



A Lagrangian decomposition scheme for choice-based optimization

Meritxell Pacheco Paneque ^{*} Bernard Gendron [†]
Shadi Sharif Azadeh [‡] Michel Bierlaire [§]

August 27, 2021

Report TRANSP-OR 210827
Transport and Mobility Laboratory
École Polytechnique Fédérale de Lausanne
`transp-or.epfl.ch`

^{*}Decision support and operations research group, Department of Informatics, University of Fribourg, meritxell.pacheco@unifr.ch

[†]Département d'informatique et de recherche opérationnelle, Université de Montréal, Montréal, Canada & CIRRELT, bernard.gendron@cirrelt.ca

[‡]Civil Engineering and Geosciences, Department Transport & Planning, TU Delft, Delft, The Netherlands, s.sharifazadeh@tudelft.nl

[§]Transport and Mobility Laboratory (TRANSP-OR), School of Architecture, Civil and Environmental Engineering (ENAC), École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, michel.bierlaire@epfl.ch

Abstract

We consider a family of mixed-integer linear programming (MILP) problems that explicitly includes choice models using the random utility paradigm. In order to use behavioral models published in the literature, the specification relies on a simulation-based approximation to linearize the choice model. The price to pay is the large size of the approximation. To enhance the tractability of the problem, we propose a Lagrangian decomposition methodology inspired by scenario decomposition and scenario grouping in the stochastic programming framework. In addition, we develop an algorithm that exploits the properties of choice-based optimization in order to generate feasible solutions to the original problem from the solution of the Lagrangian problem. Hence, at each iteration of the subgradient method, we provide both an upper and a lower bound to the original problem. This enables the calculation of the duality gap to assess the quality of the generated solutions. Computational results show that the decomposition method provides solutions with optimality gaps below 0.5% and restricted duality gaps within low computational times. We show that scenario grouping leads to high quality feasible solutions and lower duality gaps.

Keywords: choice-based optimization, discrete choice models, combinatorial optimization, Lagrangian decomposition, scenario decomposition, scenario grouping

1 Introduction

Choice-based optimization problems enable planners to make decisions while taking into account individual's choice behavior. They are receiving increasing attention because they allow to explicitly capture the interplay between these decisions and the expected demand provided that the decisions are explanatory variables of the discrete choice model. In this paper, we focus on parametric choice models rooted in the random utility theory. They assume that each individual associates a utility with every alternative and chooses the alternative with the highest utility. Optimization problems that consider these models can be found in facility location (e.g., Benati, 1999, Haase and Müller, 2013), revenue management (e.g., Talluri and Van Ryzin, 2004, Shen and Su, 2007), network pricing (e.g., de Palma et al., 2005, Gilbert et al., 2014b) and routing and pricing for same-day deliveries (e.g., Prokhorchuk et al., 2019, Ulmer, 2020), to name a few.

As utilities are random variables, the expected demand of each alternative is derived from the associated probability functions. These probabilities are highly non-convex and non-linear in the explanatory variables of the choice model, and

they are not always available in closed form. Furthermore, choice-based optimization problems come with the computational burden associated with the individual representation of demand from a planning viewpoint (e.g., capacity allocation). This makes only small to moderate size problems solvable to optimality (e.g., Andersson, 1998, Benati and Hansen, 2002).

This is also the case in Pacheco Paneque et al. (2021), where a mixed-integer linear formulation of choice models is proposed. To overcome the stochastic nature of the choice model, we simulate realizations of the probability distribution associated with its random components, called scenarios. The individual choice probabilities, and therefore the expected demand, are approximated by means of the sample-average approximation principle on the scenarios. This comes at the cost of large mixed-integer linear programming (MILP) formulations. Although small to medium-size instances can be optimally solved with general-purpose MILP solvers, practically relevant problems might involve a larger number of individuals and a considerable number of scenarios to enhance precision.

Each scenario can be seen as an independent behavioral situation where individuals consider to make a choice. These scenarios provide the problem with a decomposable structure that can be exploited to address the computational complexity of the exact method. In this paper, we propose a heuristic method inspired by scenario decomposition in the stochastic programming (SP) framework for the general choice-based optimization problem introduced in Pacheco Paneque et al. (2021). In scenario decomposition, copies of the first-stage decisions of multi-stage SP problems are introduced for each scenario. The so-called non-anticipativity constraints, which impose that these copies should be the same across scenarios, are dualized. Then, the subproblems associated with each scenario are independently solved (i.e., Lagrangian decomposition or variable splitting in combinatorial optimization). The main advantage of this approach is that the interplay between the decisions and the expected demand is preserved in the subproblems. The bound yielded by scenario decomposition can be improved by grouping scenarios at the expense of solving larger subproblems. To this end, we test different strategies based on similarity measures (i.e., Euclidean distance).

We enhance the proposed method with an algorithm that solves a restricted MILP formulation of the original problem. This formulation uses information from the solutions of the subproblems at each iteration of the subgradient method applied to solve the Lagrangian dual. The algorithm exploits the properties of the choice-based optimization problem and allows to efficiently generate feasible solutions. Computational experiments show that the proposed methodology outperforms the considered commercial solver with respect to the best feasible solution found for a given time budget. At the same time, the generation of feasible solutions at each iteration of the subgradient method enables the calculation of the duality gap. This gap allows to assess the quality of the generated solu-

tions, and therefore the performance of the method, as optimal solutions cannot be obtained for large instances.

This paper is organized as follows. Section 2 reviews the related work and discusses our main contributions. Section 3 introduces the general problem and its MILP formulation. Section 4 explains how the problem is decomposed via scenario decomposition and outlines the algorithm to generate feasible solutions. Section 5 describes the revenue maximization problem upon which the proposed methodology is tested and includes the numerical experiments. Finally, Section 6 concludes with some final remarks and future research avenues.

2 Related work

Because of the probabilistic nature of choice models, the formulations resulting from their integration in optimization models are typically non-linear and non-convex. Researchers have therefore focused on the development of reformulations and efficient algorithms to solve practical problems. In Section 2.1, we provide an overview of solution methodologies in various contexts. Recent literature on scenario decomposition and scenario grouping is reviewed in Section 2.2. To conclude, Section 2.3 outlines the main contributions and discusses the positioning of the paper.

2.1 Solution methodologies

Benati and Hansen (2002) introduce a logit-based model for the optimal location of new facilities. The resulting formulation is a hyperbolic sum integer programming (IP) model. They develop a branch-and-bound algorithm with a concave relaxation of the problem as a dual bound. Computational results show that only problems of moderate size (50 nodes) can be solved to optimality. For the same problem, Freire et al. (2016) propose a branch-and-bound algorithm that embeds a greedy algorithm to solve a relaxation of the original problem. *Outer approximation and submodular cuts for maximum capture facility location problems with random utilities* (2018) propose a branch-and-cut algorithm based on outer-approximation and submodular cuts. This approach has been recently enhanced in Mai and Lodi (2020) with a cutting-plane algorithm that requires less cuts thanks to the clustering of demand points. The last three algorithms are tested on a real instance for the location of park-and-ride facilities in New York City, with 59 candidate locations and more than 82000 customers. In general, the branch-and-cut algorithm is better in terms of number of solved instances but the cutting-plane algorithm is remarkably faster.

Gilbert et al. (2014a) consider the toll setting problem on a transportation network where users are assigned to paths according to the logit model. The resulting optimization model is non-linear and non-convex, and may have several local optima. An exhaustive algorithmic study of this problem is carried out in Gilbert et al. (2015). To solve the problem, a mixed-integer approximation scheme that provides starting points from which a local search converges to near-optimal solutions is implemented. Numerical experiments on dense circular networks (20 nodes, 90 arcs, 10 origin-destination pairs and up to 10 toll arcs) show that near-optimal solutions can be obtained. In Gilbert et al. (2014b), the path assignment is performed according to a mixed logit model, whose probabilities do not have a closed form. A similar solution method is considered to solve the problem. The large duality gaps obtained for some of the tested instances (for a given time budget) confirm the fact that the combinatorial approximations of the problems are hard to solve.

Other works consider linear reformulations of the original non-linear models. Three of such reformulations for the problem on the optimal location of new facilities are compared in Haase and Müller (2014). Numerical experiments with up to 400 customer zones and 50 candidate locations show that the linearization by Haase (2009) outperforms the other two. This formulation is later strengthened in Freire et al. (2016). *Outer approximation and submodular cuts for maximum capture facility location problems with random utilities* (2018) show that the proposed branch-and-cut algorithm outperforms the existing exact approaches, including the discussed linear reformulations. In the context of school location, Haase and Müller (2013) propose a linear IP formulation under a mixed logit model. The performed experiments show that real instances (up to 113000 students and 26 candidate locations) are solved (close) to optimality within a few minutes. Lin et al. (2020) formulate the optimal location of self-service lockers as a multi-ratio linear-fractional 0–1 programming model and provide two solution methodologies: an MILP reformulation for small-scale problems (networks with up to 100 nodes) and a quadratic transform with linear alternating algorithm for large-scale problems that outperforms the MILP approach.

Gallego et al. (2004) introduce a dynamic programming (DP) model for a choice-based general network revenue management problem. They show that its optimal value can be closely approximated by the optimal solution of an appropriately constructed linear programming (LP) model. Since then, researchers have focused on various approximations of the underpinning DP formulation, offering provable bounds on the optimal expected revenue instead of performance guarantees on the optimality gap (Strauss et al. (2018)). Yan et al. (2008) address the problem on fleet routing and flight scheduling with both stochastic market demands (via scenarios) and market shares (via a logit model). To solve the two-stage SP model, they propose two heuristic algorithms that fix a market demand

scenario and approximate the resulting mixed-integer non-linear programming (MINLP) model by fixing the decisions that are explanatory variables of the logit model.

2.2 Scenario decomposition and scenario grouping

Scenario decomposition is applied to the deterministic equivalent formulation that results from considering a finite number of scenarios of a multi-stage SP problem. Carøe and Schultz (1999) developed an algorithm for multi-stage SP problems with mixed-integer variables in all stages. They introduce copies of the first-stage variables (which do not depend on the random data) for each scenario and relax the non-anticipativity constraints in a Lagrangian manner. This method is incorporated as a bounding procedure within a branch-and-bound algorithm to achieve convergence. Numerical experiments with up to 10 scenarios show that the Lagrangian dual provides considerably better lower bounds than the LP relaxation and that feasible solutions within 0.2% of the optimum are found.

Scenarios can be gathered into groups such that a copy of the first-stage variables is introduced for each group. Escudero et al. (2013) are the first to explore such an idea. They propose a cluster-based Lagrangian decomposition procedure for the two-stage SP mixed 0-1 problem. Each cluster is a set of scenarios built at random where the non-anticipativity constraints are implicitly considered. Computational experiments on instances with up to 500 scenarios show that this technique outperforms traditional Lagrangian decomposition for single scenarios both in the quality of the bounds and computational effort. This idea has been extended to the multi-stage SP problem (Escudero et al., 2016 for the binary case, Gade et al., 2016 for the integer case). Scenario grouping has also been considered together with scenario reduction techniques in van Ackooij et al. (2018) to dynamically update the groups during the iterative solution process.

Crainic et al. (2014) evaluate multiple grouping strategies for the two-stage network design problem. They are defined by the type of splitting (cover or partition), grouping (at random or based on a similarity/dissimilarity measure inspired by the k-means clustering algorithm), and scenario characteristics according to which the similarity/dissimilarity is measured. Numerical experiments for up to 32 scenarios on instances with 10 nodes, up to 83 arcs and up to 50 commodities show that the covering strategy is the one reporting the highest quality solutions. More recently, Ryan et al. (2020) introduce an optimization problem for grouping scenarios to maximize the bound improvement. This technique provides stronger initial relaxation bounds when compared with random and k-means clustering (at the cost of an increase in computational time). The main advantage is that it can be incorporated into any general scenario decomposition algorithm as a preprocessing step.

2.3 Contributions

The works analyzed in Section 2.1 illustrate the complexity of solving choice-based optimization problems, especially for large instances. In this paper, we rely on the mixed-integer linear formulation of choice models introduced in Pacheco Paneque et al. (2021). The resulting MILP model is sensitive from a computational point of view. Indeed, the exact method fails at solving instances with a large number of scenarios, alternatives and/or individuals, which usually arise in real-life problems. To tackle such instances, we introduce a heuristic solution method that takes advantage of the simulation-based linearization in the MILP model by exploiting the principles of scenario decomposition and scenario grouping.

The proposed approach solves a general choice-based optimization problem. This problem aims at deciding on (some of) the characteristics (e.g., price) of a given set of alternatives such that an objective function determined by the planner (e.g., revenue) is optimized (see Section 3). Notice that these decisions are the same across scenarios. As seen in Section 2.2, they play the role of first-stage variables in multi-stage SP problems. To the best of our knowledge, this is the first time that a scenario-based decomposition method is used to solve a choice-based optimization problem.

Our method is also general with respect to the choice model. This is an important feature because we observe that, in most cases, the proposed methods are specific to the assumed choice model, which is typically the logit model because of the simplicity of its closed-form probability formula. However, advanced choice models like the mixed logit model overcome the main limitations of the logit model and have shown a better prediction power. We numerically test our method on a revenue maximization problem under a mixed logit model. The results show that it outperforms by a considerable margin a general-purpose MILP solver for a given time budget and that the duality gaps are as well restricted.

3 Problem definition and mathematical model

The optimization problem considered in this paper embeds a discrete choice model whose parameters have been exogenously estimated to represent the expected demand. Let N be the number of individuals in the sample (population) and J the number of alternatives that can be chosen by the individuals according to the choice model. For each alternative i , we assume a capacity denoted by $c_i \geq 1$, which indicates the maximum number of individuals who can choose it. We denote the number of scenarios by R . These scenarios are generated outside of the optimization problem (known as exterior approach in the SP framework).

In a choice model, U_{in} represents the utility that individual n obtains from

choosing alternative i . It consists of a systematic component and a random component that captures everything that has not been explicitly included. The behavioral assumption is that each individual wants to maximize their own utility. For each scenario r , we introduce a deterministic utility U_{inr} that has the same systematic component as U_{in} but replaces the random component by a realization of its probability distribution (scenario).

To formulate the problem, we denote the K decisions associated with alternative i and individual n by x_{in1}, \dots, x_{inK} . These variables are assumed to be bounded, i.e., $x_{ink} \in [a_{ink}, b_{ink}]$, $\forall i, n, k$, and might be binary. The binary variables y_{inr} represent the availability of alternatives, i.e., they are equal to 1 if the capacity of alternative i has not been reached for individual n in scenario r . The auxiliary variables z_{inr} prevent an unavailable alternative from obtaining the highest utility. They take value U_{inr} if alternative i is available to individual n in scenario r , and a given low value otherwise. The behavioral assumption of the choice model is expressed in terms of z_{inr} , i.e., individual n chooses alternative i in scenario r if $i = \operatorname{argmax}_{1 \leq j \leq J} z_{jnr}$. The variables $U_{nr} = \max_{1 \leq j \leq J} z_{jnr}$ capture the highest utility for individual n in scenario r , and the binary variables w_{inr} indicate the choice, i.e., they are equal to 1 if U_{nr} is achieved at alternative i . The expected demand of alternative i is then obtained by

$$D_i^R = \frac{1}{R} \sum_{n=1}^N \sum_{r=1}^R w_{inr}. \quad (1)$$

The objective function relates the planner's decisions to the expected demand. We denote it by $f(x, D^R)$, where x is the $J \times N \times K$ -dimensional vector containing the variables x_{ink} and D^R is the J -vector that contains the demands D_i^R . We assume it is linear in x and the choice variables. Without loss of generality, we assume $f(x, D^R)$ is to be maximized. We define Model 2 for the general choice-based optimization problem considered in this paper.

$$z^{\text{MILP}} = \max f(x, D^R) \quad (2a)$$

$$\text{s.t. } U_{inr} = \sum_{k=1}^K \beta_{ink} x_{ink} + d_{inr} \quad \forall i, n, r \quad (2b)$$

$$\ell_{nr} \leq z_{inr} \quad \forall i, n, r \quad (2c)$$

$$z_{inr} \leq \ell_{nr} + M_{inr} y_{inr} \quad \forall i, n, r \quad (2d)$$

$$U_{inr} - M_{inr}(1 - y_{inr}) \leq z_{inr} \quad \forall i, n, r \quad (2e)$$

$$z_{inr} \leq U_{inr} \quad \forall i, n, r \quad (2f)$$

$$z_{inr} \leq U_{nr} \quad \forall i, n, r \quad (2g)$$

$$U_{nr} \leq z_{inr} + M_{nr}(1 - w_{inr}) \quad \forall i, n, r \quad (2h)$$

$$\sum_i w_{inr} = 1 \quad \forall n, r \quad (2i)$$

$$w_{inr} \leq y_{inr} \quad \forall i, n, r \quad (2j)$$

$$\sum_{m=1}^n w_{imr} \leq (c_i - 1)y_{inr} + (n - 1)(1 - y_{inr}) \quad \forall i, n > c_i, r \quad (2k)$$

$$c_i(1 - y_{inr}) \leq \sum_{m=1}^n w_{imr} \quad \forall i, n > 1, r \quad (2l)$$

$$h(x) \leq 0 \quad (2m)$$

$$a_{ink} \leq x_{ink} \leq b_{ink} \quad \forall i, n, k \quad (2n)$$

$$y_{inr}, w_{inr} \in \{0, 1\} \quad \forall i, n, r \quad (2o)$$

Equations (2b) define U_{inr} as the sum of the systematic component that de-

depends on the planner’s decisions and a constant term denoted by d_{inr} . This term includes the r -th scenario associated with alternative i and individual n and other elements of the systematic component that are constant to the optimization problem. As $x_{ink}, \forall i, n, k$, is bounded and $d_{inr}, \forall i, n, r$, is constant, we can derive bounds on U_{inr} : $\ell_{inr} \leq U_{inr} \leq m_{inr}$. Constraints (2c)–(2f) provide the linear formulation of the variables z_{inr} , where $\ell_{nr} = \min_{1 \leq j \leq J} \ell_{jn}$ and $M_{inr} = m_{inr} - \ell_{nr}$. Constraints (2g)–(2h) provide the linear formulation of U_{nr} , where $m_{nr} = \max_{1 \leq j \leq J} m_{jn}$ and $M_{nr} = m_{nr} - \ell_{nr}$. Constraints (2i) ensure that one and only one alternative is chosen per individual and scenario. Constraints (2j) prevent an unavailable alternative from being chosen. Constraints (2k)–(2l) handle capacity constraints on the alternatives. We assume that the access of individuals to the alternatives for each scenario is modeled by an exogenously given priority list. This list determines the access order of individuals, i.e., if individual n does not have access to the alternative in that scenario, neither does individual $n + 1$, and consequently the upcoming individuals. Constraints (2k) forbid the access to an alternative for a scenario when its capacity has been reached, whereas constraints (2l) ensure its availability otherwise. Constraint (2m) represents the set of linear constraints that identify the requirements of x .

This mathematical model lacks efficiency for solving large instances with a general-purpose MILP solver, as showed in the numerical experiments performed in Pacheco Paneque et al. (2021). Indeed, the individual representation of the expected demand and the simulation-based linearization of the choice model, which involves the presence of the so-called big-M constraints, i.e., (2c)–(2f) and (2g)–(2h), contribute to the complexity of the formulation.

4 Scenario decomposition method

We introduce a heuristic solution method that exploits the properties of the choice-based optimization problem described in Section 3. It involves a subgradient procedure (see Section 4.3) that generates an upper bound and a feasible solution (lower bound) to Model 2 at each iteration. Section 4.1 describes the decomposition based on scenario groups and outlines the scenario grouping strategies. Section 4.2 presents the algorithm to generate feasible solutions from the Lagrangian solution.

4.1 Decomposition by scenario groups

Let S be the number of scenario groups (indexed by s) and R_s the set that contains the scenarios belonging to group s . Notice that the scenarios are partitioned into groups, i.e., each scenario belongs to one and only one group. Both S and $R_s, \forall s$,

are determined according to one of the grouping strategies described below. The copy of x_{ink} associated with group s is denoted by x_{ink}^s . Constraints (3) represent the non-anticipativity constraints imposing that the copied variables x_{ink}^s must be equal across groups. Constraints (3b) are redundant, but allow to obtain tighter upper bounds, as shown by preliminary experiments performed on the instances described in Section 5.2. Note that other characterizations of the non-anticipativity constraints are possible, such as relating all copies with each other (e.g., Escudero et al., 2013).

$$x_{ink}^s - x_{ink}^{s+1} = 0, \quad \forall i, n, k, s < S, \quad (3a)$$

$$x_{ink}^s - x_{ink}^1 = 0, \quad \forall i, n, k, s = S. \quad (3b)$$

The Lagrangian relaxation of constraints (3) with associated multipliers $\alpha_{ink}^s \in \mathbb{R}, \forall i, n, k, s$, yields independent subproblems for each group s . As the objective function $f(x, D^R)$ is linear, it can be split into the functions denoted by $f^s(x^s, D^{R,s})$ such that $f(x, D^R) = \sum_{s=1}^S f^s(x^s, D^{R,s})$, where x^s is the $J \times N \times K$ -vector containing the variables x_{ink}^s , $D^R = \sum_{s=1}^S D^{R,s}$ and $D^{R,s}$ is the J -vector containing the demands $D_i^{R,s}$:

$$D_i^{R,s} = \frac{1}{R} \sum_{n=1}^N \sum_{r \in R_s} w_{inr}. \quad (4)$$

The subproblem associated with group s is formulated in Model 5. It is essentially Model 2 for a reduced number of scenarios whose objective function is modified. As shown in Pacheco Paneque et al. (2021), the computational complexity of Model 2 grows exponentially with respect to R . Hence, for a given number of individuals, it is usually more efficient to solve multiple problems with a small number of scenarios each rather than a single problem containing multiple

scenarios. This trade-off is analyzed in Section 5.4.

$$z_s^{\text{UB}}(\alpha^s, \alpha^{s-1}) = \max f^s(x^s, D^{\mathcal{R},s}) + \sum_{i \in \mathcal{C}} \sum_{n=1}^N \sum_{k=1}^K (\alpha_{\text{ink}}^s - \alpha_{\text{ink}}^{s-1}) x_{\text{ink}}^s \quad (5a)$$

$$\text{s.t. } \mathbf{U}_{\text{inr}} = \sum_{k=1}^K \beta_{\text{ink}} x_{\text{ink}}^s + \mathbf{d}_{\text{inr}} \quad \forall i, n, r \in \mathcal{R}_s \quad (5b)$$

$$\ell_{\text{nr}} \leq z_{\text{inr}} \quad \forall i, n, r \in \mathcal{R}_s \quad (5c)$$

$$z_{\text{inr}} \leq \ell_{\text{nr}} + M_{\text{inr}} y_{\text{inr}} \quad \forall i, n, r \in \mathcal{R}_s \quad (5d)$$

$$\mathbf{U}_{\text{inr}} - M_{\text{inr}}(1 - y_{\text{inr}}) \leq z_{\text{inr}} \quad \forall i, n, r \in \mathcal{R}_s \quad (5e)$$

$$z_{\text{inr}} \leq \mathbf{U}_{\text{inr}} \quad \forall i, n, r \in \mathcal{R}_s \quad (5f)$$

$$z_{\text{inr}} \leq \mathbf{U}_{\text{nr}} \quad \forall i, n, r \in \mathcal{R}_s \quad (5g)$$

$$\mathbf{U}_{\text{nr}} \leq z_{\text{inr}} + M_{\text{nr}}(1 - w_{\text{inr}}) \quad \forall i, n, r \in \mathcal{R}_s \quad (5h)$$

$$\sum_{i \in \mathcal{C}} w_{\text{inr}} = 1 \quad \forall n, r \in \mathcal{R}_s \quad (5i)$$

$$w_{\text{inr}} \leq y_{\text{inr}} \quad \forall i, n, r \in \mathcal{R}_s \quad (5j)$$

$$\sum_{m=1}^n w_{\text{imr}} \leq (c_i - 1)y_{\text{inr}} + (n - 1)(1 - y_{\text{inr}}) \quad \forall i, n > c_i, r \in \mathcal{R}_s \quad (5k)$$

$$c_i(1 - y_{\text{inr}}) \leq \sum_{m=1}^n w_{\text{imr}} \quad \forall i, n > 1, r \in \mathcal{R}_s \quad (5l)$$

$$h(x^s) \leq 0 \quad (5m)$$

$$y_{\text{inr}}, w_{\text{inr}} \in \{0, 1\} \quad \forall i, n, r \in \mathcal{R}_s \quad (5n)$$

The Lagrangian problem associated with Model 2 is defined as follows:

$$z^{\text{UB}}(\alpha) = \sum_{s=1}^S z_s^{\text{UB}}(\alpha^s, \alpha^{s-1}). \quad (6)$$

It provides an upper bound on z^{MLP} for any values of α . Notice that α_{ink}^0 (obtained when $s = 1$) refers to $\alpha_{\text{ink}}^S, \forall i, n, k$. We consider various scenario grouping strategies that are inspired by the strategies introduced in Crainic et al. (2014). In our case, the descriptive statistics for each scenario r are given by the $J \times N$ -dimensional vector that contains the realizations associated with r .

Random (RAN) This strategy serves as a benchmark to evaluate the more refined strategies described below. To balance the number of scenarios per group, we define σ groups with $\lceil R/S \rceil$ scenarios and $S - \sigma$ groups with $\lfloor S/R \rfloor$ scenarios, where σ is the remainder in the Euclidean division of R by S .

Similar (SIM) This approach is inspired by the k-means clustering algorithm. In this case, the distance between scenarios is defined by the Euclidean distance between the associated descriptive vectors. In addition to creating the scenario groups, this strategy determines the number of groups by assuming a lower (\underline{S}) and an upper bound (\bar{S}) on S and selecting $S \in \{\underline{S}, \dots, \bar{S}\}$ such that the difference between the error associated with S and $S - 1$ is maximized. Since this strategy might generate very unbalanced groups with respect to the number of scenarios, we set $\lceil R/S \rceil$ as the maximum number of scenarios in a group. As soon as this value is reached, the group is no longer updated. The scenarios are assigned to groups according to the difference between their distance to the nearest and the farthest group. Algorithm 1 outlines the pseudocode of this strategy.

Similar without dissimilar scenarios (SIM-D) This strategy first creates *similar* groups using SIM and generates additional *dissimilar* groups that contain one scenario from each similar group. The selected scenario is the one that is closest to its mean. To balance the number of scenarios, we also set a maximum of $\lceil R/S \rceil$ scenarios per group. As soon as this value is reached, a new dissimilar group is created.

Dissimilar (DIS) As opposed to SIM, scenarios are assigned to the groups that are the farthest away from the mean. They are ranked in increasing order of the difference between the distance to the nearest and to the farthest group. The selected number of groups $S \in \{\underline{S}, \dots, \bar{S}\}$ is the one with the largest average of the distances between each group mean and the scenario closest to that mean.

Algorithm 1: Similar grouping

```
1 Initialize error_diff =  $-\infty$ , max_error_diff = error_diff,  $S^* = S$ ;  
2 for  $S \in \{\underline{S}, \dots, \bar{S}\}$  do  
3   Initialize one group with a randomly chosen scenario  $r_1$ :  $g_1 = \{r_1\}$ ,  
   mean( $g_1$ ) =  $r_1$ ;  
4   for  $s = 2 \dots S$  do  
5     Initialize  $g_s = \{r_s\}$ ,  $r_s$  selected at random with a higher probability  
     for distant scenarios with respect to  $r_{s-1}$ , mean( $g_s$ ) =  $r_s$ ;  
6   for  $r \notin \cup_{s=1}^S g_s$  do  
7     Assign  $r$  to the group with the closest mean;  
8   while groups change do  
9     Update mean( $g_s$ ) =  $r_s, \forall s$ ;  
10    for  $r = 1 \dots R$  according to distance difference do  
11      Assign  $r$  to the group with the closest mean;  
12      if  $|g_s| \geq \lceil R/S \rceil$  then  
13         $g_s$  becomes unavailable;  
14  Calculate error( $S$ ) =  $\sum_{r=1}^R \text{dist}(r, \text{mean}(g_s) | r \in g_s)$ ;  
15  if  $S > \underline{S}$  then  
16    error_diff = error( $S$ ) - error( $S - 1$ );  
17  if error_diff > max_error_diff then  
18    max_error_diff = error_diff;  
19    Update  $S^* = S$  and store the associated scenario groups  
     $g_s, \forall s = 1 \dots S^*$ ;  
20 Return  $S^*$  and the associated scenario groups  $g_s, \forall s = 1 \dots S^*$ ;
```

4.2 Generation of feasible solutions

The decomposition technique described in Section 4.1 does not necessarily yield a feasible solution to Model 2. A feasible solution provides a lower bound to the optimal solution of Model 2. It allows to compute the *duality* gap at each iteration of the subgradient method. This gap is defined according to the relative difference between the decomposition's upper bound and the generated lower bound.

To efficiently generate a feasible solution, we solve the restricted MILP formulation to Model 2 that fixes the x -variables to the values these variables obtain in the Lagrangian solution. When the x -variables are fixed, the resulting problem can be iteratively solved with Algorithm 2. It iterates over the scenarios and the individuals in the order provided by the priority list. For each individual and sce-

nario, it generates the associated choice according to the discounted utility and updates the occupancy level of the chosen alternative.

Notice that S values from the Lagrangian solution are available to each variable x_{ink} . Let $\bar{x}_{ink}^s, \forall i, n,$
 k, s , be the value associated with the copied variable x_{ink}^s . We evaluate all combinations that can be characterized from $\bar{x}_{ink}^s, \forall i, n, k, s$, as shown in Algorithm 3. This results into $S^{J \times N \times K}$ combinations, where J represents the number of alternatives, N the number of individuals and K the number of decisions. For each combination, Algorithm 2 is performed, and the feasible solution with the largest objective function value is selected. Despite Algorithm 3 being efficient (see Section 5.3), evaluating all combinations might prompt a considerable computational burden. In this case, Algorithm 3 can be simplified by only evaluating one combination per group, which generates S feasible solutions.

Algorithm 2: Solution method for Model 2 with $x_{ink} = \bar{x}_{ink}, \forall i, n, k$.

```

1 for  $r = 1 \dots R$  do
2   Initialize occupancy level  $o_{ir} = 0, \forall i$ , and availability variables
    $y_{inr} = 1, \forall i, n$ ;
3   for  $n = 1 \dots N$  do
4     for  $i = 1 \dots J$  do
5       if  $o_{ir} < c_i$  then
6         Calculate  $U_{inr} = \sum_{k=1}^K \beta_{ink} \bar{x}_{ink} + d_{inr}$  and set
            $z_{inr} = U_{inr}$ ;
7       else
8         Set the alternative unavailable:  $y_{inr} = 0$ ;
9         Set the discounted utility to the corresponding lower
           bound:  $z_{inr} = \ell_{nr}$ ;
10      Determine  $U_{nr} = \max_{\{1 \leq i \leq J | y_{inr}=1\}} z_{inr}$  and  $j = \operatorname{argmax} U_{nr}$ ;
11      Set  $w_{jnr} = 1$  and  $w_{inr} = 0, \forall i \neq \{j\}$ ;
12      Update the occupancy level  $o_{jr} = o_{jr} + 1$ ;
13 Obtain  $D^R$  by calculating  $D_i^R = \frac{1}{R} \sum_{n=1}^N \sum_{r=1}^R w_{inr}, \forall i$ ;
14 Obtain the objective function  $f(\bar{x}, D^R)$ ;

```

Algorithm 3: Method to generate feasible solutions to Model 2 with $\bar{x}_{ink}^s, \forall i, n, k, s$.

- 1 Initialize the best lower bound $z^{LB} = -\infty$;
 - 2 **for** all $S^{J \times K \times N}$ combinations from $\bar{x}_{ink}^s, \forall i, n, k, s$ **do**
 - 3 Solve Model 2 with $x_{ink}, \forall i, n, k$ fixed to the corresponding combination (\bar{x}) with Algorithm 2 and obtain $z_{current}^{LB} = f(\bar{x}, D^R)$;
 - 4 **if** $z_{current}^{LB} > z^{LB}$ **then**
 - 5 Update $z^{LB} = z_{current}^{LB}$ and keep the associated feasible solution;
 - 6 Return the best lower bound z^{LB} and the associated feasible solution;
-

4.3 Subgradient method

To obtain the best possible upper bound to Model 2, we need to solve the Lagrangian dual

$$z^{LD} = \min_{\alpha \in \mathbb{R}^{J \times N \times K \times S}} z^{UB}(\alpha). \quad (7)$$

The Lagrangian function $z^{UB}(\alpha)$ is non-differentiable. However, subgradient directions can be easily generated. The subgradient method constructs a sequence $\{\alpha^\nu\}_\nu$ using

$$\alpha^{\nu+1} = \alpha^\nu + \gamma^\nu d^\nu, \quad \forall \nu, \quad (8)$$

where ν denotes the iteration, γ^ν is a positive scalar called step size and d^ν is a vector representing the direction of motion called step direction.

The step direction d^ν can be directly defined as the subgradient direction. The $J \times N \times K \times S$ -dimensional vector v defined by $v_{ink}^s = x_{ink}^s - x_{ink}^{s+1}, \forall i, n, k, s$, and evaluated at the Lagrangian solution, is a subgradient of $z^{UB}(\alpha)$ at any value of α . Nevertheless, preliminary experiments showed that the angle between v^ν and $v^{\nu-1}$ is obtuse in multiple occasions. This leads to a next Lagrangian multiplier $\alpha^{\nu+1}$ that is close to the previous one, which slows down the convergence of the procedure. This effect is known as zigzagging of kind I, and can be overcome by deflecting the step direction (Camerini et al., 1975):

$$d^\nu = -(v^\nu + \zeta^\nu d^{\nu-1}), \quad (9)$$

where $\zeta^\nu \in \mathbb{R}_{\geq 0}$ is a suitable scalar called deflection parameter (notice the negative sign because the Lagrangian dual is to be minimized). By defining this parameter as

$$\zeta^\nu = \begin{cases} -\tau \frac{v^\nu d^{\nu-1}}{\|d^{\nu-1}\|^2} & \text{if } v^\nu d^{\nu-1} < 0, \\ 0 & \text{otherwise,} \end{cases} \quad (10)$$

with $1 \leq \tau < 2$, the step direction is forced to always form an acute angle with the preceding direction, which eliminates the zigzagging of kind I. Notice that in the absence of zigzagging of kind I (i.e., $v^\nu d^{\nu-1} \geq 0$), the step direction d^ν is equal to $-v^\nu$ (the negative subgradient). We consider the step size most commonly used in practice (Held et al., 1974):

$$\gamma^\nu = \lambda^\nu \frac{z^{\text{UB}}(\alpha^\nu) - z^{\text{LB,best}}}{\|d^\nu\|^2}, \quad (11)$$

where $\|d^\nu\|^2$ is the norm of the step direction, i.e., $\|d^\nu\|^2 = \sum_{i=1}^J \sum_{n=1}^N \sum_{k=1}^K \sum_{s=1}^S (d_{\text{ink}}^{s,\nu})^2$, and λ^ν is a step size decreasing parameter satisfying $0 < \lambda^\nu \leq 2$. Fisher (1973) suggests to halve this value whenever $z^{\text{UB}}(\alpha^\nu)$ has failed to decrease in a given number θ of consecutive iterations.

Algorithm 4 presents the pseudocode of the subgradient method. Notice that unlike the ordinary gradient method, the subgradient method is not a descent method. This is why we keep track of the best upper and lower bounds found throughout the process. We set a time budget as stopping criterion.

Algorithm 4: Deflected subgradient method.

```

1 Initialize  $v = 0$ ,  $z^{\text{UB,best}} = +\infty$ ,  $z^{\text{LB,best}} = -\infty$ ,  $\alpha^0$ ,  $\lambda^0$  and  $d^0 = 0$ ;
2 while stopping criterion do
3   for  $s = 1 \dots S$  do
4     Solve Model 5;
5     Obtain  $\bar{x}_{\text{ink}}^s$ ,  $\forall i, n, k, s$ , and  $z^{\text{UB}}(\alpha^\nu)$ ;
6     if  $z^{\text{UB}}(\alpha^\nu) < z^{\text{UB,best}}$  then
7       Update  $z^{\text{UB,best}} = z^{\text{UB}}(\alpha^\nu)$ ;
8     Run Algorithm 3 and obtain  $z^{\text{LB}}$ ;
9     if  $z^{\text{LB}} > z^{\text{LB,best}}$  then
10      Update  $z^{\text{LB,best}} = z^{\text{LB}}$ ;
11     if  $z_{\text{UB}}(\alpha^\nu)$  has not improved in the last  $\theta$  consecutive iterations then
12       Set  $\lambda^\nu \leftarrow \lambda^\nu / 2$ ;
13     Calculate  $\gamma^\nu$  according to (11);
14     Update  $\alpha_{\text{ink}}^{s,\nu+1} = \alpha_{\text{ink}}^{s,\nu} + \gamma^\nu d_{\text{ink}}^{s,\nu}$ ,  $\forall i, n, k, s$ , where
         $d_{\text{ink}}^{s,\nu} = -(v_{\text{ink}}^{s,\nu} + \zeta^\nu d_{\text{ink}}^{s,\nu-1})$  and  $\zeta^\nu$  is calculated according to (10);
15     Set  $v \leftarrow v + 1$ ;
16     Check stopping criterion;

```

5 Computational experiments

The goal of the experiments carried out in this section is threefold. First, we evaluate the performance of the decomposition method against a general-purpose MILP solver for a given time budget and compare the different scenario grouping strategies against each other (Section 5.3). Second, we assess the trade-off between the solution quality and duality gap and the number of scenarios per group (Section 5.4). Third, we analyze the impact in the algorithm’s performance of scaling up the problem size by increasing the dimensions not involved in the decomposition (Section 5.5 for an increase in the number of alternatives and Section 5.6 for an increase in the number of individuals). Before that, Section 5.1 characterizes the revenue maximization problem. This choice-based optimization problem yields a concrete MILP formulation that we use to run the experiments. Section 5.2 describes the case study and the calibration of parameters for the sub-gradient method.

5.1 Revenue maximization problem

The planner aims at finding the best pricing strategy in order to maximize its revenue by offering services to a market. Each service in the set of services \mathcal{C} has a given associated capacity. The market is composed of N individuals, which are assumed to be heterogeneous and price elastic. In a revenue maximization context, we need to model competition so that individuals are not captive. To this end, we incorporate an opt-out option into the model to capture individuals leaving the market, either because they choose a competitor’s service or because they do not choose anything at all. The opt-out option is denoted by $i = 0$ and is always available to all individuals.

We assume that the price is the only planner’s decision. We define $p_i \in [a_i, b_i]$ as the price to be paid to access service $i \in \mathcal{C}$. The expected revenue obtained from the services in \mathcal{C} is calculated as

$$f(p, D^R) = \sum_{i \in \mathcal{C}} p_i D_i^R = \frac{1}{R} \sum_{i \in \mathcal{C}} \sum_{n=1}^N \sum_{r=1}^R p_i w_{inr}. \quad (12)$$

Model 13 comprises the MILP formulation of the revenue maximization problem. Notice that the product of the price (continuous variable) and the choice (binary variable) can be linearized because bounds on the former are assumed. We introduce the variables $\eta_{inr} = p_i w_{inr}$ to capture the product of the two, with linearizing constraints (13o)–(13r). We denote by $\bar{\mathcal{C}} = \mathcal{C} \cup \{0\}$ the set containing all services and the opt-out option. The utilities associated with the opt-out option for each

individual and scenario are included in (13b), and since the opt-out option is always available, we add constraints (13h) to enforce the availability variables y_{inr}

to be equal to 1 when $i = 0$.

$$z^{\text{MILP}} = \max \frac{1}{R} \sum_{i \in \mathcal{C}} \sum_{n=1}^N \sum_{r=1}^R \eta_{inr} \quad (13a)$$

$$\text{s.t. } \mathbf{U}_{0nr} = \mathbf{d}_{0nr} \quad \forall n, r \quad (13b)$$

$$\mathbf{U}_{inr} = \beta_{in} \mathbf{p}_i + \mathbf{d}_{inr} \quad \forall i \in \mathcal{C}, n, r \quad (13c)$$

$$\ell_{nr} \leq z_{inr} \quad \forall i \in \bar{\mathcal{C}}, n, r \quad (13d)$$

$$z_{inr} \leq \ell_{nr} + M_{inr} y_{inr} \quad \forall i \in \bar{\mathcal{C}}, n, r \quad (13e)$$

$$\mathbf{U}_{inr} - M_{inr}(1 - y_{inr}) \leq z_{inr} \quad \forall i \in \bar{\mathcal{C}}, n, r \quad (13f)$$

$$z_{inr} \leq \mathbf{U}_{inr} \quad \forall i \in \bar{\mathcal{C}}, n, r \quad (13g)$$

$$y_{0nr} = 1 \quad \forall n, r \quad (13h)$$

$$z_{inr} \leq \mathbf{U}_{nr} \quad \forall i \in \bar{\mathcal{C}}, n, r \quad (13i)$$

$$\mathbf{U}_{nr} \leq z_{inr} + M_{nr}(1 - w_{inr}) \quad \forall i \in \bar{\mathcal{C}}, n, r \quad (13j)$$

$$\sum_{i \in \bar{\mathcal{C}}} w_{inr} = 1 \quad \forall n, r \quad (13k)$$

$$w_{inr} \leq y_{inr} \quad \forall i \in \bar{\mathcal{C}}, n, r \quad (13l)$$

$$\sum_{m=1}^n w_{imr} \leq (c_i - 1)y_{inr} + (n - 1)(1 - y_{inr}) \quad \forall i \in \mathcal{C}, n > c_i, r \quad (13m)$$

$$c_i(1 - y_{inr}) \leq \sum_{m=1}^n w_{imr} \quad \forall i \in \mathcal{C}, n > 1, r \quad (13n)$$

$$\mathbf{a}_{in} w_{inr} \leq \eta_{inr}, \quad \forall i \in \mathcal{C}, n, r, \quad (13o)$$

$$\eta_{inr} \leq \mathbf{b}_{in} w_{inr}, \quad \forall i \in \mathcal{C}, n, r, \quad (13p)$$

$$\mathbf{p}_i - (1 - w_{inr})\mathbf{b}_{in} \leq \eta_{inr}^{19}, \quad \forall i \in \mathcal{C}, n, r, \quad (13q)$$

$$\eta_{inr} \leq \mathbf{p}_i - (1 - w_{inr})\mathbf{a}_{in}, \quad \forall i \in \mathcal{C}, n, r, \quad (13r)$$

$$y_{inr}, w_{inr} \in \{0, 1\} \quad \forall i \in \bar{\mathcal{C}}, n, r \quad (13s)$$

5.2 Experimental setting

We consider the case study on parking services in a park-and-ride context used in Pacheco Paneque et al. (2021). The choice model for parking choices (mixed logit model) is specified and estimated in Ibeas et al. (2014). They interview 197 individuals to model their preferences with respect to three parking services: paid on-street parking (PSP), paid parking in an underground car park (PUP) and free on-street parking (FSP). We assume that FSP represents the opt-out option because it does not provide any revenue to the parking manager.

For the integration of the choice model in Model 13, we replace $\beta_{in}, \forall i \in \bar{C}, n$, with their estimates, and we compute $d_{inr}, \forall i \in \bar{C}, n, r$, with the values of the other explanatory variables of the choice model in the data and the scenario ξ_{inr} . We determine the capacity associated with PSP and PUP (the opt-out option, FSP, is assumed to have unlimited capacity) according to the number of individuals in the instance, so that it is appropriate for the sample under consideration but restrictive enough so that some users are forced to choose FSP because PSP and/or PUP become unavailable. For the sake of simplicity, we assume $c_{PSP} = c_{PUP}$, and denote it simply by c . We assume lower price bounds for PSP than PUP, i.e., $p_{PSP} \in [0.5, 0.65]$ and $p_{PUP} \in [0.7, 0.85]$.

The instances used in the upcoming sections are characterized from the available data by setting specific values for N , R and $J = |\bar{C}|$. More precisely, for each configuration of these parameters, we generate various instances by randomly selecting N individuals from the whole dataset, together with the associated R scenarios for the J parking services. The configurations are labeled as $NX_{RY}_{JZ}_{cT}$, where X indicate the number of individuals, Y the number of scenarios, Z the number of services and T the capacity. After performing some tests on different settings for the subgradient method, we resolve to initialize the Lagrangian multipliers to 0, and we consider $\lambda^0 = 0.5$, $\theta = 5$ and $\tau = 1.5$. The code is implemented in C++ using ILOG Concert Technology to access CPLEX 12.8, and all the instances were run using 12 threads in a 3.33 GHz Intel Xeon X5680 server running a 64-bit Ubuntu 16.04.2.

5.3 Performance and scenario grouping strategies

In this experiment, we compare the results obtained with the decomposition method for a given time budget against the exact approach while evaluating the different grouping strategies. We consider two configurations: $N50_{R100}_{J3}_{c20}$ and $N50_{R200}_{J3}_{c20}$. As shown in Pacheco Paneque et al. (2021), very large values of R are not required to ensure stability of the obtained quantities (e.g., revenue, prices). Furthermore, we need to restrict the number of scenarios to be able to solve to optimality Model 13 with CPLEX. Nevertheless, the decomposition

method can be applied for larger values of R . Note that in this case a larger time budget might be considered to ensure that the algorithm runs for a few iterations.

To increase statistical significance, we allow for 10 instances of each configuration (see Table 1 for computational details). Furthermore, we determine 10% and 25% of the average computational time (reported by CPLEX) as two time budget values to evaluate the performance of the algorithm at different stages of the search. Notice that the computational time of an iteration of the subgradient method is unknown before being executed. At every iteration, we compute an average computational time per iteration to avoid exceeding the time budget. The subgradient method is then terminated as soon as the expected computational time, i.e., the sum of the current computational time and the average time per iteration, is larger than the time budget.

Table 1: Computational details on the tested configurations (10 instances per configuration)

Configuration	N	R	J	c	CPLEX time (min)			Time budget (min)	
					Average	Lowest	Highest	10%	25%
N50_R100_J3_c20	50	100	3	20	191.4	112.8	318.2	19.1	47.9
N50_R200_J3_c20	50	200	3	20	1038.8	725.2	1508.0	103.9	259.7

Given the size of the instances, we consider scenario groups that have approximately five scenarios each. We could consider a larger number of scenarios per group while still being able to solve the subproblems within the given time budget at the expense of performing a smaller number of iterations. In the case of random grouping, we define S such that the number of scenarios per group does not exceed five, i.e., $S = \lceil R/5 \rceil$. For the other strategies, we define the lower and upper bound on S as $\underline{S} = \lceil R/5 \rceil - 2$ and $\bar{S} = \lceil R/5 \rceil + 2$, respectively (see Section 4.1). For example, when $R = 100$, $S \in \{18, 19, 20, 21, 22\}$. Table 2 summarizes the computational results obtained for the four grouping strategies and the two time budget values. We report the average, lowest and highest values of the duality and optimality gaps over the 10 instances, which are calculated as follows:

$$\text{gap}_{\text{opt}} = \frac{z^{\text{MILP}} - z^{\text{LB,best}}}{z^{\text{RM}}}, \quad (14a)$$

$$\text{gap}_{\text{dual}} = \frac{z^{\text{UB,best}} - z^{\text{LB,best}}}{z^{\text{LB,best}}}. \quad (14b)$$

Furthermore, we include the results on the best feasible solution found by CPLEX for the given time budget. Note that in this case the duality gap is replaced by the gap reported by CPLEX, which is computed as the relative difference between the

Table 2: Performance by scenario grouping strategies and comparison with the best feasible solution provided by CPLEX for a given time budget (10 instances per configuration)

	10% time budget					25% time budget				
	RAN	SIM	SIM-D	DIS	CPLEX	RAN	SIM	SIM-D	DIS	CPLEX
N50_R100_J3_c20										
Avg. time (min)	15.26	17.81	17.84	16.02	19.16	45.77	42.84	44.99	44.73	47.90
Avg. UB time per iter. (min)	7.61	7.60	5.05	6.79	-	7.75	7.90	4.75	6.55	-
Avg. LB time per iter. (s)	0.09	0.10	0.14	0.11	-	0.09	0.10	0.14	0.11	-
Avg. num. iter.	2.1	2.4	3.6	2.4	-	6.0	5.6	9.8	7.1	-
Avg. iter. best UB is reached	2.1	2.4	2.9	2.3	-	5.1	5	6.6	5.7	-
Avg. iter. best LB is reached	1.6	2	1.9	1.7	-	3.4	3.1	4.7	3.1	-
Avg. S	20	20.3	24.6	21.9	-	20	20.3	24.6	21.9	-
gap _{opt} (%)										
Average	0.08	0.07	0.05	0.10	5.77	0.05	0.05	0.04	0.08	0.10
Lowest	0.00	0.00	0.00	0.05	2.35	0.00	0.00	0.00	0.02	0.25
Highest	0.16	0.12	0.10	0.16	8.78	0.12	0.12	0.08	0.13	3.91
gap _{dual} (%)										
Average	1.88	1.85	1.93	2.03	25.69 ¹	1.62	1.65	1.74	1.73	16.48 ¹
Lowest	1.52	1.55	1.47	1.63	20.25 ¹	1.29	1.25	1.42	1.42	11.46 ¹
Highest	2.25	2.10	2.30	2.31	30.26 ¹	1.97	1.98	2.03	2.03	20.16 ¹
N50_R200_J3_c20										
Avg. time (min)	98.47	99.35	100.4	94.79	103.9	251.0	253.7	256.3	249.1	259.6
Avg. UB time (min)	16.35	16.79	11.67	18.90	-	15.02	16.12	8.69	16.32	-
Avg. LB time (s)	0.79	0.80	1.18	0.78	-	0.77	0.77	1.10	0.73	-
Avg. num. iter.	6.1	6.0	8.7	5.3	-	16.9	15.9	29.9	15.7	-
Avg. iter. best UB is reached	5.6	5.6	7.8	4.3	-	15.3	14	28.7	13.8	-
Avg. iter. best LB is reached	2.6	4.6	4.5	3.2	-	6.5	6.2	8.7	7.8	-
Avg. S	40	40.2	48.4	39.5	-	40	40.2	48.4	39.5	-
gap _{opt} (%)										
Average	0.03	0.03	0.03	0.03	1.46	0.02	0.02	0.02	0.03	0.54
Lowest	0.01	0.00	0.01	0.02	0.22	0.00	0.00	0.01	0.02	0.08
Highest	0.05	0.06	0.04	0.05	4.16	0.04	0.03	0.04	0.04	1.83
gap _{dual} (%)										
Average	1.82	1.82	2.03	1.84	19.05 ¹	1.62	1.64	1.73	1.64	16.71 ¹
Lowest	1.53	1.57	1.54	1.62	17.43 ¹	1.36	1.38	1.38	1.46	13.05 ¹
Highest	2.07	2.13	2.39	2.13	22.57 ¹	1.85	1.90	2.00	1.83	19.02 ¹

¹ In this case, gap_{dual} corresponds to the gap reported by CPLEX (i.e., relative difference between the best integer solution and the solution associated with the best node of the branch-and-bound tree).

best integer solution and the solution associated with the best node of the branch-and-bound tree.

We observe that the average computational time for any grouping strategy is lower than the time budget because of the above-mentioned stopping criterion. The average number of scenario groups S is larger in SIM-D because more scenario groups are created in addition to the ones generated by SIM. Even though more subproblems are solved, each of them has in average less scenarios, which results in a lower average UB time per iteration. This has also an impact on the average LB time, as more price combinations are evaluated. In any case, the algorithm to generate feasible solutions is much faster compared to solving the Lagrangian subproblem. Notice that more iterations are performed for $R = 200$ than $R = 100$ within each time budget. This is due to the fact that the time budget values are obtained from the average computational time of the exact method, that does not increase linearly with respect to R (see Table 1). In the case of $R = 200$, we observe a decrease both in the average UB time and LB time per iteration from 10% to 25% time budget. These results indicate that the subproblems and the restricted MILP formulations are more rapidly solved in the long run. We also notice that the best LB is reached earlier than the best UB. This means that the method is able to find high-quality feasible solutions at a relatively early stage whereas the best UB gets refined throughout the iterations. Thus, as long as the duality gap is low, the subgradient method can be terminated after a small number of iterations without improvement of the best LB.

The optimality gaps are very low for any strategy (below 0.16% in all cases), and they decrease as the time budget increases. We notice that they are lower on average for $R = 200$ than $R = 100$ because of the larger time budget values. On the contrary, the optimality gaps with respect to the best feasible solutions provided by CPLEX are much higher and more dispersed. This is especially the case for $R = 100$ and a 10% time budget, where the optimality gap fluctuates between 2.35% and 8.78% with an average value equal to 5.77%. The duality gaps are larger and of the same order of magnitude across grouping strategies (they are below 2.31% in all cases). As expected, the average duality gaps decrease as the time budget increases by a relatively low margin. For instance, for N50_R200_J3_c20 and RAN, it decreases from 1.82% for a 10% time budget to 1.62% for a 25% time budget. This reveals the slow convergence behavior of the subgradient method.

Regarding the grouping strategies, we observe an impact on the obtained results but we do not identify a clear winner. This is statistically supported by the ANOVA test, that evaluates whether optimality and duality gap means are equal across strategies. This is indeed the case, and post hoc tests to evaluate these means for each pair of strategies show that there is no significant difference between them. In general, DIS (dissimilar) reports the largest optimality and duality gaps. As opposed to Crainic et al. (2014), the obtained results are not superior to

RAN (random). They also show that SIM-D (similar without dissimilar scenarios) often leads to higher quality solutions at the expense of longer computational times. In our case, this strategy provides the smallest optimality gaps for $R = 100$ but not for $R = 200$. With respect to the duality gap, other strategies yield lower values. As analyzed in Section 5.4, this has to do with the fact that the subproblems have on average less scenarios, and despite the method is able to perform more iterations, it does not contribute to reducing the duality gap. RAN and SIM (similar) report reasonable duality gaps for both values of R . SIM reports a lower average optimality gap for the 10% time budget, whereas RAN reports lower values for the 25% time budget. Therefore, we cannot conclude that any of the refined grouping strategies outperforms random grouping. RAN is easier to implement and shows slightly faster solution times with relatively lower duality gaps.

5.4 Size of the scenario groups

To analyze the trade-off between the size of the scenario groups and the performance of the method, we consider various values of S and run the configuration N50_R200_J3_c20 (5 instances) for a 2-hour time budget. Table 3 includes the obtained results for groups with 1-5, 10, 15, 20 and 25 scenarios each. As expected, the larger the number of scenarios per group, the larger the average UB time per iteration, resulting in a lower average number of iterations within the time budget.

Concerning the optimality gap, we notice an increasing trend in the obtained values as S decreases. Indeed, as the number of scenarios per group increases, the number of subproblems to solve decreases, and thus less price combinations are evaluated. Nevertheless, the average optimality gaps remain below 0.06% in all cases. The decrease in S has a more noticeable impact on the duality gaps, which decrease as S decreases. Figure 1 shows for each instance the evolution of the best UB obtained with respect to the number of scenarios per group. Indeed, we observe a decrease in their values as the number of scenarios per group increases.

In the case of $S = 200$ (1 scenario per group), even though the method performs on average 101.2 iterations, it can only reach an average duality gap of 3.67%. For larger groups (i.e., 15, 20 and 25 scenarios per group), the algorithm runs for a very limited number of iterations, but the resulting duality gaps are below 1.02% in average. We observe that despite the lower number of iterations and combinations for the generation of feasible solutions, it is beneficial to define subproblems with a large number of scenarios per group as long as they can be efficiently solved within the considered time budget. Notice that we might need to consider groups of lower size in other settings that involve more individuals, more scenarios, a lower time budget, etc.

Table 3: Impact of the size of scenario groups for the configuration N50_R200_J3_c20 with a 2-hour time budget (5 instances)

	Number of scenario groups (S)								
	200	100	66	50	40	20	13	10	8
Number of scenarios per group	1	2	3 ¹	4	5	10	15 ²	20	25
Avg. time (min)	119.2	118.6	116.9	116.1	110.7	109.5	96.0	85.8	72.1
Avg. UB time per iter. (min)	0.86	2.78	4.60	7.83	12.41	32.62	43.97	56.05	72.14
Avg. LB time per iter. (s)	19.13	4.75	2.05	1.15	0.75	0.18	0.08	0.05	0.03
Avg. num. iter.	101.2	41.6	25.8	15	9	3.4	2.2	1.6	1
Avg. iter. best UB is reached	100	41.2	24.2	14.2	8.2	3	3	1.4	1
Avg. iter. best LB is reached	7.8	7.4	8.8	8.4	3.6	1	1	1	1
gap_{opt} (%)									
Average	0.02	0.01	0.02	0.03	0.03	0.04	0.04	0.06	0.06
Lowest	0.00	0.00	0.00	0.02	0.01	0.03	0.00	0.03	0.03
Highest	0.04	0.03	0.04	0.05	0.05	0.06	0.08	0.08	0.09
gap_{dual} (%)									
Average	3.67	2.80	2.30	2.02	1.83	1.31	0.99	0.90	0.78
Lowest	3.56	2.68	2.12	1.85	1.73	1.22	0.87	0.83	0.61
Highest	3.84	2.97	2.38	2.18	1.96	1.38	1.09	1.02	0.98

¹ 2 groups have 4 scenarios.

² 5 groups have 16 scenarios.

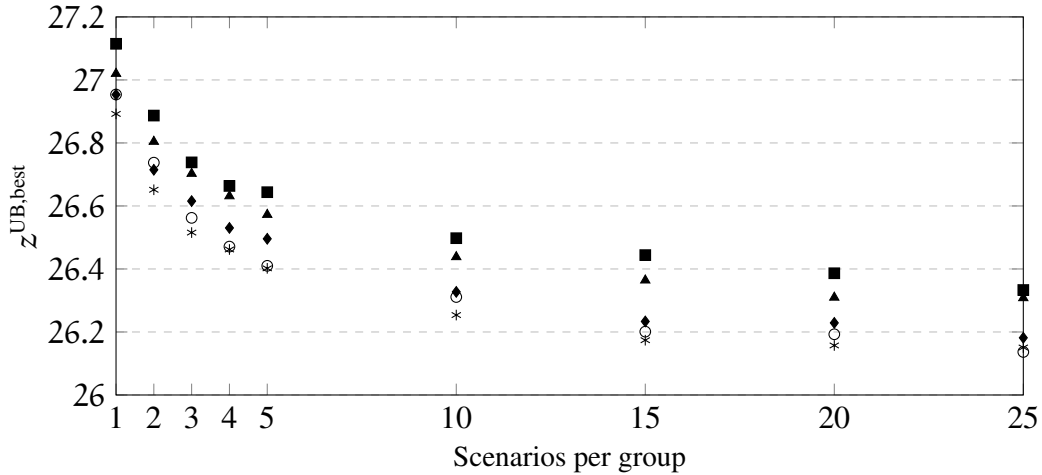


Figure 1: Best upper bounds ($z^{\text{UB,best}}$) across the scenario groups (each shape represents a different instance)

5.5 Impact of the number of alternatives

We now evaluate the performance of the method when the problem is scaled up by increasing the number of services J . To do so, we segment the prices associated with PSP and PUP by residency, which is one of the explanatory variables of the choice model. This means that different prices are proposed to the residents in the study area and to non-residents. It can be seen as extending \bar{C} from $J = 3$ to $J = 5$ services where residents have only access to PSP and PUP associated with the resident price (and the opt-out option FSP) and analogously for non-residents. Since non-residents are willing to pay larger prices than residents, we modify the price bounds as follows: $p_{\text{PSP}}^{\text{res}}, p_{\text{PSP}}^{\text{non-res}} \in [0.5, 0.75]$ and $p_{\text{PUP}}^{\text{res}}, p_{\text{PUP}}^{\text{non-res}} \in [0.75, 1.0]$.

Table 4 shows the results for three configurations (5 instances) for a 2-hour time budget. Due to the complexity induced by the increase in J , solely the MILP formulation for N50_R25_J5_c20 could be solved to optimality in a reasonable time (18.72 h on average). We therefore consider 3 scenarios per group in the decomposition method. The average UB time per iteration is much larger for $J = 5$. For instance, it increases from 7.8 min for N50_R100_J3_c20 to 34.4 min for N50_R100_J5_c20. The average LB time per iteration is also higher but remains below 0.3 s for all configurations. The optimality gaps are larger than the ones obtained for $J = 3$ in Section 5.3 but they remain below 0.5%. As expected, the duality gaps are also larger and they increase as R increases for the given time budget.

Table 4: Impact of the size of scenario groups for the configuration N50_R200_J3_c20 with a 2-hour time budget (5 instances)

	N50_R25_J5_c20	N50_R50_J5_c20	N50_R100_J5_c20
Avg. time (min)	116.8	110.9	114.9
Avg. UB time per iter. (min)	7.61	20.03	34.38
Avg. LB time per iter. (s)	0.00	0.03	0.28
Avg. num. iter.	16.2	5.8	3.4
Avg. iter. best UB is reached	13.8	4.8	1
Avg. iter. best LB is reached	4.2	3.4	1.4
S	8	16	33
<hr/>			
gap _{opt} (%)			
Average	0.38	N/A ¹	N/A ¹
Lowest	0.21	N/A ¹	N/A ¹
Highest	0.49	N/A ¹	N/A ¹
<hr/>			
gap _{dual} (%)			
Average	3.34	4.57	5.19
Lowest	2.79	3.72	4.69
Highest	3.91	5.09	5.59

¹ Instances could not be solved to optimality with CPLEX in a reasonable time.

5.6 Impact of the number of individuals

In this experiment we test the method on configurations with larger values of N , i.e., $N \in \{50, 100, 150, 197\}$, with capacities $c \in \{20, 40, 60, 80\}$, respectively. Table 5 shows the results for these four configurations (5 instances) with $R = 100$ and $J = 3$ for a 2-hour time budget. We consider scenario groups of 2 scenarios each. The average UB time per iteration grows exponentially. In the case of $N = 197$, it even exceeds the 2-hour time budget (the first iteration is always performed). The average number of iterations follows the opposite trend, as it decreases in average from 98.2 iterations for $N = 50$ to a single iteration for $N = 197$. Interestingly, we observe that the duality gaps decrease as N increases.

Table 5: Impact of an increase in the number of individuals for a 2-hour time budget (5 instances per configuration)

	N50_R100_J3_c20	N100_R100_J3_c40	N150_R100_J3_c60	N197_R100_J3_c80
Avg. time (min)	119.5	110.2	98.2	193.2 ¹
Avg. UB time per iter. (min)	1.21	15.11	61.58	193.18
Avg. LB time per iter. (s)	0.59	1.21	1.80	2.89
Avg. num. iter.	98.2	7.4	1.6	1
Avg. iter. best UB is reached	96.6	6	1.6	1
Avg. iter. best LB is reached	18.2	3.6	1.2	1
S	50	50	50	50
<hr/>				
gap _{opt} (%)				
Average	0.03	N/A ²	N/A ²	N/A ²
Lowest	0.00	N/A ²	N/A ²	N/A ²
Highest	0.07	N/A ²	N/A ²	N/A ²
<hr/>				
gap _{dual} (%)				
Average	2.38	1.98	1.91	1.56
Lowest	2.05	1.91	1.75	1.49
Highest	2.63	2.05	2.25	1.63

¹ The first iteration of the solution method is always performed (even if it exceeds the given time budget).

² Instances could not be solved to optimality with CPLEX in a reasonable time.

6 Conclusions

In this work, we introduce a tailored heuristic solution approach for choice-based optimization based on scenario decomposition and scenario grouping. It is built upon the choice-based optimization model introduced in Pacheco Paneque et al. (2021). This decomposition strategy allows to preserve the supply-demand interplay that is explicitly captured by the model because all the constraints from the original optimization problem are also included in the Lagrangian subproblem. We also develop an algorithm to efficiently generate feasible solutions to the original problem from the solution of the Lagrangian subproblem. To gather the scenarios into groups, we adapt several strategies from Crainic et al. (2014).

The grouping strategies are implemented as a preprocessing step of the subgradient method, which iteratively generates upper bounds (by solving the Lagrangian subproblem) and lower bounds (by generating feasible solutions) to the optimization problem.

To experimentally test the Lagrangian decomposition scheme, we characterize a revenue maximization problem for a case study on parking choices. We show that near-optimal solutions can be obtained for all scenario grouping strategies and various values of R in a much lower computational time in comparison with the exact method. Such solutions are usually generated at an early stage of the subgradient method, which allows to promptly terminate the algorithm if the best feasible solution has not improved after a certain number of consecutive iterations (provided that the duality gap is below an acceptable threshold). We cannot identify a scenario grouping strategy that outperforms the others, so we resolve to rely on the random partition for the performed experiments. As long as the Lagrangian subproblems are computationally manageable, a large number of scenarios per group is recommended, as it leads to smaller duality gaps for a given computational time. We also observe low duality gaps for configurations with a larger number of alternatives and individuals.

In conclusion, Lagrangian decomposition provides a relevant scheme to address the tractability of choice-based optimization problems. Additionally, the proposed method could be combined with other decomposition techniques according to the model requirements. For instance, if the capacity of the alternatives is included as a decision variable (that is formulated with a reduced set of binary variables), Benders decomposition could be as well explored. Furthermore, parallelization routines that allow to solve multiple subproblems simultaneously could be implemented within the subgradient method in order to reduce the total computational time. These approaches allow to further enhance and expand the applicability of the method to the application areas discussed in Section 1, such as facility location, revenue management and transportation-related problems.

Acknowledgements

This research is supported by the Swiss National Science Foundation [Grant 200021-165636 “Incorporating advanced behavioral models in mixed integer linear optimization”]. The work of Bernard Gendron is supported by the Natural Sciences and Engineering Research Council of Canada [Discovery Grant 2017-06054 “Large-Scale Combinatorial and Network Optimization”]. The authors would like to thank the authors in Ibeas et al. (2014), who provided the dataset employed in the performed experiments.

References

- Andersson, S.-E. (1998). Passenger choice analysis for seat capacity control: a pilot project in scandinavian airlines, *International Transactions in Operational Research* **5**(6): 471 – 486.
- Benati, S. (1999). The maximum capture problem with heterogeneous customers, *Computers & Operations Research* **26**(14): 1351 – 1367.
URL: [http://dx.doi.org/10.1016/S0305-0548\(99\)00040-4](http://dx.doi.org/10.1016/S0305-0548(99)00040-4)
- Benati, S. and Hansen, P. (2002). The maximum capture problem with random utilities: Problem formulation and algorithms, *European Journal of Operational Research* **143**(3): 518–530.
- Camerini, P. M., Fratta, L. and Maffioli, F. (1975). On improving relaxation methods by modified gradient techniques, *Nondifferentiable optimization*, Springer, pp. 26–34.
- Carøe, C. C. and Schultz, R. (1999). Dual decomposition in stochastic integer programming, *Operations Research Letters* **24**(1-2): 37–45.
- Crainic, T. G., Hewitt, M. and Rei, W. (2014). Scenario grouping in a progressive hedging-based meta-heuristic for stochastic network design, *Computers & Operations Research* **43**: 90–99.
- de Palma, A., Kilani, M. and Lindsey, R. (2005). Congestion pricing on a road network: A study using the dynamic equilibrium simulator metropolis, *Transportation Research Part A: Policy and Practice* **39**(7): 588 – 611.
- Escudero, L. F., Garín, M. A., Pérez, G. and Unzueta, A. (2013). Scenario cluster decomposition of the lagrangian dual in two-stage stochastic mixed 0–1 optimization, *Computers & Operations Research* **40**(1): 362–377.
- Escudero, L. F., Garín, M. A. and Unzueta, A. (2016). Cluster lagrangean decomposition in multistage stochastic optimization, *Computers & Operations Research* **67**: 48–62.
- Fisher, M. L. (1973). Optimal solution of scheduling problems using lagrange multipliers: Part i, *Operations Research* **21**(5): 1114–1127.
- Freire, A. S., Moreno, E. and Yushimito, W. F. (2016). A branch-and-bound algorithm for the maximum capture problem with random utilities, *European journal of operational research* **252**(1): 204–212.

- Gade, D., Hackebeil, G., Ryan, S. M., Watson, J.-P., Wets, R. J.-B. and Woodruff, D. L. (2016). Obtaining lower bounds from the progressive hedging algorithm for stochastic mixed-integer programs, *Mathematical Programming* **157**(1): 47–67.
- Gallego, G., Iyengar, G., Phillips, R. and Dubey, A. (2004). Managing flexible products on a network, *Available at SSRN 3567371* .
- Gilbert, F., Marcotte, P. and Savard, G. (2014a). Logit network pricing, *Computers & operations research* **41**: 291–298.
- Gilbert, F., Marcotte, P. and Savard, G. (2014b). Mixed-logit network pricing, *Computational Optimization and Applications* **57**(1): 105–127.
- Gilbert, F., Marcotte, P. and Savard, G. (2015). A numerical study of the logit network pricing problem, *Transportation Science* **49**(3): 706–719.
- Haase, K. (2009). Discrete location planning, *Technical Report ITLS-WP-09-07*, Institute of Transport and Logistics Studies, University of Sydney, Australia.
- Haase, K. and Müller, S. (2013). Management of school locations allowing for free school choice, *Omega* **41**(5): 847–855.
- Haase, K. and Müller, S. (2014). A comparison of linear reformulations for multinomial logit choice probabilities in facility location models, *European Journal of Operational Research* **232**(3): 689–691.
- Held, M., Wolfe, P. and Crowder, H. P. (1974). Validation of subgradient optimization, *Mathematical programming* **6**(1): 62–88.
- Ibeas, A., dell’Olio, L., Bordagaray, M. and Ortúzar, J. (2014). Modelling parking choices considering user heterogeneity, *Transportation Research Part A: Policy and Practice* **70**(C): 41 – 49.
- Lin, Y. H., Wang, Y., He, D. and Lee, L. H. (2020). Last-mile delivery: Optimal locker location under multinomial logit choice model, *Transportation Research Part E: Logistics and Transportation Review* **142**: 102059.
- Mai, T. and Lodi, A. (2020). A multicut outer-approximation approach for competitive facility location under random utilities, *European Journal of Operational Research* **284**(3): 874–881.
- Outer approximation and submodular cuts for maximum capture facility location problems with random utilities* (2018). *European Journal of Operational Research* **266**(1): 46 – 56.

- Pacheco Paneque, M., Bierlaire, M., Gendron, B. and Sharif Azadeh, S. (2021). Integrating advanced discrete choice models in mixed integer linear optimization, *Transportation Research Part B: Methodological* **146**: 26–49.
- Prokhorchuk, A., Dauwels, J. and Jaillet, P. (2019). Stochastic dynamic pricing for same-day delivery routing, *arXiv preprint arXiv:1912.02946* .
- Ryan, K., Ahmed, S., Dey, S. S., Rajan, D., Musselman, A. and Watson, J.-P. (2020). Optimization-driven scenario grouping, *INFORMS Journal on Computing* .
- Shen, Z.-J. M. and Su, X. (2007). Customer behavior modeling in revenue management and auctions: A review and new research opportunities, *Production and operations management* **16**(6): 713–728.
- Strauss, A. K., Klein, R. and Steinhardt, C. (2018). A review of choice-based revenue management: Theory and methods, *European Journal of Operational Research* **271**(2): 375–387.
- Talluri, K. T. and Van Ryzin, G. J. (2004). Revenue management under a general discrete choice model of consumer behavior, *Management Science* **50**(1): 15–33.
- Ulmer, M. W. (2020). Dynamic pricing and routing for same-day delivery, *Transportation Science* **54**(4): 1016–1033.
- van Ackooij, W., de Oliveira, W. and Song, Y. (2018). Adaptive partition-based level decomposition methods for solving two-stage stochastic programs with fixed recourse, *Informatics Journal on Computing* **30**(1): 57–70.
- Yan, S., Tang, C.-H. and Fu, T.-C. (2008). An airline scheduling model and solution algorithms under stochastic demands, *European Journal of Operational Research* **190**(1): 22–39.