\left(z_i(Y)\right) \cdot \frac{e^{V_{ij}/\lambda}}{z_i(Y)} \cdot Y_j.$$

As $\hat{w}(z)$ defined in (33) is an increasing function, the choice probabilities are bounded as follows:

$$\hat{w}\left(z_i^{\min(j)}\right) \cdot \frac{e^{V_{ij}/\lambda}}{z_i^{\max(j)}} \cdot Y_j \leq P_{ij} \leq \hat{w}\left(z_i^{\max(j)}\right) \cdot \frac{e^{V_{ij}/\lambda}}{z_i^{\min(j)}} \cdot Y_j \tag{42}$$

where $z_i^{\max(j)}$ and $z_i^{\min(j)}$ are defined by (40) and (41), respectively. The bounding constraints tighten constraints (7). The CMILP-NL strengthened by the bounding constraints (42) is denoted as Enhanced(λ).

Table 1 Important Cutting Planes

`Cutting Planes:`
`Implied bound`
`MIR`
`Flow cover`
`RLT`
`Relax-and-lift`

Adding only the important cutting planes also accelerates the Enhanced(λ). Table 1 lists the important cutting planes found in the preliminary experiments on the small instances. In particular, the integer variables of the Enhanced(λ) are all binary, and all cutting planes are theoretically mixed integer rounding (MIR) cuts (See Nemhauser and Wolsey [25]). The other cuts are added by their own algorithms, but theoretically, they are all MIR cuts, and MILP solvers tend to frequently add MIR cuts to the MILPs with all binary integer variables.

The two proposed enhancements, by adding the bounding constraints (42) and by adding only the important cutting planes (Table 1), were verified by the computational experiments in Section 5 (See Table 3).

4.2 Variable neighborhood search

In this section, we introduce a VNS method, which was first developed by Mladenović and Hansen [23], to solve the capacitated p-hub approach (32) by considering the polyhedron whose vertices are the integer solutions $Y^{\text{INT}} \in \{0,1\}^{\mathcal{J}}$ satisfying $\sum_{j\in\mathcal{J}} Y_j^{\text{INT}} = p$. We do not consider the entire polyhedron but rather the adjacency graph defined by the vertices and edges of the polyhedron. Similar to the simplex method, our VNS method starts at one vertex of the polyhedron and moves along an edge to the next vertex (*i.e.*, its neighbor), thereby improving the capacitated NL demand function $\text{DEMAND}_\lambda^C(Y)$. This method iterates until there are no better neighbors at a local optimum. On the adjacency graph $G = (V, E)$ of polyhedron REL(p), where V is the set of vertices and E is the set of edges of the polyhedron, a *local optimum* Y^{LOPT} is defined to have only inferior neighbors $Y^{\text{NBR}(k)}$, *i.e.*,

$$\text{DEMAND}_\lambda^C\left(Y^{\text{LOPT}}\right) \geq \text{DEMAND}_\lambda^C\left(Y^{\text{NBR}(k)}\right) \text{ for } k = 1, ..., p \times (|\mathcal{J}| - p).$$

The pseudocode of the proposed VNS method is provided in Algorithm 1. In Line 1, the search begins with an integer solution Y^{INT}, which is the initial best-known integer solution. In Lines 3–15, the method moves the best-known integer solution Y^{INT} to its neighbor, improving $\mathrm{DEMAND}_{\lambda}^{C}\left(Y^{\mathrm{INT}}\right)$, until Y^{INT} reaches a local optimum Y^{LOPT}. During this process, $\mathsf{is\_local\_optimum} = \mathsf{False}$, indicating that Y^{INT} is not guaranteed to be a local optimum; in this case, the algorithm proceeds. In Lines 5–14, the method enumerates the neighbors Y^{NBR} of Y^{INT}. If a superior neighbor is found (line 9), This method updates the best-known integer solution with the superior neighbor by setting Y^{INT} to Y^{NBR} (Line 10) and sets $\mathsf{is\_local\_optimum}$ to False, by which the best-known integer solution proceeds (Line 11). If there is no superior neighbor (*i.e.*, $\mathsf{is\_local\_optimum} = \mathsf{True}$), the method returns the local optimum $Y^{\mathrm{LOPT}} = Y^{\mathrm{INT}}$. Note that the 1-norm distance between Y^{INT} and Y^{NBR} equals 2, *i.e.*, Y^{NBR} is given by swapping a location $j \in \operatorname{supp}\left(Y^{\mathrm{INT}}\right) = \left\{j \in \mathcal{J} : Y_j^{\mathrm{INT}} = 1\right\}$ of the support of Y^{INT} and a non-support location $k \in \mathcal{J} \setminus \operatorname{supp}\left(Y^{\mathrm{INT}}\right)$.

Algorithm 1 Variable Neighborhood Search

1: Start with any integer solution Y^{INT}
2: Set $\mathsf{is\_local\_optimum} = \mathsf{False}$
3: **while** $\mathsf{is\_local\_optimum} = \mathsf{False}$ **do**
4:   Set $\mathsf{is\_local\_optimum} = \mathsf{True}$
5:   **for** $j \in \operatorname{supp}\left(Y^{\mathrm{INT}}\right)$ and $k \in \mathcal{J} \setminus \operatorname{supp}\left(Y^{\mathrm{INT}}\right)$ **do**
6:     Set $Y^{\mathrm{NBR}} \leftarrow Y^{\mathrm{INT}}$
7:     $Y_j^{\mathrm{NBR}} \leftarrow 0$
8:     $Y_k^{\mathrm{NBR}} \leftarrow 1$
9:     **if** $\mathrm{DEMAND}_{\lambda}^{C}\left(Y^{\mathrm{INT}}\right) < \mathrm{DEMAND}_{\lambda}^{C}\left(Y^{\mathrm{NBR}}\right)$ **then**
10:       Update $Y^{\mathrm{INT}} \leftarrow Y^{\mathrm{NBR}}$
11:       Set $\mathsf{is\_local\_optimum} = \mathsf{False}$
12:       **break**
13:     **end if**
14:   **end for**
15: **end while**
16: **return** the local optimum $Y^{\mathrm{LOPT}} = Y^{\mathrm{INT}}$

4.3 Adaptive randomized rounding

In this section, the VNS method is randomized and developed into an ARR procedure. Given a fractional solution $\tilde{Y} \in [0,1]^{\mathcal{J}} \subset \mathbb{R}^{\mathcal{J}}$, designated as a *seed*, randomized rounding determines an integer solution Y^{INT} near the seed. A randomized rounding procedure determines multiple integer solutions and selects the best-known integer solution Y^{BEST}. An ARR procedure moves the seed $\tilde{Y}$ toward the best-known integer solution Y^{BEST} to obtain a better integer solution near Y^{BEST} (*i.e.*, a better integer solution in the approximate neighborhood of Y^{BEST}).

The traditional randomized rounding procedure first solves the relaxation problem.

$$\mathrm{DEMAND}_{\lambda}^{C}\left(\tilde{Y}^{*}\right)=\max\left\{\mathrm{DEMAND}_{\lambda}^{C}\left(\tilde{Y}\right):\tilde{Y}\in\mathrm{REL}(p)\right\} \tag{43}$$

and then finds an integer solution Y^{INT} based on the fractional optimal solution $\tilde{Y}^{*}$ such that

$$P\left[Y_{j}^{\mathrm{INT}}=1\right]=\tilde{Y}_{j}^{*}\text{ for }j\in\mathcal{J}. \tag{44}$$

A randomized rounding procedure conducts multiple trials to find integer solutions $Y^{\mathrm{INT}(t)}, t=1,...,T$ and selects the best solution Y^{BEST} as follows:

$$Y^{\mathrm{BEST}}=\arg\max\left\{\mathrm{DEMAND}_{\lambda}^{C}\left(Y^{\mathrm{INT}(t)}\right):t=1,...,T\right\}.$$

Because it is not easy to solve relaxation (43), the initial seed for our randomized rounding procedure will be the centroid $\dot{Y}=\left(\dot{Y}_{j}=0.5:j\in\mathcal{J}\right)$ of $[0,1]^{\mathcal{J}}$, and the seed will move to converge to the best-known integer solution Y^{BEST} thus far, searching for better integer solutions in the vicinity of the best-known integer solution. Because $\tilde{Y}$ is interpreted as the likelihood of Y^{INT}, we consider only $\tilde{Y}\in[0,1]^{\mathcal{J}}\subseteq\mathbb{R}^{\mathcal{J}}$, with components $\tilde{Y}_{j}$ between 0 and 1. Section 4.3.1 describes the three main components of our ARR technique and Section 4.3.2 describes the construction of the full ARR procedure.

4.3.1 Two main components

Randomized rounding

Instead of the exact probability distribution (44) of traditional randomized rounding, our randomized rounding procedure perturbs a given seed $\tilde{Y}$ randomly for the perturbed seed $\tilde{Y}^{\mathrm{PTB}}$ to satisfy proportionality:

$$\tilde{Y}_{j}<\tilde{Y}_{k}\Rightarrow P\left[Y_{j}^{\mathrm{INT}}=1\right]<P\left[Y_{k}^{\mathrm{INT}}=1\right]\text{ for }j\neq k\in\mathcal{J}, \tag{45}$$

where $Y^{\mathrm{INT}}=\mathrm{ROUND}\left(\tilde{Y}^{\mathrm{PTB}}\right)$ is the integer solution given by rounding the perturbed seed $\tilde{Y}^{\mathrm{PTB}}$ (*i.e.*, $Y_{j}^{\mathrm{INT}}=1$ if and only if $\tilde{Y}_{j}^{\mathrm{PTB}}$ is one of the p largest components of $\tilde{Y}^{\mathrm{PTB}}$). We consider the following two perturbation methods:

$$\tilde{Y}_{j}^{\mathrm{PTB}}=\tilde{Y}_{j}\tilde{\pi}_{j}\text{ with }\tilde{\pi}_{j}\in\mathrm{U}\left[0,1\right]\text{ for }j\in\mathcal{J}, \tag{46}$$

$$\tilde{Y}_{j}^{\mathrm{PTB}}=\tilde{Y}_{j}+(1-\tilde{Y}_{j})\tilde{\pi}_{j}\text{ with }\tilde{\pi}_{j}\in\mathrm{U}\left[0,1\right]\text{ for }j\in\mathcal{J}, \tag{47}$$

which are uniform distributions $\mathrm{U}[0, \tilde{Y}_j]$ and $\mathrm{U}[\tilde{Y}_j, 1]$ respectively. Both perturbation methods (46) and (47) satisfy (45) and

$$P\left[\mathrm{ROUND}\left(\tilde{Y}^{\mathrm{PTB}}\right) = Y^{\mathrm{INT}}\right] = 1 \text{ for } \tilde{Y} = Y^{\mathrm{INT}} \text{ with } \sum_{j \in \mathcal{J}} \tilde{Y}_j = p.$$

Moving seed

Because it is not easy to solve relaxation (43), the initial seed for our randomized rounding procedure will be the centroid $\tilde{Y}^{(0)} = \dot{Y} = \left(\dot{Y}_j = 0.5 : j \in \mathcal{J}\right)$ of $[0,1]^{\mathcal{J}} \subset \mathbb{R}^{\mathcal{J}}$, and the seed will move to converge to the best-known integer solution Y^{BEST} while better integer solutions are sought in the vicinity of Y^{BEST}. The procedure first finds an integer solution $Y^{\mathrm{INT}(0)} = \mathrm{ROUND}\left(\tilde{Y}^{\mathrm{PTB}(0)}\right)$, where $\tilde{Y}^{\mathrm{PTB}(0)}$ is obtained by perturbing $\tilde{Y}^{(0)} = \dot{Y}$. The first is the best-known integer solution, $Y^{\mathrm{BEST}} = Y^{\mathrm{INT}(0)}$. In each trial $t \geq 1$, the procedure moves the last seed $\tilde{Y}^{(t-1)}$ converging to the best-known integer solution Y^{BEST} by exponential smoothing:

$$\tilde{Y}^{(t)} = (1 - \alpha(t))\, \tilde{Y}^{(t-1)} + \alpha(t) Y^{\mathrm{BKI}}, \tag{48}$$

where $0 < \alpha(t) < 1$, When $\mathrm{DEMAND}_{\lambda}^{C}\left(Y^{\mathrm{BEST}}\right) < \mathrm{DEMAND}_{\lambda}^{C}\left(Y^{\mathrm{INT}(t)}\right)$, the procedure updates the best-known integer solution to $Y^{\mathrm{BEST}} = Y^{\mathrm{INT}(t)}$.

Algorithm 2 Adaptive Randomized Rounding

Initialize $\tilde{Y}^{(0)} \leftarrow \dot{Y} = \left(\dot{Y}_j = 0.5 \text{ for } j \in \mathcal{J}\right)$; $Y^{\mathrm{BEST}} \leftarrow \mathrm{ROUND}\left(\dot{Y}^{\mathrm{PTB}}\right)$; $t \leftarrow 1$; $n^{\mathrm{local}} \leftarrow 0$
while all termination criteria are not met **do**
  Adjust smoothing constant $\alpha(t)$ by Decelerator (50)
  Move $\tilde{Y}^{(t)}$ by exponential smoothing (48) with the smoothing constant $\alpha(t)$
  Find $Y^{\mathrm{INT}(t)} = \mathrm{ROUND}\left(\tilde{Y}^{\mathrm{PTB}(t)}\right)$
  if $\mathrm{DEMAND}_{\lambda}^{C}\left(Y^{\mathrm{INT}(t)}\right) > \mathrm{DEMAND}_{\lambda}^{C}\left(Y^{\mathrm{BEST}}\right)$ **then**
    $Y^{\mathrm{BEST}} \leftarrow Y^{\mathrm{INT}(t)}$
    $n^{\mathrm{local}} \leftarrow 0$
  else if $Y^{\mathrm{INT}(t)} = Y^{\mathrm{BEST}}$ **then**
    $n^{\mathrm{local}} \leftarrow n^{\mathrm{local}} + 1$
    $p^{\mathrm{R}} \leftarrow \min\left(\frac{n^{\mathrm{local}}}{20}, 1\right) * \mathrm{RMSD}\left(\tilde{Y}^{(t)}\right)$
    Reset $\tilde{Y}^{(t)} \leftarrow \dot{Y}$ and $n^{\mathrm{local}} \leftarrow 0$ with probability of p^{R}
  else
    $n^{\mathrm{local}} \leftarrow 0$
  end if
  $t \leftarrow t + 1$
end while
Return the best-known integer solution Y^{BEST}

4.3.2 Full procedure

In this section, we discuss the development of a full ARR procedure that changes seed $\tilde{Y}$. By converging to the best known integer solution Y^{BEST}, the seed obtains a better integer solution near Y^{BEST}. However, if the seed is extremely close to the best-known integer solution, it adheres to the best-known integer solution and determines only the same integer solution. To resolve this problem, ARR adopts the following two criteria:

- Criterion 1. Control of Speed: The seed $\tilde{Y}$ slows down near the best-known integer solution.
- Criterion 2. Rule of Reset: If the seed is excessively close, reset $\tilde{Y} = \dot{Y}$.

During the process of converging to the best-known integer solution, a decelerator slows the convergence (Criterion 1). To determine $\alpha(t)$, we define the root-mean-square deviation (RMSD) of $\tilde{Y}^{(t)}$ with respect to the centroid $\dot{Y}$:

$$\mathrm{RMSD}(\tilde{Y}^{(t)}) = \sqrt{\frac{\sum_{j\in\mathcal{J}}\left(\tilde{Y}_j^{(t)} - 0.5\right)^2}{|\mathcal{J}|}}\,, \tag{49}$$

which ranges from 0 at $\dot{Y}$ to 0.5 at an integer solution Y^{INT}. Thus, as $\tilde{Y}^{(t)}$ converges to the best-known integer solution Y^{BEST}, the RMSD converges to 0.5. The smoothing constant $\alpha(t)$ is determined by the decelerator (DEC), which slows $\alpha = \frac{1}{2}$ (at $\dot{Y}$) to $\alpha = \frac{1}{1+e^2}$ (at Y^{BEST}):

$$\alpha(t) = \mathrm{DEC}(t-1) = \frac{1}{1 + e^{4*\mathrm{RMSD}(\tilde{Y}^{(t-1)})}}\,. \tag{50}$$

The pseudocode of the ARR procedure is presented in Algorithm 2. The number of consecutive trials in which the same solution Y^{BEST} was obtained is denoted by n^{local} (for Criterion 2). Because the same solution Y^{BEST} is found in the run of trials (*i.e.*, n^{local} increases), the probability that the seed will be reset to $\dot{Y}$ increases. In particular, the probability p^{R} of the reset reaches $\mathrm{RMSD}\left(\tilde{Y}^{(t)}\right)$ after 20 trials with the same Y^{BEST}. Thus, a computational time limit (TL) was set as the termination criterion for the entire procedure.

5 Numerical experiments

In this section, numerical experiments were conducted to validate the three proposed solution methods: MILP, VNS, and ARR. In this study, we focus on ARR to solve large-scale P&R FLPs under NL demand functions. In the next section (Section 6), ARR addresses a real-world case of P&R FLP in the Seoul metropolitan area.

For the experiments described in this section, we randomly generated simulation data from the Seoul metropolitan area. The data randomly sampled individuals I and alternatives J from 3,405 origin–destination pairs and 170 public parking spaces across the entire Seoul metropolitan area of South Korea. Table 2 lists the profiles of the datasets. Each dataset consisted of ten random instances for which we performed the computational experiments. We reported the mean, standard error, maximum, and minimum of each statistic for a sample size of 10.

Table 2 Simulation Data

$\lvert I \rvert$	$\lvert J \rvert$	number of instances (IDs)
300	30	10 (inst0 - inst9)
600	60	10 (inst0 - inst9)
900	90	10 (inst0 - inst9)
1200	120	10 (inst0 - inst9)

The computational experiments were performed on a Pittsburgh Supercomputer [38] running on 128 cores with 256 GB of RAM. The supercomputer performed parallel computations of the ARR procedure with 128 threads and simultaneously ran 128 iterations. Using the same 128 available cores, the supercomputer solved the MILPs by running GUROBI 12.0.1 [8]. To run the ARR procedure and the MILP solver, we wrote code in Python 3 [37]. All data, codes, and results were posted on GitHub.[2]

5.1 Exact Solution Method

In this section, we compare CMILP-NL (39) with Enhanced(λ), which strengthens CMILP-NL by adding constraint (42) bounding the continuous variables that indicate choice probabilities. We turned off the cutting planes that were added in the default setting of the solver (GUROBI) and added to Enhanced(λ) only the important cutting planes listed in Table 1. In practice, the branch-and-cut procedures for (CMILP-NL) and (Enhanced) may provide suboptimal solutions or may be aborted owing to infeasibility. The preference weights are exponential functions, and the ratios between the preference weights may be extremely large or small. In (36)-(38), the numeric errors caused by extreme coefficients may cut off the feasible solutions because a pair of inequalities induces an equation. To avoid the infeasibility caused by numerical errors, we added the tolerance $\varepsilon = 10^{-7}$ to the right-hand side of (36) and (38) and subtracted the tolerance from the right-hand side of (37).

Table 3 compares CMILP-NL (39) with the Enhanced(λ) of various log-sum parameters, $\lambda = 1.00, 0.75, 0.50, 0.25$ for the smallest instances ($|I| = 300$ and $|J| = 30$). The enhancements introduced in Section 4.1 can at least double the speed. As the log-sum parameter decreased, the rate of improvement increased. The optimal solutions to CMILP-NL are slightly better than those for Enhanced(λ) of $\lambda = 0.75$ and $\lambda = 0.50$ as shown by

$$\text{Optimality Gap} = \frac{\text{(CMILP-NL)} - \text{(Enhanced)}}{\text{(CMILP-NL)}} \times 100\%$$

[2] https://github.com/s-shim/cap_pnr_nl2

When $\lambda = 0.25$, the optimality gap is negative, and the solutions to the Enhanced(λ) are better than those to CMILP-NL.

Table 3 CMILP-NL vs. Enhanced(λ) on 10 instances under the NL demand function

$\lvert I\rvert$	$\lvert J\rvert$	p	Instances	λ	CMILP-NL		Enhanced(λ)		Optimality	Row
					Time	B&B	Time	B&B	Gap	Type
300	30	15	10	1.00	253.47 s	331.8	101.44 s	636.4	0.00000 %	Mean
					44.47 s	65.6	14.99 s	125.6	0.00000 %	SE
					623.20 s	794.0	183.87 s	1270.0	0.00000 %	Max
					147.62 s	114.0	55.28 s	190.0	0.00000 %	min
300	30	15	10	0.75	100.26 s	434.5	46.58 s	764.7	0.00009 %	Mean
					16.03 s	76.6	14.42 s	328.3	0.00009 %	SE
					220.53 s	688.0	166.11 s	3538.0	0.00090 %	Max
					51.40 s	35.0	16.10 s	90.0	0.00000 %	min
300	30	15	10	0.50	21.95 s	494.6	7.42 s	347.5	0.00076 %	Mean
					3.12 s	131.7	1.01 s	72.5	0.00076 %	SE
					44.88 s	1380.0	13.57 s	697.0	0.00076 %	Max
					8.03 s	1.0	3.74 s	12.0	0.00763 %	min
300	30	15	9*	0.25	191.01 s	6994.6	1.76 s	229.5	−0.67465 %	Mean
					181.06 s	6795.3	0.18 s	55.1	0.64665 %	SE
					1638.86 s	61356.0	2.84 s	526.0	0.10441 %	Max
					2.27 s	47.0	1.21 s	17.0	−5.83778 %	min

Note. *inst1 was not solved by CMILP-NL with $\lambda = 0.25$ within 1 hour of time limit.

On most of the medium-scale instances ($|I| = 600$ and $|J| = 60$), CMILP-NL (39) takes too long (more than two hours). In Section 5.2.1, Enhanced(λ) provides the benchmark solutions to verify ARR under the NL demand function.

5.2 Heuristic Algorithms

To address a large-scale P&R FLP, Section 4 introduces two heuristic algorithms: VNS and ARR. In practice, ARR outperforms VNS and this study focuses on ARR. In Section 5.2.1, we compare the ARR with Enhanced(λ) on the smallest instances of ($|I| = 300, |J| = 30$) for various $\lambda = 1.00, 0.75, 0.50, 0.25$. For ($|I| = 600, |J| = 60$), the Enhanced($\lambda$) requires more than 4 h, and Section 5.2.2 compares the ARR with CMILP (24) under MNL demand function on the medium-scale instances listed in Table 2. Section 5.2.3 verifies that ARR outperforms VNS on a large-scale P&R FLP.

5.2.1 ARR vs. Enhanced(λ)

In this section, we compare the performance of Enhanced(λ) with that of the ARR procedure and observe that the ARR procedure reaches the exact optimal solution much faster than the MILP on the smallest instances. Moreover, the ARR procedure detects the minor suboptimality of the MILP solutions caused by the numerical errors discussed in Section 5.1. A comparison of the MILP and ARR solutions also provided an idea of stopping criteria for the ARR procedure.

Table 4 compares ARR with Enhanced(λ) of various log-sum parameters, $\lambda = 0.75, 0.50, 0.25$ for the medium-scale instances ($|I| = 600$ and $|J| = 60$). Let $Y_{\lambda}^{\mathrm{BEST}}$ denote the best solution obtained by using the ARR procedure within (Time Limit). The time and trials to first reach $Y_{\lambda}^{\mathrm{BEST}}$ through 128 simultaneous iterations are shown in columns (Time) and (Trial). The ARR procedure achieved the optimal solution for every instance. For $\lambda \geq 0.50$, the ARR procedure solved the medium-scale instances optimally much faster than CMILP under MNL demand function and Enhanced(λ) with $\lambda = 0.75, 0.50$. For Enhanced(λ) with $\lambda < 1.00$, suboptimal optimization was detected by the ARR solution owing to numerical errors in the MILP. The suboptimality is represented by negative optimality gaps (Optimality Gap), where

$$\text{Optimality Gap} = \frac{(\text{Enhanced}) - \text{DEMAND}_{\lambda}^{C}\left(Y_{\lambda}^{\mathrm{BEST}}\right)}{(\text{Enhanced})} \times 100\%.$$

The time taken by the ARR to reach the best solution is much less than half of the time limit (TL). This provides a stopping criterion for the ARR procedure: The ARR procedure may add elapsed time to the time limit (TL) to allow the best solution to be found within the first half of the entire TL.

Table 4 Enhanced(λ) vs. ARR with (TL: Time Limit) on 10 instances under the NL demand function

$\|I\|$	$\|J\|$	p	Instances	λ	Enhanced(λ)	ARR (across 10 instances)			Optimality	Row
					Time	Time	Trial	TL	Gap	Type
600	60	30	10	1**	217.05 s	26.79 s	847.20	600 s	0.0000 %	Mean
					82.51 s	1.91 s	73.63	0 s	0.0000 %	SE
					892.90 s	39.48 s	1335-th	600 s	0.0000 %	Max
					52.31 s	18.26 s	527-th	600 s	0.0000 %	min
600	60	30	10	0.75	10159.32 s	28.27 s	901.30	600 s	−0.07730 %	Mean
					1960.37 s	1.33 s	49.46	0 s	0.07678 %	SE
					21356.51 s	36.85 s	1194-th	600 s	0.00000 %	Max
					4301.43 s	21.83 s	625-th	600 s	−0.76831 %	min
600	60	30	10	0.50	1327.46 s	33.81 s	1116.70	600 s	−0.01388 %	Mean
					388.19 s	2.34 s	99.37	0 s	0.00951 %	SE
					4216.12 s	49.04 s	1735-th	600 s	0.00000 %	Max
					372.72 s	24.78 s	724-th	600 s	−0.08413 %	min
600	60	30	9*	0.25	83.60 s	29.99 s	971.55	600 s	−0.10926 %	Mean
					23.06 s	1.17 s	50.73	0 s	0.03468 %	SE
					176.06 s	36.11 s	1241-st	600 s	0.00000 %	Max
					12.63 s	24.46 s	751-st	600 s	−0.30826 %	min

Note. *inst4 was not solved by Enhanced(λ) with $\lambda = 0.25$ within 8 hours of time limit. Enhanced(λ) with $\lambda = 1$** took too long (longer than two hours) on all instances. Instead, CMILP optimally solved the instances, providing the benchmark solutions under MNL demand function as compared with ARR in Table 5.

5.2.2 ARR vs. MILP under MNL demand function

In this section, we compare the performance of CMILP with that of the ARR procedure under the MNL demand function and observe that the ARR procedure reaches the exact optimal solution much faster than CMILP on large-scale instances. To avoid infeasibility, we add (8) and (9) to CMILP only when the odds are not excessively small (*e.g.*, $\text{odds}(i,j) > \varepsilon = 10^{-7}$), and we set the choice probability to zero with excessively small odds (*e.g.*, adding constraints $P_{ij} = 0$ for $\text{odds}(i,j) < \varepsilon = 10^{-7}$).

Table 5 compares the computational performances of the CMILP and ARR procedures under the MNL demand function. Let Y^{CMILP} and Y^{BEST} denote the optimal solution to CMILP and the best solution obtained by using the ARR procedure within (Time Limit). The ARR procedure achieved the optimal solution for every instance. As the problem size increases, the ARR procedure becomes increasingly faster than the MILP solver (GUROBI). For the instances of $(|I| = 900, |J| = 90, p = 45)$, suboptimal optimization was detected by the ARR solution owing to numerical errors in the MILP. The suboptimality is represented by negative optimality gaps (Optimality Gap), where

$$\text{Optimality Gap} = \frac{\text{DEMAND}^C\left(Y^{\text{CMILP}}\right) - \text{DEMAND}^C\left(Y^{\text{BEST}}\right)}{\text{DEMAND}^C\left(Y^{\text{CMILP}}\right)} \times 100\%.$$

Table 5 CMILP vs. ARR with (TL: Time Limit) on 10 instances under the MNL demand function

$\|I\|$	$\|J\|$	p	Instances	CMILP	ARR (across 10 instances)			Optimality	Row
				Time	Time	Trial	TL	Gap	Type
300	30	15	10	4.17 s	1.47 s	184.50	300 s	0.0000 %	Mean
				0.57 s	0.12 s	15.82	0 s	0.0000 %	SE
				7.39 s	2.08 s	275-th	300 s	0.0000 %	Max
				1.86 s	0.91 s	102-nd	300 s	0.0000 %	min
600	60	30	10	217.05 s	26.79 s	847.20	600 s	0.0000 %	Mean
				82.51 s	1.91 s	73.63	0 s	0.0000 %	SE
				892.90 s	39.48 s	1335-th	600 s	0.0000 %	Max
				52.31 s	18.26 s	527-th	600 s	0.0000 %	min
900	90	45	10	3227.50 s	143.23 s	1884.10	1800 s	−0.0435 %	Mean
				481.91 s	5.18 s	79.96	0 s	0.0235 %	SE
				6975.69 s	177.15 s	2350-th	1800 s	0.0000 %	Max
				1399.54 s	117.07 s	1523-th	1800 s	−0.1906 %	min
1200	120	60	10	> 4 h	574.35 s	4129.20	3600 s	—	Mean
				—	31.32 s	246.73	0 s	—	SE
				> 4 h	735.41 s	5337-th	3600 s	—	Max
				> 4 h	406.10 s	2828-th	3600 s	—	min

For $(|I| = 1200, |J| = 120, p = 60)$, there are no instances in which the solver (GUROBI) can solve optimally within the time limit of 4 h. While MILP solved only one instance (ID: inst0) in two days, the ARR procedure performed 128 simultaneous iterations on the same instance with a 1 h time limit. Of the 128 ARR procedure iterations, 82 reached an optimal solution. Table 6 lists the results of the 82 optimal ARR procedure iterations. The best iteration reached the optimal solution within 10 min (564.42 s), whereas the solver for the CMILP problem took more than two days. The minimum time (564.42 s $<$ 0.5 h) fell within 30 mins, meeting the stopping criterion.

Table 6 CMILP vs. ARR (82 optimal iterations among 128 cores) on a large-scale MNL model (ID: inst0) of ($|I| = 1200, |J| = 120, p = 60$)

$\lvert I\rvert$	$\lvert J\rvert$	p	Instances	CMILP	ARR (82 optimal iterations)			Row
				Time	TL	Time	Trial	Type
1200	120	60	1 (inst0)	2.64 d	1 h	2104.92 s	16695.15	Mean
						88.46 s	726.98	SE
						3580.92 s	29251-th	Max
						564.42 s	4039-th	min

5.2.3 ARR vs. VNS

The two proposed heuristic algorithms, VNS and ARR, can address large-scale instances. On the NL model for example, (ID: inst0) of ($|I| = 1200, |J| = 120, p = 60$) with the log-sum parameter $\lambda = 0.5$, the computational experiment verified that ARR outperformed VNS. Table 7 presents the performance test results. Beginning with a random integer solution, VNS reached a local optimum Y_λ^{LOPT} of demand 9140.31 in 13665.57 seconds ($<$ 4 h). With a TL of 4 h, a single iteration of ARR reached its best solution Y_λ^{BEST} of demand 9486.11 at the 8229th second, when VNS required 8992.84 seconds. The demand gap is

$$\text{Demand Gap} = \frac{\text{DEMAND}_\lambda^C\left(Y_\lambda^{\text{BEST}}\right) - \text{DEMAND}_\lambda^C\left(Y_\lambda^{\text{LOPT}}\right)}{\text{DEMAND}_\lambda^C\left(Y_\lambda^{\text{BEST}}\right)} \times 100\% = 3.65\%.$$

Figure 3 compares the demand curves of ARR and VNS over the entire time horizon. ARR is superior to VNS regarding computational time and solution quality; therefore, we used to solve the NL model for the remainder of this study.

Table 7 Demand Gap between VNS (Y_λ^{LOPT}) and ARR (Y_λ^{BEST}) on the NL model of $\lambda = 0.5$

$\lvert I\rvert$	$\lvert J\rvert$	p	Instances	λ	Time	DEMAND_λ^C		Demand
						VNS	ARR	Gap
1200	120	60	1 (inst0)	0.50	8229.00 s (Y_λ^{BEST})	8992.84	9486.11	5.20 %
					13665.57 s (Y_λ^{LOPT})	9140.31	9486.11	3.65 %

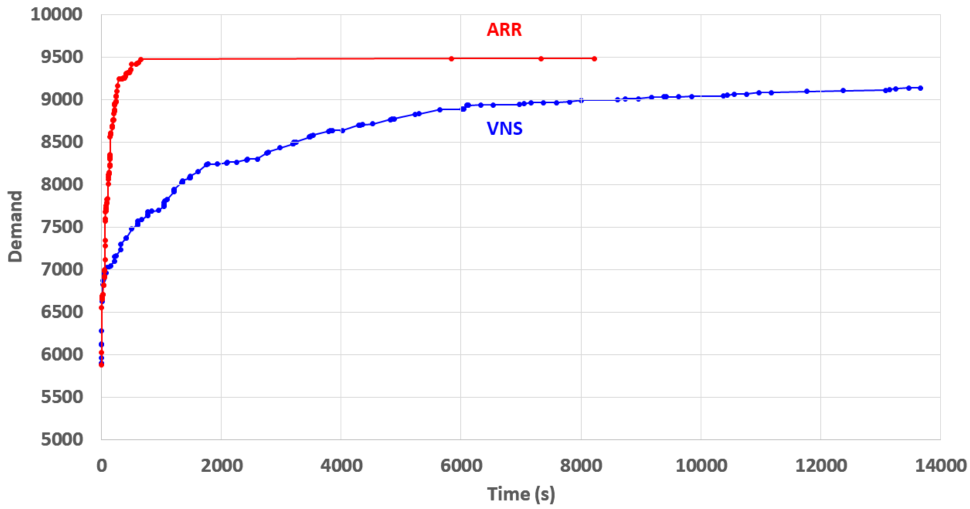

Fig. 3 ARR vs. VNS on the NL model (inst0) of ($|I| = 1200, |J| = 120, p = 60$) with $\lambda = 0.5$

6 Case study: Seoul metropolitan area

The numerical experiments conducted in this study used traffic and geographic data from the Seoul metropolitan area. According to the Organization for Economic Co-operation and Development [26], Seoul is the 6th largest metropolitan area in the world, with a population of 24.3 million. In 2021, the modal share of public transportation (*i.e.*, bus and subway) was 52.9% and that of private cars was 38% [31]. As metropolitan areas have become more populated, roads have become more congested, as evidenced by the decrease in the average speed of cars over the past decade [32]. Although many people still use private cars, the modal share of public transportation in the Seoul metropolitan area is high owing to its wide coverage and high service level. As a result, in 2022, the average speed of private cars in downtown Seoul was 19.2 km/h; by contrast, the average speed of buses in downtown Seoul was 21.4 km/h during the morning peak [32]. However, despite the high modal share of public transportation and the congested road traffic in downtown areas, P&R has not yet been widely implemented in Seoul.

This study uses calibrated traffic volume data from the publicly available Korea Transport DataBase (KTDB)[3]. The KTDB measures traffic volumes in Korea every five years and calibrates it every year. The traffic data in the Seoul metropolitan area include the origin–destination traffic volumes for the smallest administrative unit (*i.e.*, Dong in Seoul) by mode (*e.g.*, private car, bus, subway, and walking). This study assumed that commuters mostly used P&Rs during the morning peak hours, and our numerical experiments used traffic volumes between 8 am and 10 am. Commuters

[3] https://www.ktdb.go.kr

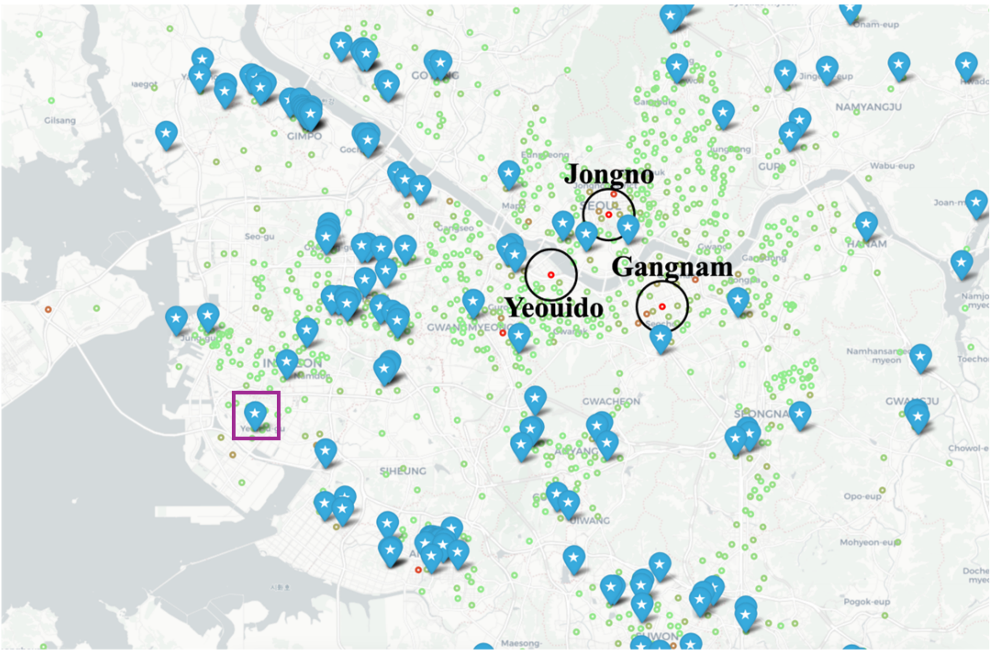

Fig. 4 Top three destinations (black circles), origin/destination (small dots), and P&R facilities (blue balloons) in the Seoul metropolitan area

move from home to work in the morning, and there are several common destinations. Figure 4 shows the inbound traffic volume during the morning peak, where red indicates high-inbound (*i.e.*, arriving) traffic. The top three destinations are the central business districts (CBDs) in Yeouido, Gangnam, and Jongno. Given that many companies are located in these CBDs and most commuters work in the morning, the numerical experiments used traffic data destined for one of these CBDs.

There are 1,135 traffic analysis zones (TAZs) in the Seoul metropolitan area, and each origin–destination (O-D) pair connects two TAZs. The TAZ in Seoul is a Dong, the smallest administrative unit. TAZs that are not very close (*e.g.*, $>$ 4 km) to a destination (*e.g.*, the CBD) are considered feasible origins. Therefore, each destination, *that is*, Yeouido, Gangnam, and Jongno, had approximately 1,000 routes, and the total number of O-D pairs (or segments) in the numerical experiments exceeded 3,000. In total, 170 public parking lots in the Seoul metropolitan area, obtained from a previous study [34], were considered as candidates for P&Rs.

KTDB also provides utility functions for private cars, taxis, buses, and subways [18]. This study used the utility function of public transportation for P&R's because P&R shares many characteristics (such as transfer and wait) with public transportation. (P&Rs are not yet widely used in Korea, and the study did not use the utility functions of this small number of P&Rs for potential P&R's). The KTDB categorizes utility functions according to the purpose of the trip as home-based, home-based, or non-home-based. As this study considered only commuters, the utility functions for home-based work were used, and those for private cars and public transportation were as follows:

$$V_{i0} = \beta^{\text{TIME}}\text{TIME}_{i0} + \beta^{\text{COST,car}}\text{COST}_{i0} \quad (51)$$

$$\begin{aligned} V_{ij} &= \text{ASC} + \beta^{\text{TIME,in}}\text{TIME}_{ij}^{\text{in}} \\ &\quad + \beta^{\text{TIME,out}}\text{TIME}_{ij}^{\text{out}} + \beta^{\text{COST,pnr}}\text{COST}_{ij}. \end{aligned} \quad (52)$$

where β^{TIME} and $\beta^{\text{COST,car}}$ are the travel time and cost coefficients for private cars, respectively; and $\beta^{\text{TIME,in}}$, $\beta^{\text{TIME,out}}$, and $\beta^{\text{COST,pnr}}$ are the in-vehicle, out-vehicle, and cost coefficients for P&R, respectively. TIME_{i0} and COST_{i0} are the travel time and cost for a private car for segment i, respectively, and $\text{TIME}_{ij}^{\text{in}}$, $\text{TIME}_{ij}^{\text{out}}$, and COST_{ij} are the in-vehicle time, out-vehicle time, and cost of P&R j for segment i, respectively. ASC is the alternative specific constant for P&R. Note that ASC is not defined for private cars, because it is the baseline mode. The coefficients of the utility functions are given in Table 8 [18]. The time unit is minutes, and the cost unit is 100 KRW.

Table 8 Utility function coefficients

Coefficient	Value
β^{TIME}	-0.07227
$\beta^{\text{COST,car}}$	-0.01915
$\beta^{\text{TIME,in}}$	-0.04908
$\beta^{\text{TIME,out}}$	-0.10216
$\beta^{\text{COST,pnr}}$	-0.11524
ASC	1.88940

TIME_{i0} was calculated by dividing the distance of segment i by the average speed of private cars in downtown Seoul. The segment distance was multiplied by a circuity factor, which was 1.215 for the Seoul metropolitan area [12]. COST_{i0} is the sum of operating costs and parking fees of the CBDs. The operating costs of private cars include fuel, replacement, maintenance, and depreciation [18]. The average monthly parking fee in Seoul is 250,000 KRW, and the daily parking fee is used for the utility function because commuters who use P&R park their cars from morning till evening. $\text{TIME}_{ij}^{\text{in}}$ consists of two parts: the access time from a TAZ (*i.e.*, home) to a P&R by private car and the travel time from the P&R to a CBD by public transport. The first part was calculated by dividing the access distance by the average speed of private cars, and the second part was calculated by dividing the travel distance by the average speed of buses. Note that the same circuity factor is multiplied by the distance. $\text{TIME}_{ij}^{\text{out}}$ is the transfer and waiting time at P&R and is assumed to be 20 min in this study. Similar to $\text{TIME}_{ij}^{\text{in}}$, COST_{ij} comprises private cars and public transportation. The first comprises the operating cost of a private car for access from a TAZ to a P&R, whereas the second includes the daily parking fee at a P&R and the public transportation fare. Each P&R has a different price plan; thus, this study collected all price plans for public parking lots in the Seoul metropolitan area. Public transportation fares in the Seoul metropolitan area include a base fare and an additional fare. The base fare was 1,500 KRW up to 10 km, and an additional 100 KRW was charged every 5 km.

Table 9 reports the solutions with various $p = 20, 40, 80$ given the 170 potential P&Rs values in the Seoul metropolitan area. We computed the approximate (Demand Gap), defined as

$$\text{Demand Gap} = \frac{\text{DEMAND}^{C}_{\lambda'}\left(Y^{\text{BEST}}_{\lambda'}\right) - \text{DEMAND}^{C}_{\lambda'}\left(Y^{\text{BEST}}_{\lambda}\right)}{\text{DEMAND}^{C}_{\lambda'}\left(Y^{\text{BEST}}_{\lambda'}\right)},$$

where $Y^{\text{BEST}}_{\lambda}$ is the best solution provided by the ARR to the P&R FLP under the nested logit demand function $\text{DEMAND}^{C}_{\lambda}$. The literature assumes that P&R facilities are independent; *that is*, $\lambda = 1$. Although we generalize the log-sum parameter $0 < \lambda < 1$, we assume that λ is not significantly different from $\lambda = 1$. In this case study, we assumed that $0.5 < \lambda < 1$. With $\lambda' = 0.5$, the demand gap of the MNL solutions (*i.e.*, $Y^{\text{BEST}}_{\lambda=1}$) is not negligible: 7.45, 6.59, and 5.74 % for $p = 20, 40, 80$, respectively. By contrast, $Y^{\text{BEST}}_{\lambda=0.75}$ has a gap of < 2 % in the extreme cases ($\lambda' = 0.5$ or $\lambda' = 1$). Note that the ARR reached the best solutions within 50 % of the computational time limit.

Consider the best solution $Y^{\text{BEST}}_{\lambda}$ with $\lambda = 0.75$ for $p = 20, 40, 80$. All ($p = 20$) P&Rs were also included among the ($p = 40$) P&Rs and among the ($p = 80$) P&Rs. Except for the P&R of ID 152 among the ($p = 40$) P&Rs, the other 39 P&Rs were also included among the ($p = 80$) P&Rs. (The P&R of ID 152 is highlighted by a rectangle in Figure 4.) This phased expansion plan encourages policymakers to start installing a small number immediately (*e.g.*, $p = 20$) of P&Rs, subject to the present budget limitations. Subsequently, an optimal expansion to a larger number (*e.g.*, $p = 40$ or $p = 80$) of optimal P&Rs will include all initial installations of (*e.g.*, $p = 20$) P&Rs.

7 Conclusion

This study considered capacitated p-hub solutions for P&R FLP under an NL demand function and introduced three polyhedral approaches: MILP, VNS, and ARR. We improved MILP (CMILP-NL) by adding constraints that bound the continuous variables representing choice probabilities. This improvement doubles the computational performance of MILP. The improved MILP (denoted as Enhanced(λ)) provided benchmark solutions to test VNS and ARR on simulation data generated from the Seoul metropolitan area. In large-scale instances, ARR outperformed VNS and tackled real-world cases in the Seoul metropolitan area. In the entire Seoul metropolitan area, the optimal solutions present a phased expansion plan that encourages policymakers to start immediately installing a small number of P&Rs.

Considering the reality of the Seoul metropolitan area, we assume that a commuter who drives to a P&R site will continue driving to their final destination (without finding other P&R sites) if the site is full. To capture the real conditions in other areas, subsequent research may consider further constraints, ensuring accessibility issues, such as commuter behavior in response to parking lots being full. Even within the same Seoul metropolitan area, customers who fail to find a parking space at their favorite

Table 9 Best Demand $\text{DEMAND}^C_\lambda\left(Y^{\text{BEST}}_\lambda\right)$ across Different p and λ. Demand Gap between $\text{DEMAND}^C_{\lambda'}\left(Y^{\text{BEST}}_\lambda\right)$ and $\text{DEMAND}^C_{\lambda'}\left(Y^{\text{BEST}}_{\lambda'}\right)$ in the Case Study

$\lvert I \rvert$	$\lvert J \rvert$	p	λ	Adaptive Randomized Rounding $\left(Y^{\text{BEST}}_\lambda\right)$				λ'	Demand
				Iterations	DEMAND^C_λ	Min Time	TL		Gap
3405	170	20	1.00	128	7874.38	501.46 s	15 h	1.00	0.00 %
								0.75	4.11 %
								0.50	7.45 %
			0.75	128	7765.03	363.68 s	15 h	1.00	**0.64** %
								0.75	**0.00** %
								0.50	**0.00** %
			0.50	128	7702.38	375.73 s	15 h	1.00	0.64 %
								0.75	0.00 %
								0.50	0.00 %
		40	1.00	128	12005.91	1980.06 s	15 h	1.00	0.00 %
								0.75	3.30 %
								0.50	6.59 %
			0.75	128	11869.74	1533.62 s	15 h	1.00	**0.75** %
								0.75	**0.00** %
								0.50	**0.37** %
			0.50	128	11671.74	1472.39 s	15 h	1.00	1.49 %
								0.75	0.74 %
								0.50	0.00 %
		80	1.00	128	17608.89	4897.79 s	15 h	1.00	0.00 %
								0.75	2.07 %
								0.50	5.74 %
			0.75	128	17075.42	8088.62 s	15 h	1.00	**1.83** %
								0.75	**0.00** %
								0.50	**1.36** %
			0.50	128	16492.29	6594.05 s	15 h	1.00	3.26 %
								0.75	0.60 %
								0.50	0.00 %

site may avoid the same site the next time. A follow-up study may consider different modes such as a P&R site with good availability or a site with poor availability. For details on the choice model used for facility location problem with multiple modes, refer to Krohn et al. [16].

The full procedure of ARR is similar to the framework of the well-known greedy randomized adaptive search procedure (GRASP). In the literature on facility location, GRASP has been proven useful for adding locations that can maximize market share. Developing GRASP for constrained p-hub approaches under the NL demand function and comparing it with the proposed heuristic algorithms, VNS and ARR, would be very interesting. In particular, the use of the simulation of utilities has performed well in discrete choice models with capacity constraints (Ulloa et al. [36]) and should be explored as a future approach.

Acknowledgements

The authors are grateful to the anonymous referees for thoughtful comments that significantly improved this paper. We would like to thank Editage (www.editage.co.kr) for English language editing.

Declarations

- Ethics approval and consent to participate: N/A
- Consent for publication: N/A
- Funding: Not Funded
- Data availability: Data Available at https://github.com/s-shim/cap_pnr_nl2
- Competing interests: The authors declare that they have no competing interests
- Author contribution: S.H.K developed adaptive randomized rounding to solve a large-scale P&R FLP and wrote the first draft of the paper. He also collected the real-world data on the Seoul metropolitan area. F.Q and S.S. generated the simulation data from the real-world data on the Seoul metropolitan area and performed computational experiments on the simulation and the real-world data using the Pittsburgh Supercomputer. S.S. proved the redundancy of a constraint in a MILP of the unconstrained model. The constraint is necessary for the capacitated model.
- Acknowledgements: The authors are grateful to the anonymous referees for thoughtful comments that significantly improved this paper.

References

[1] Aros-Vera, F., V. Marianov and J.E. Mitchell. 2013. p-Hub approach for the optimal park-and-ride facility location problem. *European Journal of Operational Research* **226** (2) 277–285.

[2] Benati, S. 1999. The maximum capture problem with heterogeneous customers. *Computers & Operational Research* **26** (14) 1351–1367.

[3] Benati, S. and P. Hansen. 2002. The maximum capture problem with random utilities: Problem formulation and algorithms. *European Journal of Operational Research* **143** 518–530.

[4] Davis J.M., G. Gallego and H. Topaloglu. 2014. Assortment optimization under variants of the nested logit model. *Operations Research* **62** 250–273.

[5] Dam, T.T., T.T. Ta and E. Moreno. 2022. Submodularity and local search approaches for maximum capture problems under generalized extreme value models. *European Journal of Operational Research* **300** 953–965.

[6] Feo, T. and M. Resende. 1989. A Probabilistic Heuristic for a Computationally Difficult Set Covering Problem. *Operations Research Letters* **8** 67–71.

[7] Feo, T. and M. Resende. 1995. Greedy Randomized Adaptive Search Procedures. *Journal of Global Optimization* **6** 109–133.

[8] Gurobi Optimization, LLC. 2024. Gurobi Optimizer Reference Manual, https://www.gurobi.com

[9] Haase, K. 2009. Discrete location planning. *Technical Report WP-09-07.* Institute for Transport and Logistics Studies, University of Sydney.

[10] Haase, K. and S. Müller. 2014. A comparison of linear reformulations for multinomial logit choice probabilities in facility location models. *European Journal of Operational Research* **232** 689–691.

[11] Hunt, J.D. and S. Teply. 1993. A nested logit model of parking location choice. *Transportation Research, Part B* **27B** 253–265.

[12] Kim, T., Shin, Y., Lee, J. and Suh, K., 2013. Calculation of regional circuity factors using road network distance in South Korea. *Journal of Korea Planning Association*, **48**, 319–329.

[13] Kitthamkesorn, S., A. Chen, S. Ryu, S. Opasanon. 2024. Maximum capture problem based on paired combinatorial weibit model to determine park-and-ride facility locations. *Transportation Research, Part B* **179** p.102855.

[14] Klincewicz, J.G. 1992. Avoiding Local Optima in the p-Hub Location Problem Using Tabu Search and GRASP. *Annals of Operations Research* **40** 283–302.

[15] Koppelman, F.S. and V. Sethi. 2005. A nested logit model of parking location choice. *Transportation Research, Part B* **39** 825–853.

[16] Krohn, R., S. Müller and K. Haase. 2021. Preventive healthcare facility location planning with quality-conscious clients. *OR Spectrum* **43** 59–87.

[17] Krohn, R., S. Müller and K. Haase. 2024. Correction to: Preventive healthcare facility location planning with quality-conscious clients. *OR Spectrum* https://doi.org/10.1007/s00291-024-00776-w

[18] Korea Transport Institute. 2020. National Passenger O/D Supplement Renewal.

[19] Lamontagne, S., M. Carvalho, E. Frejinger, B. Gendron, M. Anjos and R. Atallah. 2023. Optimising electric vehicle charging station placement using advanced discrete choice models. *INFORMS Journal on Computing* **35** (5) 1195–1213.

[20] Ljubíc, I. and E. Moreno. 2018. Outer approximation and submodular cuts for maximum capture facility location problems with random utilities. *European Journal of Operational Research* **266** 46–56.

[21] T. Mai and A. Lodi. 2020. A multicut outer-approximation approach for competitive facility location under random utilities. *European Journal of Operational Research* **284** 874–881.

[22] Méndez-Vogel, G., V. Marianov and A. Lüer-Villagra. 2022. The follower competative facility location problem under the nested logit choice rule. *European Journal of Operational Research* **310** 834–846.

[23] Mladenović, N and P. Hansen. 1997. Variable neighborhood search. *Computers Ops. Res.* **24** (11) 1097–1100.

[24] Motwani, R. and P. Raghavan. 1995. *Randomized Algorithms.* Cambridge University Press, New York, NY, USA

[25] Nemhauser, G.L. and L.A. Wolsey. 1988. *Integer and Combinatorial Optimization.* Wiley, New York.

[26] OECD/European Commission. 2020. *Cities in the World: A New Perspective on Urbanisation*, OECD Urban Studies, OECD Publishing, Paris, https://doi.org/10.1787/d0efcbda-en

[27] Parent P.-L., M. Carvalho, M.F. Anjos and R. Atallah. 2024. Maximum flow-based formulation for the optimal location of electric vehicle charging stations. *Networks* **84** 109–131.

[28] Paneque M.P., M. Bierlaire, B. Gendron and S.S. Azadeh. 2021. Integrating advanced discrete choice models in mixed integer linear optimization. *Transportation Research, Part B* **146** 26–49.

[29] Resende, M.G.C., L.S. Pitsoulis, and P.M. Pardalos. 1997. *Approximate Solution of Weighted MAX-SAT Problems Using GRASP.* In D.-Z. Du, J. Gu, and P.M. Pardalos (eds.), The Satisfiability Problem: Theory and Applications, vol. 35 of DIMACS Series on Discrete Mathematics and Theoretical Computer Science. American Mathematical Society, Providence, RI. pp. 393–405.

[30] Resende M.G.C. 1998. Computing Approximate Solutions of the Maximum Covering Problem with GRASP. *Journal of Heuristics* **4** (2) 161–177.

[31] Seoul Metropolitan Government. 2023. Modal shares. https://news.seoul.go.kr/traffic/?p=289 Retrieved November 28, 2023.

[32] Seoul Metropolitan Government. 2022. 2022 Seoul Traffic Speed Report. Tech. Rep. 51-6110000-002492-10.

[33] Seoul Institute. Parking. https://data.si.re.kr/node/304. Retrieved November 28, 2023.

[34] So, J., Chae, M., Hong, J., Youm, J., Kim, S. H., and Kim, J. 2023. Location Selection Problem of Integrated Mobility Hub for Sustainable Urban Mobility, *Sustainable Cities and Society*, **99**, Article 104950.

[35] Train, K.E. 2009. *Discrete choice methods with simulation.* Cambridge University Press

[36] Ulloa, D.P., Metnan, A., and Frejinger, E. 2024. A Capacitated Collection-and-Delivery-Point Location Problem with Random Utility Maximizing Customers. arXiv preprint arXiv:2411.04200

[37] Van Rossum, G. and Drake, F.L., 2009. Python 3 Reference Manual, CreateSpace, Scotts Valley, CA, USA.

[38] Westropp, J.C., 1996. IEEE Computational Science and Engineering. *IEEE Computational Science and Engineering*, **3** (1), 8–12.

[39] Vazirani, V.V. 2001. *Approximation Algorithms.* Springer Berlin Heidelberg

[40] Williams, H.C.W.L. and J.D. Ortúzar. 1982. Behavioral theories of dispersion and the mis-specification of travel demand models. *Transportation Research, Part B* **16B** (3) 167–219.

[41] Zhang, Y., O. Berman and V. Verter. 2012. The impact of client choice on preventive healthcare facility network design. *OR Spectrum* **34** 349–370.