Journal of Choice Modelling 39 (2021) 100285

Contents lists available at ScienceDirect

Journal of Choice Modelling

journal homepage: http://www.elsevier.com/locate/jocm

Assisted specification of discrete choice models

Nicola Ortelli [a,b,*], Tim Hillel [b], Francisco C. Pereira [c], Matthieu de Lapparent [a], Michel Bierlaire [b]

[a] School of Management and Engineering Vaud (HEIG-VD), University of Applied Sciences and Arts Western Switzerland (HES-SO), Switzerland
[b] Transport and Mobility Laboratory (TRANSP-OR), École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
[c] Machine Learning for Smart Mobility Group (MLSM), Danmarks Tekniske Universitet (DTU), Denmark

ARTICLE INFO

Keywords:
Discrete choice models
Utility specification
Multi-objective optimization
Combinatorial optimization
Metaheuristics

ABSTRACT

Determining appropriate utility specifications for discrete choice models is time-consuming and prone to errors. With the availability of larger and larger datasets, as the number of possible specifications exponentially grows with the number of variables under consideration, the analysts need to spend increasing amounts of time on searching for good models through trial-and-error, while expert knowledge is required to ensure these models are sound. This paper proposes an algorithm that aims at assisting modelers in their search. Our approach translates the task into a multi-objective combinatorial optimization problem and makes use of a variant of the variable neighborhood search algorithm to generate sets of promising model specifications. We apply the algorithm both to semi-synthetic data and to real mode choice datasets as a proof of concept. The results demonstrate its ability to provide relevant insights in reasonable amounts of time so as to effectively assist the modeler in developing interpretable and powerful models.

1. Introduction

In the last 40 years, discrete choice models (DCMs) have been used to tackle a wide variety of demand modeling problems. This is due to their high *interpretability*, which allows researchers to verify their compliance with well-established behavioral theories (McFadden, 1974) and to provide support for policy and planning decisions founded on utility theory from microeconomics. However, the development of DCMs through manual specification is laborious. The predominant approach for this task is to *a priori* include a certain number of variables that are regarded as essential in the model, before testing incremental changes in order to improve its goodness of fit while ensuring its behavioral realism (Koppelman and Bhat, 2006). Because the set of candidate specifications grows beyond manageable even with a moderate number of variables under consideration, *hypothesis-driven* approaches of this kind can be time-consuming and prone to errors. Modelers tend to rely on common sense or intuition, which may lead to incorrectly specified models. The implications of misspecification include lower predictive power, biased parameter estimates and erroneous interpretations (Bentz and Merunka, 2000; Torres et al., 2011; Van Der Pol et al., 2014).

This issue, together with the advent of big data and the need to analyze ever-larger datasets, has induced an increasing focus on machine learning (ML) and other *data-driven* methods as a way of relieving the analyst of the burden of model specification. Unlike

[*] Corresponding author. School of Management and Engineering Vaud (HEIG-VD), University of Applied Sciences and Arts Western Switzerland (HES-SO), Switzerland.

E-mail addresses: nicola.ortelli@heig-vd.ch, nicola.ortelli@epfl.ch (N. Ortelli), tim.hillel@epfl.ch (T. Hillel), camara@dtu.dk (F.C. Pereira), matthieu.delapparent@heig-vd.ch (M. de Lapparent), michel.bierlaire@epfl.ch (M. Bierlaire).

https://doi.org/10.1016/j.jocm.2021.100285
Received 8 July 2020; Received in revised form 4 December 2020; Accepted 16 March 2021
Available online 29 March 2021

DCMs that require the form of their utility functions to be specified *a priori*, ML allows for more flexible model structures to be directly learned from the data. In the past years, numerous studies have therefore investigated the usefulness of ML classifiers as an alternative to logit, mixtures of logit and nested logit models; these studies indicate that DCMs are generally outperformed in terms of prediction accuracy (Hagenauer and Helbich, 2017; Wang and Ross, 2018). However, most ML classifiers suffer from a severe limitation: they lack interpretability. The mathematical structure of DCMs enables the modeler to verify the behavioral realism of the relationships between explanatory variables and choice outcomes; such a feature is crucial to support policy and planning decisions. Also, it allows the derivation of useful indicators that cannot be obtained from ML models, such as willingness to pay or consumer surplus. Whilst there is existing research focusing on extracting utility interpretations and economic indicators from certain ML classifiers (Zhao et al., 2018), obtaining a closed-form expression that maps inputs to outputs is still not possible.

The present study introduces a method to assist the analyst in the specification of random utility models. Our approach involves a metaheuristic procedure that mimics the way an experienced modeler would investigate various specifications, while ensuring that the set of candidates is explored thoroughly, impartially, and efficiently. We define a set of operators that modify an existing model into another one that is not too different and make use of a multi-objective variable neighborhood search algorithm to organize the model development phase. The source code is currently being consolidated and integrated into the Biogeme software (Bierlaire, 2018, 2020).

It is worth emphasizing that it is not our intention to replace the analyst with a black-box algorithm, nor to provide the best possible model for a given application. Rather, we aim at *assisting* modelers in the task of utility specification in order to save them as much time as possible and to consider specifications that they would not necessarily have investigated. Our algorithm is designed to provide a broader understanding of the dataset under consideration and relevant insights about the modeling possibilities associated with it, in reasonable amounts of time. It is likely that experienced analysts are able to find better specifications than our algorithm; nevertheless, the proposed method is valuable in that it allows for a far more thorough search than any human modeler would care to perform. By virtue of its multi-objective nature, our algorithm explores and improves a variety of promising models simultaneously, which effectively avoids committing excessively to a single “path” in the search space. Such an approach is cognitively intractable for human modelers, as the number of models to be kept in memory easily surpasses the capabilities of the human brain.

The remainder of this paper is organized as follows: Section 2 is a non-exhaustive overview of the recent attempts at “enhancing” discrete choice models with data-driven methods, Section 3 introduces a formal definition of the utility specification problem and Section 4 describes the algorithm we propose to solve it. Section 5 presents the results obtained through the application of our method both on real and semi-synthetic mode choice datasets of different sizes and Section 6 summarizes the findings of the present study and identifies the future steps of this research.

2. Literature review

The problem of model specification has been a topic of interest for as long as statistical models have existed. Increasing attention has been brought to this problem as new modeling techniques have appeared; as a result, it has been extensively studied across disciplines such as statistics, econometrics and machine learning (ML). Similarly, interest has recently emerged in the field of discrete choice analysis for methods that are able to “mitigate” the need for presumptive structural assumptions. We organize the relevant literature into three categories: (i) studies that use ML methods as substitutes for discrete choice models (DCMs); (ii) studies that use data-driven methods as specification tools to inform standard DCMs; and (iii) studies that translate the task of model specification into an optimization problem and use search algorithms to solve it. We discuss each of these directions of research in the following subsections.

2.1. *Machine learning methods for choice prediction*

The advancements in computational power and the availability of ever-larger datasets in the recent years have led to numerous breakthroughs in ML research. As a result, ML techniques have been applied to a wide horizon of research fields, including discrete choice analysis. Support vector machines (Hagenauer and Helbich, 2017; Paredes et al., 2017), neural networks (De Carvalho et al., 1998; Zhao et al., 2018; Alwosheel et al., 2019; Wong and Farooq, 2019), decision trees (Tang et al., 2015; Brathwaite et al., 2017) and ensemble learning (Hillel et al., 2018; Wang and Ross, 2018; Lhéritier et al., 2019) have been investigated as potential alternatives for DCMs in a variety of choice problems. Many of these empirical studies conclude that ML classifiers show superior predictive power, but there have been limited attempts to link these new methods with economic theory and human decision making. In discrete choice analysis, the behavioral interpretation of the estimated model is as important as prediction accuracy, because it provides relevant insights for policy and planning decisions. ML classifiers usually lack the coefficient readability of linear models and obtaining a closed-form expression that maps inputs to outputs is often impossible. Interpretable machine learning tools do exist (Zhao et al., 2018), but are rarely applied. As a matter of fact, out of all studies mentioned in this paragraph, only four go beyond computing variable importance. Brathwaite et al. (2017) provide a microeconomic framework for the interpretation of decision trees and combine those with DCMs to model semi-compensatory decision making; Zhao et al. (2018) use partial dependence plots, marginal effects and arc elasticities to extract behavioral findings; Alwosheel et al. (2019) use prototypical examples to examine the relationships learned by neural networks; and Wong and Farooq (2019) extract behavioral insights and econometric properties from the matrix parameters of a residual neural network.

Spurred by these observations, another stream of research has attempted combining DCMs and ML classifiers into a single framework, so as to mitigate their respective limitations. Sifringer et al. (2018), Pereira (2019) and Han et al. (2020) share the same idea of “enhancing” a standard logit model by means of a neural network (NN) with the goal of increasing its overall predictive performance while keeping some key parameters interpretable. To this end, Sifringer et al. (2018) propose a neural embedded logit model. Its utility functions are divided into a manually specified, interpretable part and a nonlinear "representation" component that is learned by a NN. In other words, the alternative-specific constants (ASCs) are modeled as flexible functions of all variables that do not enter the manually specified part of the utility. Han et al. (2020) extend the work of Sifringer et al. (2018) by allowing all parameters—rather than the ASCs alone—to be learned as functions of the individuals' socioeconomic characteristics. In a similar line of thought, Pereira (2019) introduces a method for encoding categorical variables, based on natural language processing techniques; as a preliminary step to specifying a logit model, a NN is used to learn *embeddings* of those variables, allowing for richer nuances than the typical transformations. Through different uses of NNs, the three studies succeed in improving the predictive power of the standard logit while maintaining part of the utility specification interpretable. However, the three approaches suffer from a common drawback: the explanatory variables still need to be manually selected, with no better method than context knowledge and trial-and-error. While these enhanced models are proven more flexible and powerful than traditional DCMs, they require the same effort from the modeler, if not more.

2.2. Informing DCMs with data-driven methods

The idea of combining DCMs with ML classifiers dates back to Bentz and Merunka (2000) and Hruschka et al. (2002). The main difference of these pioneering studies with the methodologies developed by Sifringer et al. (2018), Pereira (2019) and Han et al. (2020) lies in their sequential nature: in Bentz and Merunka (2000) and Hruschka et al. (2002), an exploratory NN is used as a specification tool to inform a standard logit model, which *then* provides interpretable results and significance statistics. The preliminary step allows the identification of the key variables and the detection of nonlinear effects; the modeler can therefore rely on these insights to develop a suitable utility specification.

Thus, in comparison to the hybrid models mentioned in the previous section, "feeding" a logit model with the findings of an exploratory data-driven method offers the advantage of partially relieving the analyst of the burden of model specification—the analyst still needs to provide a set of *potential* explanatory variables—while no additional effort is required to maintain the desirable interpretability of DCMs. This approach proves itself worthy in several other notable studies: Zeileis et al. (2008) use model-based recursive partitioning based on parameter instability to automatically identify relevant variables for parameter segmentation; Hillel et al. (2019) use a gradient-boosting decision-trees ensemble to inform the utility specification of a logit model; and Rodrigues et al. (2019) leverage the concept of automatic relevance determination to pinpoint the most important features for explaining a dataset. In a similar way, penalized regression techniques such as the lasso (Tibshirani, 1996) and the non-negative garrote (Breiman, 1995) are valid alternatives for identifying the key variables of a dataset; both techniques aim at inducing parameter sparsity in a model comprised of all potential features, either by introducing a penalty term in its loss function or by constraining a set of non-negative weights associated with the model parameters.

2.3. Utility specification as an optimization problem

Finally, another relevant stream of research addresses the problem of model specification by translating it into an optimization task and solving it using iterative search algorithms. The existing literature proposes a large variety of feature selection techniques that rely on metaheuristics such as simulated annealing (Lin et al., 2008; Brusco, 2014), genetic algorithms (Oh et al., 2004; Soufan et al., 2015) and tabu search (Chuang et al., 2009; Pacheco et al., 2009) to iteratively build the optimal subset of variables for inclusion in a predictive model. These methods are commonly referred to as "wrappers", as they use the model of interest to score variable subsets based on the goodness of fit or predictive accuracy they enable. The main limitation of wrappers is that they are generally designed to perform model building only within the original variable space, eventually supplemented with *predefined* transformations and interactions of variables. Because of this lack of flexibility, most of these methods may fail badly in the presence of unknown, complex interdependencies and nonlinearities such as the ones typically encountered in economic data (Doornik, 2008).

More flexible methods that overcome this limitation do exist. For example, multivariate adaptive regression splines (Friedman, 1991) rely on a greedy search algorithm to automatically select which variables to use in a regression model, specify hinge functions that account for non-linearity and combine them together to incorporate interactions between variables in the model. Alternatively, Paz et al. (2019) prove that metaheuristics can be seamlessly adapted so as to produce more complex model specifications: their simulated annealing algorithm simultaneously selects subsets of variables and parameter distributions of a mixed logit model, such that the Bayesian information criterion (Schwarz, 1978) is maximized.

We therefore believe that translating the task of utility specification into an optimization problem is a particularly promising direction of research, provided the generated specifications can account for variable interactions and nonlinearities. This approach allows researchers to benefit from the vast combinatorial optimization literature and its variety of widely recognized metaheuristics. These methods are specifically designed to find "good" solutions for problems that cannot be solved through exhaustive testing, in reasonable amounts of time; they hence seem particularly suited for the utility specification problem. An additional benefit of using metaheuristics in this context lies in their iterative nature. This allows any "post-estimation" measure of model quality to be seamlessly used as the objective function.

In this paper, we focus on the metaheuristic approach, in a similar way as Paz et al. (2019). The main contribution of this paper is a multi-objective optimization algorithm that generates sets of good models based on an *information set* provided by the analyst. The information set is used to build a space of possible specifications and may contain any form of variable interaction, nonlinear transformation, segmentation of the population in the dataset and potential choice models; the space is then explored by a variable neighborhood search algorithm that proceeds by sequentially introducing small modifications to an initial set of “promising” utility specifications. Our methodology offers the additional advantage of proposing several model specifications rather than a single one, whereas most studies mentioned in the current section propose a single model that is deemed as “best”. While a single best model may exist in experiments that use synthetic data—*i.e.*, the one model that created them—there is no such thing as an overall best model that explains all aspects of real data (Burnham and Anderson, 2002, 2004; Claeskens and Hjort, 2003, 2008). Rather, we believe that the definition of a “suitable” model is context-dependent and that different models should be developed according to the objective that is to be achieved. A number of studies try to justify the use of a single metric to identify the best model among a set of candidates; in our view, multi-objective optimization makes such approaches unnecessary. By defining appropriate objectives, sets of models can be generated that meet the plural needs of any predictive modeling problem.

3. Problem formulation

We are specifying a choice model with J alternatives, using a dataset that contains observations from N individuals. For each individual n, we have at our disposal: (a) a vector s_n of socioeconomic characteristics; (b) a vector x_{in} of attributes for each alternative i; and (c) the chosen alternative i_n. In addition, the analyst provides the following inputs, that we refer to as the “information set”:

- A partition of attributes into K groups that capture the same variable in different utility functions. Attributes in the same group may be associated with a generic or an alternative specific coefficient. They also share the same functional form. For instance, one group may correspond to the “cost” variable; in this case, if the cost of alternative i is integrated with a log transform, so is the cost of any other alternative. It is assumed that each attribute belongs to exactly one group. We denote x_{ink} the attribute of group k for alternative i. It is assumed to be zero if group k does not contain an attribute for alternative i.
- A list of potential transformations of the attributes $f_\ell(x; s_n, \lambda)$, $\ell = 1, \ldots, L$, possibly depending on socioeconomic characteristics s, and possibly involving additional parameters λ. Typical examples would be a logarithm:

$$f_\ell(x) = \ln(x), \tag{1}$$

a power:

$$f_\ell(x; \text{Income}_n) = \left(\frac{x}{\text{Income}_n}\right)^2, \tag{2}$$

or a Box-Cox transform:

$$f_\ell(x; \lambda) = \frac{x^\lambda - 1}{\lambda}. \tag{3}$$

We automatically add to this list the linear specification:

$$f_0(x) = x. \tag{4}$$

- A discrete segmentation of the population based on each socioeconomic characteristic separately. We denote $R(s)$ the number of segments characterized by the socioeconomic characteristic s. For example, we may have two income groups, five levels of education, or four age classes. As each individual belongs to exactly one segment, we define $\Delta_{srn} = 1$ if individual n belongs to segment r for socioeconomic characteristic s and 0 otherwise.
- A list of potential choice models $P_m(i|V; \mu)$, $m = 1, \ldots, M$, that calculate the choice probability as a function of the deterministic part of the utility functions. Typical models are the logit, nested logit or mixtures of logit models, among others.

We formulate the specification problem as an optimization problem. Binary decision variables are associated with each dimension introduced above:

- For each group of attributes k, δ_k is 1 if group k is introduced in the model, 0 otherwise.
- For each group k, γ_k is 1 if the group is associated with a generic coefficient, 0 otherwise.
- For each group k, $\phi_{k\ell}$ is equal to 1 if k is associated with functional transform ℓ and 0 otherwise. Each group is associated with exactly one form, so that we impose that

$$\sum_{\ell=0}^{L} \phi_{k\ell} = 1, \ \forall k. \tag{5}$$

- For each group of attributes k and each socioeconomic characteristic s, σ_{ks} is 1 if the coefficients of the attributes in the group are segmented based on socioeconomic characteristic s, and 0 otherwise. Note that the segmentation can be based on several socioeconomic characteristics simultaneously.

- For each model m, ρ_m is 1 if model m is used to calculate the choice probabilities, and 0 otherwise. We impose that only one model is used:

$$\sum_{m=1}^{M} \rho_m = 1. \tag{6}$$

We may now denote as $\omega = (\delta, \gamma, \phi, \sigma, \rho)$ the vector of all decision variables that intervene in the specification problem. The choice model that calculates the probability as a function of the deterministic part of the utility functions is defined as

$$P\big(i|V_n;\mu\big) = \sum_{m=1}^{M} \rho_m P_m(i|V_n;\mu). \tag{7}$$

The utility function for alternative i is defined as

$$V_{in} = \sum_{k=1}^{K} \delta_k T_{ikn}, \tag{8}$$

where the term T_{ikn} is defined as

$$T_{ikn} = (\gamma_k \beta_{kn} + (1-\gamma_k)\beta_{ikn}) \sum_{\ell=1}^{L} \phi_{k\ell} f_\ell(x_{ink};\lambda) \tag{9}$$

The generic coefficient β_{kn} is defined as

$$\beta_{kn} = (1-\sigma_{ks})\overline{\beta}_k + \sigma_{ks}\left(\sum_{s}\sum_{r=1}^{R(s)} \beta_{krs}\Delta_{srn}\right), \tag{10}$$

and the alternative-specific coefficients β_{ikn} are similarly defined:

$$\beta_{ikn} = (1-\sigma_{ks})\overline{\beta}_{ik} + \sigma_{ks}\left(\sum_{s}\sum_{r=1}^{R(s)} \beta_{ikrs}\Delta_{srn}\right). \tag{11}$$

The parameters to be estimated by maximum likelihood are the β parameters associated with the variables, the λ parameters associated with the functional forms, and the μ parameters specific to each choice model. The total number of parameters $Z(\omega)$ depends on the decision variables ω and so does the optimal value of the log likelihood function $\mathcal{L}(\omega)$. The optimization problem that we consider aims at identifying models with the best possible fit, that is the largest value of the log likelihood function $\mathcal{L}(\omega)$, and the most parsimonious ones, that is with the lowest value of $Z(\omega)$. By definition, these two objectives are conflicting, in the sense that there exists no solution that simultaneously optimizes both; a nontrivial multi-objective optimization problem is hence defined. We say that a model characterized by a vector of decision variables ω_1 dominates a model characterized by ω_2 if the former is not worse than the latter in any objective, that is

$$\mathcal{L}(\omega_1) \geq \mathcal{L}(\omega_2) \text{ and } Z(\omega_1) \leq Z(\omega_2), \tag{12}$$

and strictly better in at least one objective, that is

$$\mathcal{L}(\omega_1) > \mathcal{L}(\omega_2) \text{ or } Z(\omega_1) < Z(\omega_2). \tag{13}$$

The dominance of the first model over the second is denoted by $\omega_1 \prec \omega_2$. Note that the dominance relation is transitive, but not reflexive, not symmetric, and not complete, in the sense that there exist pairs of models such than none of them dominates the other one.

The objective is therefore to find the set of Pareto-optimal solutions, that is the set $\mathcal{P}$ such that, if $\omega^* \in \mathcal{P}$, there is no feasible solution ω such that $\omega \prec \omega^*$. Clearly, the complexity of the problem does not allow to solve it exactly. In the next section, we describe a heuristic designed to approximate the set of Pareto-optimal solutions in a reasonable amount of computational time.

We clarify a number of points before moving to the next section:

- Alternative-specific constants are considered as a group of attributes, so that they share the same potential transform. However, the associated decision variable γ_k is constrained to be 0 to impose that they are alternative-specific.
- We present above the problem with a generic list of nonlinear transformations and socioeconomic characteristics for segmentation. In practice, it is recommended to assign different lists of potential transformations and segmentations to each group of attributes.
- The combinatorial nature of the segmentation may produce models with an excessive number of coefficients. It is good practice to impose a maximum number of parameters and reject automatically any model that exceeds this number, without estimating it.

- The analyst is free to use other objectives, in addition or instead of the two proposed in this section. The concept of dominance introduced above is straightforwardly extended to larger dimensions.

4. Proposed algorithm

Metaheuristics typically have three ingredients: exploration, intensification and diversification. The exploration of the solution space is problem-specific, and relies on operators and neighborhood structures, designed to mimic what a modeler would do. They must be flexible enough to potentially allow any feasible solution to be reached. This is certainly the key ingredient for a successful algorithm. Intensification and diversification are more generic strategies designed to find local optima and escape from them, respectively.

The exploration strategy for the specification of choice models is described in Section 4.2. At this point, we just assume that we have access to neighborhood structures of various sizes $p = 1,\ldots,P$. We first describe the generic algorithm for intensification and diversification. It is a multi-objective variant of the variable neighborhood search (VNS) metaheuristic from Mladenovic and Hansen (1997).

The main advantage of variable neighborhood search over other metaheuristics such as simulated annealing is that it explicitly exploits neighborhood structures that are provided by the expert. It is therefore appropriate for our approach that is designed to assist an expert, as opposed to solve a combinatorial optimization problem. Additionally, it does not require any parameter tuning and is seamlessly adapted to the multi-objective context, whereas adapting simulated annealing is not as straightforward: the existing literature thereon is fragmented.

While exploring the space of possible models, the algorithm sorts them into two sets:

- the set $\mathcal{S}$ of all models considered by the algorithm so far;
- the set $\mathcal{P}\subseteq\mathcal{S}$ of models that are not dominated by any model in $\mathcal{S}$, where the concept of dominance is defined by (12) and (13).

4.1. Generic algorithm

The generic algorithm is organized as follows.

Input The ingredients provided to the algorithm are:

- A non-empty set $\mathcal{S}$ containing model specifications provided by the analyst. If the analyst does not provide any model specification, a simple model involving only alternative-specific constants (ASCs) is used for the initialization.
- A list of neighborhood structures $\mathcal{N}_p(\omega)$, for $p = 1,\ldots,P$. These are sets of models constructed from ω using operators of size p, as described in Section 4.2.
- A maximum number Q of unsuccessful candidates for each structure $\mathcal{N}_p(\omega)$.

Initialization

- The set $\mathcal{P}$ is initialized with each model in $\mathcal{S}$ that is not dominated, *i.e.*, each $\omega \in \mathcal{S}$ such that no $\omega' \in \mathcal{S}$ verifies $\omega' \prec \omega$.
- The size of the neighborhood is initialized to $p = 1$.

Loops The algorithm is composed of two nested loops:

- The outer loop iterates on the neighborhood size $p = 1,\ldots,P$.
- The inner loop iterates on candidate neighbors $q = 1,\ldots,Q$ within the current neighborhood structure.

When Q candidates have been *unsuccessfully* investigated for neighborhood structure p, the algorithm moves to the next neighborhood structure $p+1$. When the last neighborhood structure P has been processed, the algorithm stops. Every time that an improved model is found, the iterations are restarted.

Iteration Each iteration consists in the following steps:

1 A model $\omega \in \mathcal{P}$ is randomly selected.
2 A model ω^+ is selected in the neighborhood $\mathcal{N}_p(\omega)$.
3 If $\omega^+ \in \mathcal{S}$, the neighbor model has already been investigated. The current iteration is interrupted and the algorithm proceeds with the next one.
4 The neighbor model is added to the set of considered models: $\mathcal{S} = \mathcal{S} \cup \{\omega^+\}$.
5 The parameters of the model are estimated. A model is rejected if the estimation fails, takes too much time, or if the estimated values of the parameters are deemed inconsistent with the theory. Typically, a model may be rejected if some parameters have the wrong sign, or the estimate of a nest parameter in a nested logit model is below one. If a model is rejected, the algorithm proceeds with the next iteration.

6 The algorithm then checks if the candidate improves the Pareto set. We define $\mathcal{D}^{-}$, the set of models in $\mathcal{P}$ dominated by ω^{+}, as

$$\mathcal{D}^{-} = \{\omega \in \mathcal{P} | \omega^{+} \prec \omega\}. \tag{14}$$

and $\mathcal{D}^{+}$, the set of models in $\mathcal{P}$ dominating ω^{+}, as

$$\mathcal{D}^{+} = \{\omega \in \mathcal{P} | \omega \prec \omega^{+}\}. \tag{15}$$

If $\mathcal{D}^{+} = \emptyset$ it means that ω^{+} is not dominated by any model in the set $\mathcal{P}$. The iteration is successful, and $\mathcal{P}$ is updated:

$$\mathcal{P} = \mathcal{P} \cup \{\omega^{+}\} \setminus \mathcal{D}^{-}. \tag{16}$$

After a successful iteration, the loops are restarted by setting $p = 1$ and $q = 1$.

4.2. Operators

The algorithm described in Section 4.1 is generic, in the sense that it can be applied to a large family of combinatorial multi-objective optimization problems. Its key ingredients are the neighborhood structures. They must reflect the properties of the specific problem at hand. In our context, we define the neighborhood structures using a series of *p-operators*, that alter the current model specification by modifying p decisions, in a way similar to what an expert analyst would do. $\mathcal{N}_p(\omega)$, the neighborhood structure of size p, is the set of models obtained by applying each one of the p-operators.

Let us consider again $\omega = (\delta, \gamma, \phi, \sigma, \rho)$, the vector of all decision variables defined in Section 3. Each operator modifies a subset of the decision variables. We define the following operators, where p is the size of the neighborhood:

Change variables Randomly select p groups among the K groups of attributes and change their activation status:

$$\delta_k = 1 - \delta_k. \tag{17}$$

Change generic Consider all active groups of attributes:

$$\{k | \delta_k = 1\}. \tag{18}$$

Randomly select p groups and associate them with a generic coefficient instead of an alternative-specific one, or the other way around:

$$\gamma_k = 1 - \gamma_k. \tag{19}$$

Note that in practice, the analyst may want to exclude the use of a generic coefficient for some groups of attributes—such as the ASCs, as discussed in one of the comments at the end of Section 3. In that case, they are simply excluded from the list mentioned above.

Change linearity Consider all active groups of attributes k and randomly select p of them. For each selected group: if the functional form is linear, change it to a random nonlinear specification ℓ'; if it is nonlinear, change it to linear.

$$\begin{aligned} &\text{If } \phi_{k0} = 1, \text{then } \phi_{k0} = 0,\ \phi_{k\ell'} = 1, \\ &\text{If } \phi_{k0} = 0, \text{then } \phi_{k0} = 1,\ \phi_{k1} = \ldots = \phi_{kL} = 0. \end{aligned} \tag{20}$$

This operator provides a specific status to the linear specification compared to all non-linear specifications.

Change non-linearity Randomly select p active groups with a non-linear specification:

$$\{k | \delta_k = 1 \text{ and } \phi_{k0} = 0\} \tag{21}$$

For each of them, if ℓ is the current index of the nonlinear specification, *i.e.*, if $\phi_{k\ell} = 1$, randomly select another index ℓ' instead, such that $\ell' \neq 0$ and $\ell' \neq \ell$:

$$\phi_{k\ell} = 0 \text{ and } \phi_{k\ell'} = 1. \tag{22}$$

Change segmentation Consider all pairs of active groups of attributes k and socioeconomic characteristics s:

$$\{(k, s) | \delta_k = 1\}. \tag{23}$$

Randomly select p pairs and change the corresponding segmentation:

$$\sigma_{ks} = 1 - \sigma_{ks}. \tag{24}$$

Increase segmentation Consider all pairs of active groups of attributes k and socioeconomic characteristics that are not interacting:

$$\{(k,s)|\delta_k = 1 \text{ and } \sigma_{ks} = 0\} \tag{25}$$

Randomly select p pairs and activate the corresponding segmentation:

$$\sigma_{ks} = 1. \tag{26}$$

Decrease segmentation Consider all pairs of active groups of attributes k and socioeconomic characteristics that are interacting:

$$\{(k,s)|\delta_k = 1 \text{ and } \sigma_{ks} = 1\} \tag{27}$$

Randomly select p pairs and deactivate the corresponding segmentation:

$$\sigma_{ks} = 0. \tag{28}$$

Change model If m is the currently selected model, that is, if $\rho_m = 1$, randomly select another model $m^{'}$ instead, such that $m^{'} \neq m$:

$$\rho_m = 0 \text{ and } \rho_{m'} = 1. \tag{29}$$

Each time a neighbor of ω is requested, one of the operators is invoked to generate it. The selection of the operator is performed at random. The probability to select an operator can be the same for each operator, or can be updated based on statistics of success of each operator in the course of the algorithm.

4.3. *Ensuring behavioral realism*

Because inconsistency with economic theory is generally understood as evidence of misspecification and invalidates all insights extracted from an estimated model, behavioral soundness should always be verified. In practice, the analyst is given the option to indicate a collection of constraints that she deems as necessary to prove model validity: for example, taste parameters related to the cost of an alternative are usually expected to be negative, so as to decrease the utility of the alternative if its price increases; similarly, the scale parameters assigned to the nests of a nested logit model should always be constrained to be greater than 1. By default, if such a set of constraints is defined, the algorithm deems as invalid and rejects any model whose parameter estimates violate any of the constraints, regardless of its performance in terms of fit or parsimony (Step 5 of the iteration).

During our experiments, however, it appeared that a high rate of rejections may decrease the capability of the algorithm to explore the space of possible specifications. Moreover, the above-described approach may conceal more severe specification issues, such as potential endogeneity problems in the data; in that case, the analyst would have no way of suspecting that she needs to correct for them. For this reason, we also consider the use of constrained maximum likelihood estimation (Schoenberg, 1997), in which the constraints on the parameters are explicitly enforced. This serves three purposes: (i) "behaviorally valid" values are ensured for all parameters; (ii) the number of rejected models is decreased, which provides more flexibility to the algorithm; and (iii) the presence of active constraints at the end of the estimation process indicates to the analyst possible issues that need to be examined. We point out that when models are estimated by constrained maximum likelihood, the value of $Z(\omega)$ is equal to the number of parameters of the model, minus the number of active constraints at the end of the estimation.

4.4. *Post-processing*

Our algorithm is designed to approximate the Pareto front of the optimization problem described in Section 3, meaning that it returns an ensemble of promising models rather than a single solution. It is the responsibility of the analyst to select one specification based on the insights that the Pareto-optimal models *jointly* provide. In particular, we recommend the use of classical metrics in model selection as guidelines. These serve the additional role of illustrating how little information single-objective methods actually provide about the considered dataset and its associated modeling possibilities in comparison to a multi-objective approach.

We first consider out-of-sample validation for assessing how accurately the models in the Pareto front generalize to new data. Testing the model on unseen data is a way to effectively reduce the possibility of erroneous correlation between inputs and choice outcomes that may have been captured during the training. The method consists in excluding part of the data from the model identification process. After the algorithm stops, the accuracy of the Pareto-optimal models is measured on the hold-out data; the optimal log likelihood yielded “in-sample” and “out-of-sample” may be compared to detect underfitting or overfitting issues. Out-of-sample validation prevents data-leakage from the validation set during model training. However, there may be instances where the original dataset is relatively small, and so an estimation on all of the data is desired to identify an appropriate specification—where the small data size could limit the significance of included parameters. In this instance, it is advised to run the algorithm with the complete data set. Then, during the post-processing validation, each specification is re-estimated on a portion of the data, and validated on the rest. This approach limits data-leakage to the estimated parameter values.

In addition to out-of-sample validation, we also identify the optimal models in terms of the Akaike and Bayesian information criteria (Akaike, 1974; Schwarz, 1978). These criteria work by “balancing” the log likelihood $\mathcal{L}(\omega)$ yielded by a model ω with a penalty that is proportional to its number of estimated parameters $Z(\omega)$; in other words, they quantify the trade-off between the two objectives of the problem defined in Section 3. The Akaike information criterion (AIC) and the Bayesian information criterion (BIC) are defined as

$$\begin{aligned} \text{AIC} &= -2\mathcal{L}(\omega) + 2Z(\omega), \\ \text{BIC} &= -2\mathcal{L}(\omega) + Z(\omega)\log(N), \end{aligned} \tag{30}$$

where N is the size of the sample used to train the model. The penalty for additional parameters considered by the BIC is more severe than the one of the AIC; we therefore expect BIC-optimal models to be more parsimonious—and worse in terms of in-sample log likelihood—than AIC-optimal ones.

Finally, model averaging techniques may be used in order to extract additional value from the Pareto front. For example, one may leverage the entire ensemble of promising models to achieve better out-of-sample prediction, to produce better estimates of variable importance or to estimate model-based uncertainty measures for important quantities, such as the predicted probabilities for all individuals (Claeskens and Hjort, 2008). These techniques are promising, but fall out of the scope of this study. We nevertheless recommend them for further analysis of the results provided by our methodology, so as to take additional advantage of the set of Pareto-optimal models.

4.5. Implementation notes

For the sake of clarity, some details related to the implementation of our algorithm are omitted from the previous sections. In particular, two measures are undertaken to reduce the computational time of its runs. We discuss these measures here.

All model estimations are performed using the Biogeme package for Python (Bierlaire, 2018, 2020) and its implementation of the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm. Substantial amounts of time are gained by limiting the estimation outputs provided by Biogeme to what is strictly necessary. In particular, the calculation of the variance-covariance matrix of the parameter estimates is expensive but not used by the algorithm. We avoid computing it.

Additionally, whenever possible, the estimation outputs of previously tested models are used as initial values for the next estimation, in order to reduce the number of iterations of the BFGS algorithm. As every model arises from the application of an operator on a previously estimated model, it is expected that the parameter estimates of the former are good approximations of the values that should be reached by the latter; they are therefore used as initial values whenever appropriate. In practice, no parameter that has already been estimated in a previous model is ever estimated from scratch again. This effectively reduces the overall computational cost of our algorithm.

5. Case studies

This section presents the results obtained through the application of our algorithm on two mode choice datasets, all publicly available online.[1] We conduct two types of experiments: (i) using semi-synthetic data for which the true specification and related parameters are known; and (ii) using real-world data for which the data-generation process is unknown. If not stated otherwise, all results shown in this section are obtained after running the algorithm for 72 hours on a 2.3 GHz 32-core server with 192 GB of RAM.

5.1. Swissmetro

The Swissmetro dataset (Bierlaire et al., 2001) consists of stated-preference survey data collected in Switzerland in 1998, initially used to analyze the potential impact of the Swissmetro, the Swiss precursor of Hyperloop. The 1’192 respondents were asked to state their favorite transportation mode among three alternatives—train, Swissmetro and car—in nine different hypothetical situations. 10’710 data points remain after removing incomplete observations. Table 1 gives a brief description of the variables used in the context of this paper; more details may be found in Antonini et al. (2007).

We start by setting approximately 20% of the 10’710 observations aside for out-of-sample (OOS) validation. The remaining 8’568 observations are fed to the algorithm, together with an information set that contains:

- the 8 continuous attributes of Table 1, organized into $K = 3$ groups—*i.e.*, travel time, travel cost and headway;

[1] https://biogeme.epfl.ch/data.html.

Table 1
Swissmetro. Description of the considered variables.

Variable	Description
TT Train	*Train travel time. Based on the car distance.*
TT SM	*Swissmetro travel time. A speed of* 500 km/h *is considered.*
TT Car	*Car travel time.*
CO Train	*Train cost. Equal to zero if the traveler owns a GA.*
CO SM	*Swissmetro cost. Proportional to the rail fare.*
CO Car	*Car cost. A fixed average cost per kilometer is considered.*
HE Train	*Train headway.*
HE SM	*Swissmetro headway.*
Class	*1 if first-class traveler, 0 otherwise.*
GA	*Travel card. 1 if the traveler owns one, 0 otherwise.*
Luggage	*Pieces of luggage: None, one, or several.*
Gender	*Gender of the traveler.*
Who	*Who pays for the trip: self, employer or half-half.*
Income	*Yearly income class: 0–50k, 50k–100k, 100k+, or unknown.*
Age	*Age category: 0–24, 25–39, 40–55, 55–65, or 66+.*

- $L = 5$ nonlinear transformations: the logarithm, square-root and square, in addition to two piecewise linear transformations with two or four predefined segments for the travel time and travel cost.
- the 7 socioeconomic characteristics of Table 1 for segmentation;
- $M = 3$ different models, including one logit and two nested logit models; the nested logit models assume correlation either between the two existing alternatives—train and car—or between the two public alternatives—train and Swissmetro.

This setting potentially enables over 4.6×10^8 distinct specifications. In accordance with behavioral and economic theory, we impose all parameters related to time and cost to be negative. We also impose the nest coefficients of the nested logit models to be larger than one, as required by random utility theory. All models are therefore estimated through constrained maximum likelihood, as described in Section 4.3.

We apply our algorithm on the Swissmetro dataset with the above-described configuration. The initial Pareto front contains a single specification that only involves the alternative-specific constants (ASCs). In total, 4'768 distinct specifications are considered and

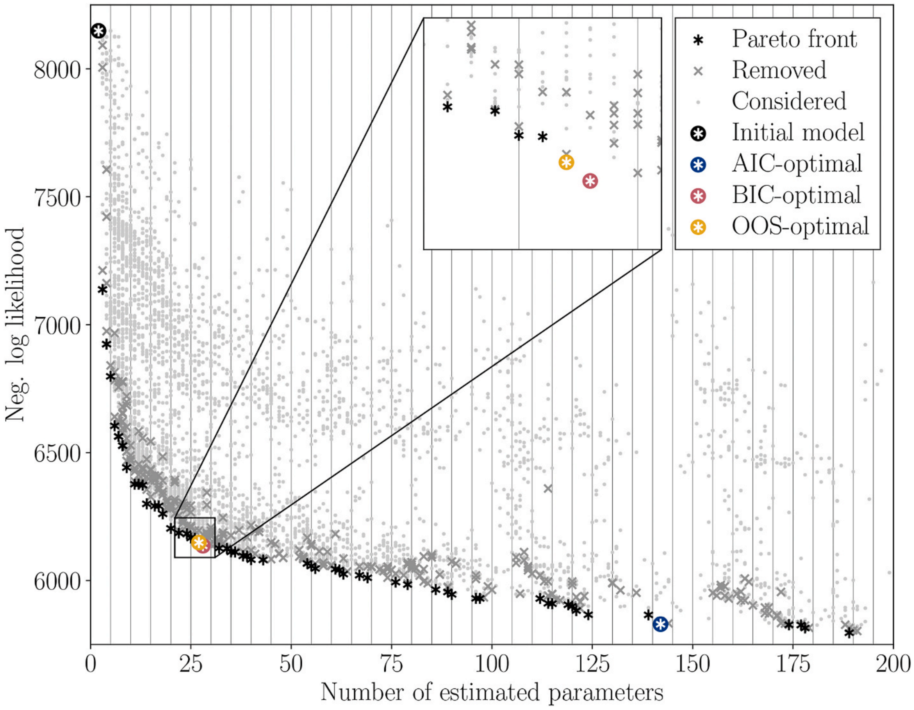

Fig. 1. Swissmetro. Pareto front visualization.

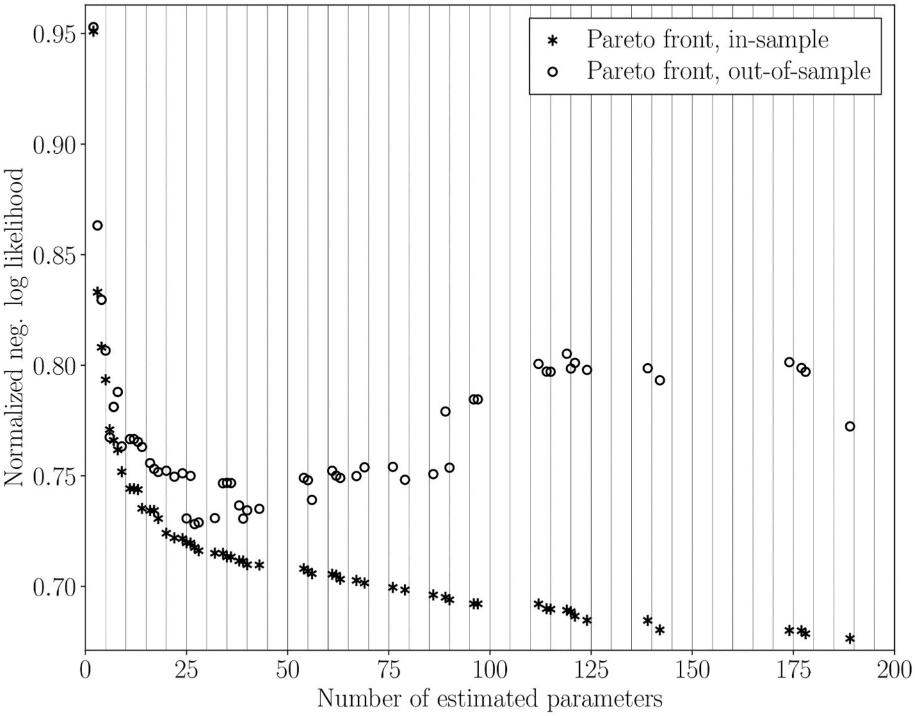

Fig. 2. Swissmetro. In-sample and out-of-sample log likelihood comparison.

Table 2
Swissmetro. Performance of the AIC-, BIC- and OOS-optimal models.

	AIC-optimal	BIC-optimal	OOS-optimal
Estimated parameters	142	28	27
In-sample log likelihood	– 5’829.2	– 6’135.9	– 6’148.3
Out-of-sample log likelihood	– 1’706.1	– 1’567.9	– 1’566.3

tested by the algorithm. Fig. 1 illustrates their performance in terms of fit and parsimony. All points labeled as “removed” were part of the Pareto front at some point during the search, until the encounter of a dominating model excluded them from the front. The 58 models in the Pareto front range between −5’795.9 and −8’148.7 in terms of log likelihood. The largest Pareto-optimal model includes 189 parameters and the smallest—*i.e.*, the ASCs-only model—only two. The front approximately follows the shape of a hyperbola branch, which highlights the trade-off between the two objective functions: when the model is small each additional parameter significantly improves its fit; but as the model grows larger the marginal improvement in log likelihood decreases. The models that perform best in terms of Akaike information criterion (AIC), Baysian information criterion (BIC) and out-of-sample log likelihood are highlighted as an illustration of the results the most common model selection criteria would provide. We report the performance of the three models in Table 2. Their specifications are shown in Tables 4–6 in the Appendix. The dense cloud of considered models with a rather small number of parameters suggests that time could be saved by directing the search towards larger model sizes after a certain stability is acquired with respect to the smaller Pareto-optimal solutions.

Fig. 2 serves the purpose of comparing the in-sample and out-of-sample log likelihoods yielded by all models in the Pareto front (see Fig. 1). These values are normalized with respect to the number of observations in the training and validation data, so as to compare them despite the difference in size between the two sets. We observe that the difference between the two log likelihoods is slight for models that include up to 30 parameters. Then, as the number of parameters grows, the accuracy on the validation data worsens, while the fit on the training data keeps improving. This trend suggests that the largest models in the Pareto front may be overfitting. On the contrary, the limited explanatory power of all models with less than 15 parameters is a clear sign of underfitting. The best out-of-sample log likelihood is obtained by the 27-parameter model, that we denote as the OOS-optimal model. As shown in Table 2, the OOS-optimal model is even more parsimonious than the BIC-optimal one; its in-sample log likelihood is slightly worse as a result. This illustrates well how the Pareto set helps the analyst to focus on a relatively small number of potentially good models, in order to select the best one.

5.2. Semi-synthetic data

Synthetic data provide a controlled environment for testing the suitability of the assisted specification algorithm. By knowing the exact process that generated a set of artificial observations, one can easily assess the quality of a model that is supposed to reflect them. Attention should be drawn to the fact that in this experiment only the response variable is created artificially, while all explanatory variables are selected from a real-world dataset. This ensures the artificial data to be realistic; in fact, fully synthetic observations tend to lack noise and correlation among variables, which are inherent to real data. We re-use the Swissmetro dataset for this purpose.

To create our semi-synthetic data, we define a 21-parameter nested logit model that will serve as the data-generation process. We refer to it as the "true model"; its specification is given in Table 7 in the Appendix. The true model is estimated through maximum likelihood estimation using the 10'710 *original* observations of the Swissmetro dataset, which ensures realistic parameter values. Then, the probabilities of choosing each alternative—as given by the true model—are used to sample a new set of synthetic choices.

We apply our algorithm on the resulting 10'710 semi-synthetic observations with the same information set as in the previous experiment. 1'760 models are explored in 24 hours; the resulting Pareto front contains 23 models. It is illustrated in Fig. 3. The Pareto-optimal models range from −10'059.5 to −8'138.2 in terms of log likelihood and have between 2 and 176 estimated parameters. The front follows a "flatter" shape than in the previous experiment: up to the 18-parameter model each additional parameter largely improves the fit, but for larger model sizes the marginal improvement quickly becomes negligible. This meets our expectations: the transition point approximately corresponds to the size of the true model, which means that all models that include more parameters only extract noise from the data and do not constitute any improvement in comparison to smaller solutions. Our algorithm fails to

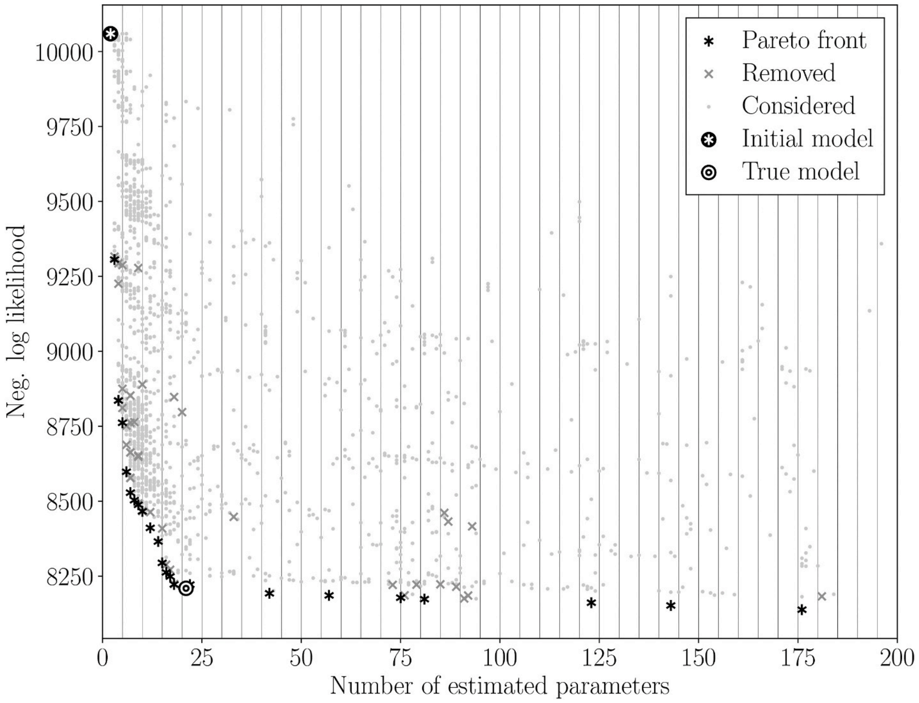

Fig. 3. Semi-synthetic data. Pareto front visualization.

recover the exact data-generation process, but reaches specifications that are similar to the one of the real model: for example, the 18 and 22-parameter Pareto-optimal models are in the neighborhoods of size 3 and 2 of the true model, respectively. The specification of these two models are shown in Tables 8–9.

5.3. London Passenger Mode Choice

Our last experiment is based on the London Passenger Mode Choice (LPMC) dataset (Hillel et al., 2018). The LPMC dataset is more challenging than the Swissmetro dataset for two reasons: (i) it is larger both in terms of observations and explanatory variables; and (ii) it consists of revealed-preference data, which are known to be much noisier than stated-preference surveys.

The LPMC dataset consists of trip records collected over three years, combined with systematically matched trip trajectories alongside their corresponding mode alternatives. In total, it contains details for over 80’000 trips. Four modes are distinguished: walking, cycling, public transport and driving. Table 3 provides a description of the variables used in this experiment; additional details concerning the dataset and its collection procedure may be found in Hillel (2019a).

We start by dividing the dataset into two parts. The first two years of data—54’766 observations—are used for model development whilst the final year of data—26’320 observations—is set aside for OOS validation. We define the following information set:

- The 12 first variables of Table 3 are considered for inclusion. They are organized into $K = 5$ groups: distance, travel time, travel cost, transfers and variability.
- $L = 3$ nonlinear transformations—logarithm, square-root and square—are available for all variables related to time and cost.
- The 8 last variables of Table 3 are considered for segmentation.
- $M = 4$ distinct model structures are defined, including one logit and three nested logit models. The nested logit models assume correlation either between the two soft modes—walking and cycling—between the two motorized modes—public transport and driving—or in the two pairs simultaneously.

The information set results in over 4.7×10^{10} possible model specifications. We impose all parameters related to distance, time and cost to be negative.

Fig. 4 shows the results obtained by applying our algorithm on the LPMC dataset with the above-described information set. The initial Pareto front includes the ASCs-only model alone. In total, “only” 2’084 different specifications are tested by the algorithm. This is due to the size of the LPMC dataset, which makes the estimation computationally intensive. The Pareto front contains 42 models, ranging from 3 to 81 estimated parameters and from −37’056.0 to −62’035.6 in terms of log likelihood. The largest model in the front

Table 3
LPMC. Description of the considered variables.

Variable	Description
Distance	*Straight line between origin and destination.*
Time Walk	*Duration of walking route.*
Time Cycle	*Duration of cycling route.*
Time PT access	*Access and egress time for public transport route.*
Time PT rail	*Rail in-vehicle time.*
Time PT bus	*Bus in-vehicle time.*
Time PT inter	*Interchange time for public transport route.*
Time Drive	*Duration of driving route.*
Cost PT	*Cost for public transport route.*
Cost Drive	*Fuel cost and congestion charge cost of driving route.*
Transfers	*Number of interchanges on public transport route.*
Variability	*Traffic variability in percentage.*
Departure	*Night, AM peak, interpeak or PM peak.*
Day	*Weekday, Saturday or Sunday.*
Winter	*1 if trip occurred in Dec., Jan. or Feb., 0 otherwise.*
Age	*Age category: 0–17, 18–64, or 65+.*
Gender	*Gender of the traveler.*
License	*1 if the traveler has a driving license, 0 otherwise.*
Car owner	*no car in the household, less than one per adult, or more.*
Purpose	*Purpose of the trip. Five distinct categories.*

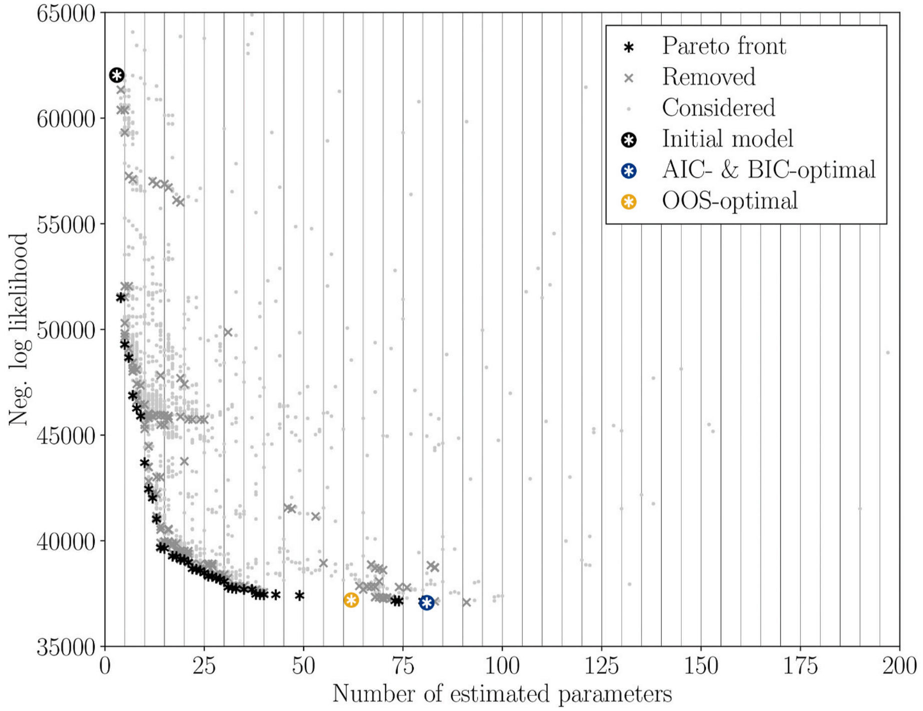

Fig. 4. LPMC. Pareto front visualization.

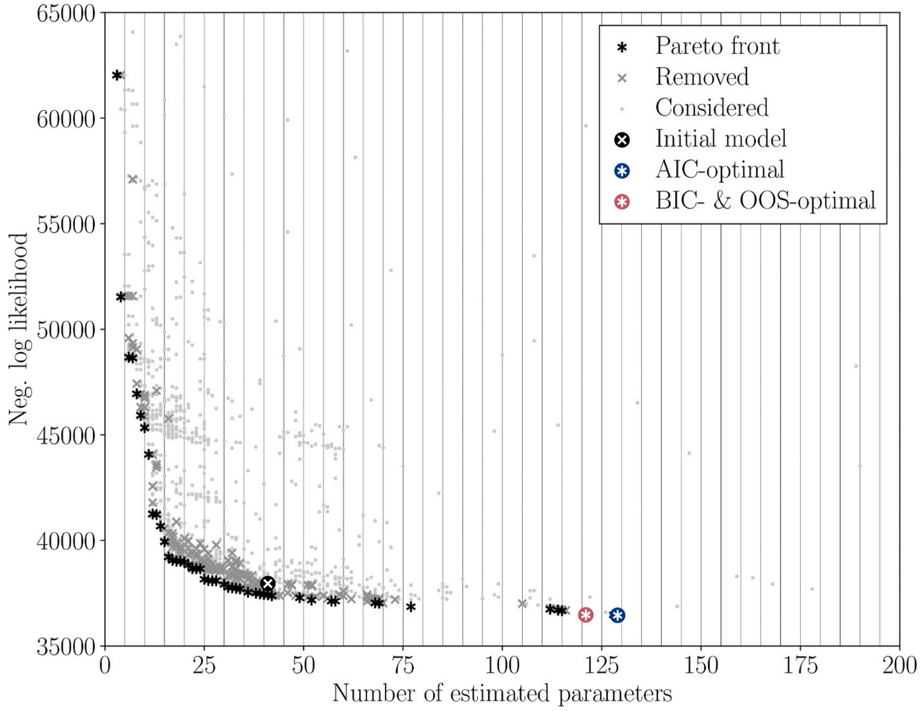

Fig. 5. LPMC. Alternate initialization. Pareto front visualization.

is also the best in terms of AIC and BIC. In comparison, the OOS-optimal model found by the algorithm has 62 parameters, but yields a log likelihood of $-$ 37’048.3.

Finally, we apply our algorithm on the LPMC dataset again, but with an alternate starting point. For this purpose, we replace the ASCs-only model in the initial Pareto front with a model proposed by Hillel (2019b). We refer to it as the “expert-defined model”. The expert-defined model is a 41-parameter nested logit; its specification is provided in Table 10 in the Appendix. Fig. 5 illustrates the results generated by our algorithm in this instance. Its comparison with Fig. 4 suggests that providing an initial model allows our algorithm to focus the search in its surroundings: the two Pareto fronts are similar, but the latter displays a more thorough search in the “crook of the elbow”. As a result, the expert-defined model is outperformed by the 30-parameter Pareto-optimal model—in terms of log likelihood, −37’957.8 against $-$ 37’917.09—whereas the 41-parameter Pareto-optimal model yields a log likelihood of $-$ 37’418.3. That constitutes an improvement of 539.5 in log likelihood for the same number of estimated parameters.

5.4. Closing remarks

The conducted experiments empirically demonstrate the validity and potential of our methodology. They show that the algorithm works well, in the sense that it provides a set of meaningful and potentially good models to the analyst in a period of time that is way shorter than a classical, “manual” model development would take. The presented results confirm the ability of the algorithm to thoroughly explore the trade-off between model parsimony and goodness of fit in different setups and with different dataset sizes. The generated Pareto fronts provide a broad understanding of the considered specification problems; their richness therefore legitimizes the use of a multi-objective approach.

6. Conclusion

In this paper, we propose an algorithm for the assisted specification of discrete choice models. We define a multi-objective combinatorial optimization problem with two conflicting objectives and make use of a variable neighborhood metaheuristic to generate solutions in a way that mimics human modelers. The validity of the proposed algorithm is empirically demonstrated using publicly available mode choice datasets. Out-of-sample validation shows that our algorithm generates sets of high-quality specifications in reasonable amounts of time. This approach can effectively assist analysts in the task of model development and provide them with relevant insights about the dataset under consideration and its associated modeling possibilities.

Intended future work includes the development of a system that allows “fixing” part of the utility specification while letting the algorithm “enrich” it freely. It is known from empirical studies that some key variables of well-studied choice problems should be considered in a particular manner; compliance with the existing literature and improvements in terms of computational effort could therefore be obtained by introducing appropriate restrictions to the search space. Additional investigation could also consist in using other objectives—such as out-of-sample validation—in the multi-objective problem, in addition or instead of the two proposed in this paper. Finally, another relevant direction of search consists in extending the framework to more advanced model structures. Our methodology is already applicable to nested, cross-nested and mixtures of logits, as it deals with the specification of utility functions. An extension would consist in providing algorithmic assistance to the definition of latent variables and classes in integrated latent variable and choice models. In such cases, computational cost and overall runtime could be reduced by using heuristics based on partial estimation, where the maximum likelihood algorithm is prematurely interrupted if the model under consideration is not promising.

CRediT authorship contribution statement

Nicola Ortelli: Conceptualization, Methodology, Software, Validation, Investigation, Writing - original draft. **Tim Hillel:** Conceptualization, Methodology, Writing - review & editing. **Francisco C. Pereira:** Conceptualization, Writing - review & editing. **Matthieu de Lapparent:** Conceptualization, Resources, Writing - review & editing, Supervision. **Michel Bierlaire:** Conceptualization, Validation, Resources, Writing - review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix. Selected Model Specifications

Swissmetro

Table 4
Swissmetro. Specification of the AIC-optimal model.

Model: nested existing (142 parameters)

Group	Form	Transformation	Segmentation
ACSs	—	—	Age Luggage Gender
Travel time	Alt.-spec.	—	Class GA Luggage Gender
Travel cost	Alt.-spec.	Piecewise (4 seg.)	Income
Headway	Generic	—	—

Table 5
Swissmetro. Specification of the BIC-optimal model.

Model: nested existing (28 parameters)

Group	Form	Transformation	Segmentation
ASCs	–	–	Gender
Travel time	Alt.-spec.	Logarithm	GA
Travel cost	Alt.-spec.	Piecewise (4 seg.)	Class
Headway	Alt.-spec.	Square-root	Age

Table 6
Swissmetro. Specification of the OOS-optimal model.

Model: logit (27 parameters)

Group	Form	Transformation	Segmentation
ASCs	–	–	Gender
Travel time	Alt.-spec.	Logarithm	GA
Travel cost	Alt.-spec.	Piecewise (4 seg.)	Class
Headway	Alt.-spec.	Square-root	Age

Table 7
Semi-Synthetic Data. Specification of the true model.

Model: nested existing (21 parameters)

Group	Form	Transformation	Segmentation
ASCs	–	–	Age
Travel time	Alt.-spec.	Logarithm	–
Travel cost	Alt.-spec.	Logarithm	Class
Headway	Generic	–	–

Table 8
Semi-Synthetic Data. Specification of the 18-parameter model.

Model: nested existing (18 parameters)

Group	Form	Transformation	Segmentation
ASCs	–	–	Age
Travel time	Alt.-spec.	Logarithm	–
Travel cost	Alt.-spec.	Logarithm	–
Headway	Alt.-spec.	Square	–

Table 9
Semi-Synthetic Data. Specification of the 22-parameter model.

Model: nested existing (22 parameters)			
Group	Form	Transformation	Segmentation
ASCs	–	–	Age
Travel time	Alt.-spec.	Logarithm	–
Travel cost	Alt.-spec.	Logarithm	–
Headway	Alt.-spec.	–	Luggage

Table 10
LPMC. Specification of the expert-defined model.

Model: nested motorized (41 parameters)			
Group	Form	Transformation	Segmentation
ASCs	–	–	License Car owner
Distance	Alt.-spec.	–	Age
Travel time	Alt.-spec.	–	–
Travel cost	Generic	–	Purpose
Transfers	Generic	–	Winter
Variability	Alt.-spec.	–	–

References

Akaike, H., 1974. A new look at the statistical model identification. IEEE Trans. Automat. Contr. 19 (6), 716–723.

Alwosheel, A., van Cranenburgh, S., Chorus, C.G., 2019. ‘computer says no’ is not enough: using prototypical examples to diagnose artificial neural networks for discrete choice analysis. J.Choice.Model 33, 100186.

Antonini, G., Gioia, C., Frejinger, E., 2007. Swissmetro: description of the data.

Bentz, Y., Merunka, D., 2000. Neural networks and the multinomial logit for brand choice modelling: a hybrid approach. J. Forecast. 19 (3), 177–200.

Bierlaire, M., 2018. Pandasbiogeme: a short introduction, *Technical Report*, TRANSP-OR 181219. Transport and Mobility Laboratory. ENAC, EPFL.

Bierlaire, M., 2020. A short introduction to Pandasbiogeme, *Technical Report*, TRANSP-OR 200605. Transport and Mobility Laboratory. ENAC, EPFL.

Bierlaire, M., Axhausen, K., Abay, G., 2001. The acceptance of modal innovation: the case of Swissmetro. In: Proceedings of the 1st Swiss Transportation Research Conference.

Brathwaite, T., Vij, A., Walker, J.L., 2017. Machine learning meets microeconomics: the case of decision trees and discrete choice arXiv preprint arXiv:1711.04826.

Breiman, L., 1995. Better subset regression using the nonnegative garrote. Technometrics 37 (4), 373–384.

Brusco, M.J., 2014. A comparison of simulated annealing algorithms for variable selection in principal component analysis and discriminant analysis. Comput. Stat. Data Anal. 77, 38–53.

Burnham, K.P., Anderson, D.R., 2002. A practical information-theoretic approach, *Model Selection and Multimodel Inference*, second ed. Springer, New York.

Burnham, K.P., Anderson, D.R., 2004. Multimodel inference: understanding AIC and BIC in model selection. Socio. Methods Res. 33 (2), 261–304.

Chuang, L.-Y., Yang, C.-H., Yang, C.-H., 2009. Tabu search and binary particle swarm optimization for feature selection using microarray data, 16 (12), 1689–1703.

Claeskens, G., Hjort, N.L., 2003. The focused information criterion. J. Am. Stat. Assoc. 98 (464), 900–916.

Claeskens, G., Hjort, N.L., 2008. Model selection and model averaging, Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

De Carvalho, M., Dougherty, M., Fowkes, A., Wardman, M., 1998. Forecasting travel demand: a comparison of logit and artificial neural network methods. J. Oper. Res. Soc. 49 (7), 717–722.

Doornik, J.A., 2008. Encompassing and automatic model selection. Oxf. Bull. Econ. Stat. 70, 915–925.

Friedman, J.H., 1991. Multivariate adaptive regression splines, *The Annals of Statistics*, pp. 1–67.

Hagenauer, J., Helbich, M., 2017. A comparative study of machine learning classifiers for modeling travel mode choice. Expert Syst. Appl. 78, 273–282.

Han, Y., Zegras, C., Pereira, F.C., Ben-Akiva, M., 2020. A neural-embedded choice model: TasteNet-MNL modeling taste heterogeneity with flexibility and interpretability arXiv preprint arXiv:2002.00922.

Hillel, T., 2019a. London Passenger Mode Choice: Description of the Data.

Hillel, T., 2019b. *Understanding Travel Mode Choice: A New Approach for City Scale Simulation*, PhD Thesis. University of Cambridge.

Hillel, T., Bierlaire, M., Elshafie, M., Jin, Y., 2019. Weak teachers: assisted specification of discrete choice models using ensemble learning.

Hillel, T., Elshafie, M., Jin, Y., 2018. Recreating passenger mode choice-sets for transport simulation: a case study of London, UK. Proc. Inst.Civ. Eng.Smart.Infrastruct. Construct 171 (1), 29–42.

Hruschka, H., Fettes, W., Probst, M., Mies, C., 2002. A flexible brand choice model based on neural net methodology a comparison to the linear utility multinomial logit model and its latent class extension. OR Spectrum 24 (2), 127–143.

Koppelman, F.S., Bhat, C., 2006. A self instructing course in mode choice modeling: multinomial and nested logit models.

Lhéritier, A., Bocamazo, M., Delahaye, T., Acuna-Agost, R., 2019. Airline itinerary choice modeling using machine learning. J.Choice.Model 31, 198–209.

Lin, S.-W., Lee, Z.-J., Chen, S.-C., Tseng, T.-Y., 2008. Parameter determination of support vector machine and feature selection using simulated annealing approach. Appl. Soft Comput. 8 (4), 1505–1512.

McFadden, D., 1974. The measurement of urban travel demand. J. Publ. Econ. 3 (4), 303–328.

Mladenovic, N., Hansen, P., 1997. Variable neighborhood search. Comput. Oper. Res. 24 (11), 1097–1100.

Oh, I.-S., Lee, J.-S., Moon, B.-R., 2004. Hybrid genetic algorithms for feature selection 26 (11), 1424–1437.

Pacheco, J., Casado, S., Núñez, L., 2009. A variable selection method based on tabu search for logistic regression models. Eur. J. Oper. Res. 199 (2), 506–511.

Paredes, M., Hemberg, E., O'Reilly, U.-M., Zegras, C., 2017. Machine learning or discrete choice models for car ownership demand estimation and prediction?. In: 2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), IEEE, pp. 780–785.

Paz, A., Arteaga, C., Cobos, C., 2019. Specification of mixed logit models assisted by an optimization framework. J.Choice.Model 30, 50–60.

Pereira, F.C., 2019. Rethinking travel behavior modeling representations through embeddings arXiv preprint arXiv:1909.00154.

Rodrigues, F., Ortelli, N., Bierlaire, M., Pereira, F., 2019. Bayesian automatic relevance determination for utility function specification in discrete choice models arXiv preprint arXiv:1906.03855.

Schoenberg, R., 1997. Constrained maximum likelihood. Comput. Econ. 10 (3), 251–266.

Schwarz, G., 1978. Estimating the dimension of a model. Ann. Stat. 6 (2).

Sifringer, B., Lurkin, V., Alahi, A., 2018. Let me not lie: learning multinomial logit arXiv preprint arXiv:1812.09747.

Soufan, O., Kleftogiannis, D., Kalnis, P., Bajic, V.B., 2015. DWFS: a wrapper feature selection tool based on a parallel genetic algorithm. PloS one 10 (2).

Tang, L., Xiong, C., Zhang, L., 2015. Decision tree method for modeling travel mode switching in a dynamic behavioral process. Transport. Plann. Technol. 38 (8).

Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. J. Roy. Stat. Soc. B 267–288.

Torres, C., Hanley, N., Riera, A., 2011. How wrong can you be? implications of incorrect utility function specification for welfare measurement in choice experiments. J. Environ. Econ. Manag. 62 (1), 111–121.

Van Der Pol, M., Currie, G., Kromm, S., Ryan, M., 2014. Specification of the utility function in discrete choice experiments. Value Health 17 (2), 297–301.

Wang, F., Ross, C.L., 2018. Machine learning travel mode choices: comparing the performance of an extreme gradient boosting model with a multinomial logit model. Transport. Res. Rec. 2672 (47), 35–45.

Wong, M., Farooq, B., 2019. ResLogit: a residual neural network logit model arXiv preprint arXiv:1912.10058.

Zeileis, A., Hothorn, T., Hornik, K., 2008. Model-based recursive partitioning. J. Comput. Graph Stat. 17 (2), 492–514.

Zhao, X., Yan, X., Yu, A., Van Hentenryck, P., 2018. Modeling stated preference for mobility-on-demand transit: a comparison of machine learning and logit models arXiv preprint arXiv:1811.01315.