A Benders decomposition for maximum simulated likelihood estimation of advanced discrete choice models

Tom Haering

Claudia Bongiovanni

Michel Bierlaire

STRC conference paper 2022 May 23, 2022

STRC | 22nd Swiss Transport Research Conference
Monte Verità / Ascona, May 18-20, 2022

A Benders decomposition for maximum simulated likelihood estimation of advanced discrete choice models

Tom Haering
TRANSP-OR
ETH Lausanne
CH-1003 Lausanne
tom.haering@epfl.ch

Claudia Bongiovanni
TRANSP-OR
ETH Lausanne
CH-1003 Lausanne
claudia.bongiovanni@epfl.ch

Michel Bierlaire
TRANSP-OR
ETH Lausanne
CH-1003 Lausanne
michel.bierlaire@epfl.ch

May 23, 2022

Abstract

In this paper, we formulate a mixed integer linear program (MILP) for the simulated maximum likelihood estimation (MLSE) problem and devise a Benders decomposition approach to speed up the solution process. This framework can be applied to any advanced discrete choice model and exploits total unimodularity to keep the master problem linear in the decomposition. The proposed decomposition approach is benchmarked against the original MILP formulation and PandasBiogeme. Computational experiments are performed on a binary logit mode choice model with up to 200 respondents. Results show that the Benders decomposition approach solves instances on average 35 and up to 100 times faster than the MILP while maintaining high quality solutions. We furthermore give detailed descriptions of ideas for future enhancements of the approach.

Keywords

benders decomposition, discrete choice, maximum likelihood estimation, mixed integer linear programming, simulation

Contents

List of Tables

List of Figures

1 Background

Maximum likelihood estimation (MLE) is a broadly used method to estimate the parameters, given observed data (Myung, 2003). It finds its use in many areas of mathematical statistics (Sur and Candès, 2019), physics (Hauschild and Jentschel, 2001), machine learning (Goodfellow *et al.*, 2016) and discrete choice modeling (Ben-Akiva and Bierlaire, 2003). The latter specifically relies on the use of MLE to estimate the optimal parameters of convex and non-convex discrete choice models (Bierlaire, 1998). The state-of-the-art approach for the estimation process is to use classical non linear optimization algorithms, that require that the likelihood function to be continuous in the unknown parameters, and to have a closed form, which is also concave. These requirements are actually verified only for the estimation of logit models with linear-in-parameters utility functions. More complex models (the nested logit model, the cross-nested logit model, choice models with latent classes) are associated with non concave likelihood functions. Non linear optimization algorithms may easily be trapped in a local optimum. Models based on mixtures (including choice models with latent variables) are associated with a likelihood function that has no closed form, and involves complex integrals. Monte-Carlo simulation is therefore required to approximate it (Train, 2009). Discrete specification decisions, such as including an explanatory variable or not, or including an alternative in a nest) are handled "by hand", in the sense that all possible specifications are explicitly enumerated, and each of them estimated separately. A general implementation approach for MSLE, that is able to handle all of these cases, has been proposed in Fernández Antolín (2018), where the problem is formulated as a mixed integer linear program (MILP). The approach relaxes any assumption on the specific shape of the error term distribution and instead only assumes that it is possible to take draws. This allows it to be flexibly applied to any model. The approach is especially well suited to handle integral estimation parameters, like class membership variables in a latent class model or to implement an assisted / automatic specification approach (Fernandez-Antolin *et al.*, 2018). With a sufficiently large number of draws, the MILP formulation guarantees convergence to global optimal solutions. However, since the complexity of MILP scales exponentially with the number of draws, the approach can currently only be applied to solving small-scale instances, i.e., with few individuals and alternatives (Pacheco *et al.*, 2021).

Recent research studies from the literature have shown that the computational complexity resulting from simulation-based optimization approaches can be overcome by mathematical decomposition techniques (e.g. Bront *et al.* 2009; Koch *et al.* 2017). Mathematical decomposition is an optimization field that aims at exploiting specific structural properties of decision-making problems to speed-up the solution process by parallelization, size reduction, and simplification (Conejo *et al.*, 2006). Discrete decision-making problems can be characterized broadly by the two following structural properties: (i) a set of complicating decision variables, (ii) a set of complicating constraints. Both of these properties have been addressed through specific mathematical decomposition techniques. Namely, (i) Benders' decomposition (Rahmaniani *et al.*, 2020), (ii) column generation (Desaulniers *et al.*, 2006), and (iii) Langrangian decomposition (Fisher, 1981). For example, Benders' decomposition has been widely applied to relax complicating variables in the integrated airline scheduling problem (e.g. Cordeau *et al.* 2001; Papadakos 2009). For the same problem, (Yan *et al.*, 2020) proposed a new choice-based reformulation that employs mathematical decomposition to reduce the size of the problem. Bortolomiol *et al.* (2021) implemented a Branch-and-Benders-Cut procedure for tackling a discrete assortment pricing problem. Similarly to Benders, Lagrangian decomposition has been widely applied to relax complicating constraints in network design problems (e.g. Heidari-Fathian and Pasandideh 2018; Gendron 2019; Shan *et al.* 2020) or facility location problems (e.g. Alenezy 2020; Yu *et al.* 2017). A recent choice-based network design problem employing Lagrangian relaxation to determine hub locations was proposed in Tiwari *et al.* (2021). Pacheco *et al.* (2018) employ a Lagrangian decomposition for choice-based optimization problems, by first duplicating the complicating variables for each simulation scenario and then adding constraints that force the duplicates to take equal values over all scenarios. In order to strengthen the decomposition and prevent loss of structure, scenarios were clustered into groups. However, the approach yielded only limited success, leading to the conclusion that this decomposition technique might not be the most appropriate for this problem. Finally, column generation has been widely applied to solve large-scale scheduling and problems (e.g. Desaulniers *et al.* 2002; Feillet 2010; Boyer *et al.* 2014). A recent choice-based multi-project scheduling and staffing problem employing column generation was proposed in Van Den Eeckhout *et al.* (2021).

In this work, we formulate a Benders' decomposition approach, which speeds-up the MILP solution process for the MSLE and enables to scale-up the tackled instances. Our designed Benders' decomposition approach exploits total unimodularity to keep the master problem linear, thus eliminating the bottleneck in computational time usually associated with Benders decomposition. The proposed approach is benchmarked against the full MILP and PandasBiogeme Bierlaire (2020). Although the method is general, the developed approach is currently being investigated on a binary logit mode choice model with up to 200 respondents. Initial results show that, while being very competitive in terms of computational time, our Benders' decomposition approach produces small optimality gaps. We present the approach in section 2 and show results, together with investigations of different remedies for the optimality gaps in Section 3.

2 Methodology

In this section, we formally introduce an MILP formulation for the MSLE problem, based on the work in Fernández Antolín (2018), and a problem-specific Benders decomposition approach. Without loss of generality, the formulation is presented in the context of a multinomial logit formulation, with examples on how to extend it to other model classes, such as probit and latent class models.

2.1 MILP formulation

Figure 1: MSLE as an MILP

$$\max_{\beta,\omega,s,z,U,H} \sum_n \sum_i y_{in} z_{in}$$

s.t.

$$\begin{aligned}
\sum_i \omega_{inr} &= 1 && (\mu_{nr}) \\
H_{nr} &= \sum_i U_{inr}\omega_{inr} && (\zeta_{nr}) \\
H_{nr} &\geq U_{inr} && (\alpha_{inr}) \\
s_{in} &= \sum_r \omega_{inr} && (\theta_{in}) \\
z_{in} &\leq L_r - K_r s_{in} && (\xi_{inr}) \\
U_{inr} &= \sum_k \beta_k x_{ink} + \varepsilon_{inr} && (\kappa_{inr}) \\
\omega &\in \{0,1\}^{INR} \\
U,H &\in \mathbb{R}^{INR} \\
s,z &\in \mathbb{R}^{IN} \\
\beta &\in \mathbb{R}^{K}
\end{aligned}$$

Consider a set of $n = \{1, \dots, N\}$ individuals choosing exactly one alternative among a set of $i = \{1, \dots, I\}$ alternatives. The data underlying the problem consists of such choices, depicted by a binary input parameter y_{in}. For each alternative i we define a utility function $U_{in} = U_{in}(\beta, \varepsilon_{in})$, where β is a vector of unknown parameters, and ε_{in} is a random vector. We assume the utility to be linear in β and the distribution of ε such that we can generate draws from it. These error term draws are necessary to get a linear specification, and we denote them as ε_{inr}, for R scenarios $r \in \{1, \dots, R\}$. This allows us to write the utilities U_{inr} in a deterministic way, meaning that they only depend on β. For this exposition, we write $U_{inr} = \sum_k \beta_k x_{ink} + \varepsilon_{inr}$, but more complex specifications are possible, if linearity in β is maintained. Denote by ω_{inr} the binary decision variable that indicates whether individual n chooses alternative i in scenario r. Random utility theory dictates that, in each scenario r, each individual n selects the alternative i corresponding to the maximal utility U_{inr}, i.e. $\omega_{inr} = 1 \Leftrightarrow U_{inr} = \max_j U_{jnr}$. The objective is to maximize the likelihood function, given by $\prod_n \prod_i P_n(i)^{y_{in}}$, where $P_n(i)$ represents the probability of individual n choosing alternative i. The choice probabilities are approximated by $P_n(i) \approx \frac{1}{R} \sum_r \omega_{inr}$ and are guaranteed to converge to the real probabilities with a sufficiently large number of scenarios R (Pacheco *et al.*, 2021). Taking the log of the likelihood and replacing $P_n(i)$ by its estimator yields an objective that still contains the nonlinear term $\ln(\sum_r \omega_{inr})$. This issue is tackled by introducing the auxiliary decision variable $s_{in} = \sum_r \omega_{inr}$, which is defined in constraints (θ_{in}). Similarly, an auxiliary variable z_{in} is introduced to represent the piece-wise linearization of the logarithm. The latter is defined in Constraints (ξ_{inr}), where $L_r = (1+r)\ln(r) - r\ln(1+r)$ and $K_r = \ln(r) - \ln(1+r)$ are constants representing intercepts and slopes used for the linearization. With such pre-processing steps and by ignoring the constant term $-N\ln(R)$, the objective of the problem can be rewritten as stated in Formulation 1. The rest of the constraints model individual choices. Constraints (μ_{nr}) guarantee that only one alternative can be chosen per individual and scenario. Constraints (κ_{inr}) model the utility of each alternative i for individual n in scenario r, i.e. U_{inr}. Constraints (ζ_{nr}), which can be easily linearized using a standard big-M approach, and constraints (α_{inr}) ensure that the choice being made corresponds to the one with the highest utility. Note that Formulation 1 is characterized by the complicating binary decision variables ω.

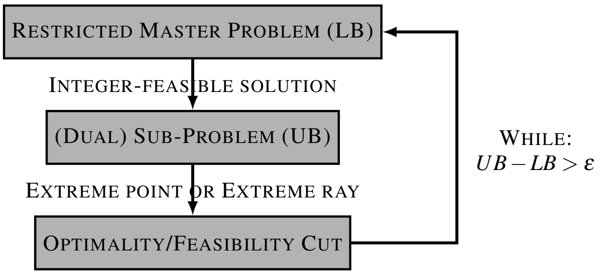
The MILP formulation can be easily adapted to other model speficiations. For example, in order to tackle a probit model, it is sufficient to add the cholesky factor of the covariance matrix (Dow and Endersby, 2004) in the right-hand side of constraints (κ_{inr}). Note that this transformation increases the number of parameters to be estimated by $\frac{I(I+1)}{2}$. Similarly, in order to tackle a latent class model, it is sufficient to add a class membership indicator γ_{cn} where c is the class index. For each class c, the constraints corresponding to making the best choice ω_{cinr} for that class are duplicated, and finally a global choice variable is defined as $\omega_{inr} = \sum_c \omega_{cinr}\gamma_{cn}$.

2.2 Benders decomposition approach

Combinatorial optimization problems that are characterized by complicating variables are typically tackled by a Benders decomposition approach (Benders, 1962). The logic of this approach lies in temporarily fixing the (usually integral) complicating variables to give rise to much simpler linear sub-problems to be solved. With this premise, a mathematical program is therefore split into two problems: (i) a problem containing all integer decision variables and constraints (i.e. the restricted master problem), and (ii) a problem containing all continuous decision variables and constraints (i.e. the sub-problem). The master problem is solved, and an integer-feasible solution is found, giving a lower bound on the objective value. This integer solution is successively used to solve the dualized linear sub-problem and produce: (i) feasibility cuts, if the subproblem is infeasible with the integer solution from the master problem; or (ii) optimality cuts, if the subproblem provides an optimal solution given the fixed integer solution from the master problem. The produced cuts are fed to the restricted master problem to tighten the upper bound, after which the restricted master problem is re-solved to provide a new integer solution. The whole Benders' process repeats up until a satisfactory solution has been found (i.e. the gap between upper and lower bound is small enough). A flowchart describing the steps for Benders' decomposition is shown in Figure 2. For a review on Benders' decomposition, we refer to Rahmaniani *et al.* (2017).

In most applications, the complicating variables are integral, resulting in the need to solve an integral master problem at each iteration. This makes Benders notorious for its slow convergence (Rahmaniani *et al.*, 2020). In our case, we can use an elegant trick to avoid this issue: by identifying the continuous estimation parameters β as the complicating variables and fixing them in the subproblem, the utilities of all the alternatives become fixed as well. Thus the problem of choosing the highest utility alternative simplifies to a knapsack problem, which is totally unimodular. This mathematical property allows us to drop the integrality constraints on the choice variables. Formulation 3 and Formulation 4 give the respective definitions of the primal and dual of the subproblem, while Formulation 5 describes the master problem.

Figure 2: Flowchart for Benders' decomposition


Restricted Master Problem (LB)
Integer-feasible solution
(Dual) Sub-Problem (UB)
Extreme point or Extreme ray
Optimality/Feasibility Cut
While:
$UB - LB > \varepsilon$

$$\min_{\beta,\omega,\chi,\eta,s,z,H} -\sum_n \sum_i y_{in} z_{in}$$

s.t.

$$\begin{aligned}
\sum_i \omega_{inr} &= 1 && (\mu_{nr})\\
\sum_k \beta_k x_{ink} - H_{nr} &\leq -\varepsilon_{inr} && (\alpha_{inr})\\
H_{nr} - \sum_{ik} \eta_{inrk} x_{ink} &\leq \sum_i \omega_{inr} \varepsilon_{inr} && (\zeta_{nr})\\
\chi_{inr} + \omega_{inr} &= 1 && (\pi_{inr})\\
\eta_{inrk} + \beta_k^{\text{fixed}} \chi_{inr} &= \beta_k^{\text{fixed}} && (\lambda_{inrk})\\
\beta_k - \sum_i \eta_{inrk} &= 0 && (\varphi^{\beta}_{nrk})\\
s_{in} - \sum_r \omega_{inr} &= 0 && (\theta_{in})\\
z_{in} + K_r s_{in} &\leq L_r && (\xi_{inr})\\
\omega, \chi, s &\in \mathbb{R}_{\geq 0}\\
\beta, \eta, z, H &\in \mathbb{R}
\end{aligned}$$

Figure 3: MSLE - Primal subproblem

$$\begin{aligned}
\max_{\mu,\alpha,\zeta,\mu,\lambda,\varphi^{\beta},\theta,\xi} \quad & \sum_{nr} \mu_{nr} - \sum_{inr} \varepsilon_{inr} \alpha_{inr} + \sum_{inr} \pi_{inr}\\
& + \sum_{inrk} \beta_k^{\text{fixed}} \lambda_{inrk} + \sum_{inr} L_r \xi_{inr}
\end{aligned}$$

s.t.

$$\begin{aligned}
\mu_{nr} - \zeta_{nr} \varepsilon_{inr} + \pi_{inr} - \theta_{in} &\leq 0 && (\omega_{inr})\\
\pi_{inr} + \sum_k \beta_k^{\text{fixed}} \lambda_{inrk} &\leq 0 && (\chi_{inr})\\
-\sum_i \alpha_{inr} + \zeta_{nr} &= 0 && (H_{nr})\\
-\zeta_{nr} x_{ink} + \lambda_{inrk} - \varphi^{\beta}_{nrk} &= 0 && (\eta_{inrk})\\
\theta_{in} + \sum_r K_r \xi_{inr} &\leq 0 && (s_{in})\\
\sum_r \xi_{inr} &= -y_{in} && (z_{in})\\
\sum_{inr} \alpha_{inr} x_{ink} + \sum_{nr} \varphi^{\beta}_{nrk} &= 0 && (\beta_k)\\
\mu, \pi, \lambda, \theta, \varphi^{\beta} &\in \mathbb{R}\\
\alpha, \zeta, \xi &\in \mathbb{R}_{\leq 0}
\end{aligned}$$

Figure 4: MSLE - Dual subproblem

As the linearization of Constraint (ζ_{nr}) using a big-M approach no longer works when integrality constraints are relaxed, the formulation in the primal subproblem needs to be modified: The product $\eta_{inrk} = \omega_{inr}\beta_k$ is modeled directly using Constraints (π_{inr}), (λ_{inrk}) and (φ^{β}_{nrk}). This formulation is equivalent to Formulation 1. It is important to mention that, in order to guarantee total unimodularity of the primal, information about β^{fixed} had to be kept in its coefficient matrix, which implies it also being contained in the matrix of the dual, i.e. Constraints (χ_{inr}). This means the feasible region of the dual subproblem is not constant over iterations, which might distort the Bender cuts (Rahmaniani *et al.*, 2020). Lastly, both the primal and the dual models are fully decomposable on the individuals n, as individuals select alternatives independently from each other.

Finally, the master problem reduces to finding optimal values for the estimation parameters β. For each β^{fixed}, after solving the dual subproblem, a Benders cut of the same type as Constraint (1) is added. The parameters of the Benders cuts are determined by the achieved objective $\mathscr{L}^*$ and $\phi^*_{nk} = \sum_{ir} \lambda^*_{inrk}$. Each optimal objective value of the master problem serves as a new lower

Figure 5: MSLE - Master problem

$$\min_{\mathscr{L},\beta} \mathscr{L}$$

s.t.

$$\begin{aligned}
\mathscr{L} &\geq \mathscr{L}^{*} + \sum_{n}\sum_{k}\phi^{*}_{nk}(\beta_k - \beta_k^{\text{fixed}}) && (1)\\
\mathscr{L} &\geq \mathscr{L}^{\text{best}} && (2)\\
\mathscr{L} &\in \mathbb{R}\\
\beta &\in \mathbb{R}^{K}
\end{aligned}$$

bound on the objective, enforced in Constraint (2).

3 Results and discussion

3.1 Application to a mode choice problem

Our approach is tested on a binary logit model. A mode choice problem between two alternatives, public transport (pt) and car, is considered. The utilities of the alternatives consist of a systematic part V and a stochastic part ε. The utilities of the alternatives for individual n are:

$$\begin{aligned}
U_{\text{car, }n} &= \beta_{\text{time}} \cdot \text{traveltime}_{\text{car, }n} + \varepsilon_{\text{car, }n}\\
U_{\text{pt, }n} &= \beta_{\text{time}} \cdot \text{traveltime}_{\text{pt, }n} + \varepsilon_{\text{pt, }n}
\end{aligned}$$

where as the stochastic part is defined by independent and identically distributed random error terms following a standard extreme value distribution, i.e. $\varepsilon \sim$ i.i.d. $\text{EV}(0,1)$. The dataset is extracted from revealed preference data on mode choice collected in 1987 for the Netherlands Railways, consisting of 228 respondents (CASE, 2017). Experiments are performed using GUROBI 9.5.0 (Gurobi Optimization, LLC, 2021) on a 2.6 GHz 6-Core Intel Core i7 processor with 16 GB of RAM, with a three hour time limit per instance. Our proposed Benders approach is benchmarked against PandasBiogeme (Bierlaire, 2020) and the full MILP, in terms of objective values and runtimes. Biogeme's objective function is the Log-Likelihood ($LL = \ln(\prod_n \prod_i P_n(i)^{y_{in}})$) , which is approximated by the simulated Log-Likelihood (sLL), the MILP objective. For the purpose of comparison, the LL is also evaluated for the decomposition and the MILP using the estimated parameters. We take random subsets of individuals from the population to get instances that are manageable for the MILP.

Table 1 shows the comparison between the decomposition and the full MILP in terms of *sLL* and computation times, while Table 3 shows the results in terms of *LL*. We highlight the following: (i) the decomposition solves the problem on average 35 and up to 100 times faster, (ii) comparing the optimal solution values for the full MILP and our decomposition reveals gaps in optimality, which are small for the objective value but larger for the estimated parameters, and (iii) increasing the number of draws reduces the optimality gap between the exact solution (PandasBiogeme) and the approximation (MILP and decomposition).

Table 1: Comparing our decomposition method with the full MILP in terms of *sLL* and runtime (N = population size, R = number of draws, sLL = simulated Log-Likelihood, M = MILP, D = decomposition, T = time in sec.)

N	R	sLL-M	sLL-D	Gap [%]	T-M	T-D
20	50	-12.607	-12.658	-0.40	64.942	10.061
20	100	-12.212	-12.258	-0.38	403.694	9.902
20	200	-12.283	-12.648	-2.97	1117.064	16.939
50	50	-30.848	-31.030	-0.59	286.679	29.780
50	100	-30.461	-31.040	-1.90	1558.604	65.006
50	200	-30.566	-30.692	-0.41	5375.655	98.206
100	50	-65.204	-65.801	-0.92	2820.229	28.781
100	100	-65.784	-67.419	-2.49	4346.067	274.163
100	200	-65.699	-66.018	-0.49	10800+	295.741
200	50	-123.551	-124.027	-0.39	1476.185	120.579
200	100	-124.000	-124.243	-0.20	10800+	327.253
200	200	-124.707	-124.106	0.48	10800+	1262.755

Table 2: Comparing our decomposition method with the full MILP and PandasBiogeme in terms of *LL* (N = population size, R = number of draws, LL = Log-Likelihood, Biog = PandasBiogeme, M = MILP, D = decomposition)

N	R	LL-Biog	LL-M	Gap [%]	LL-D	Gap [%]
20	50	-12.303	-12.444	-1.15	-12.493	-1.55
20	100	-12.303	-12.395	-0.75	-12.411	-0.88
20	200	-12.303	-12.378	-0.61	-12.463	-1.30
50	50	-30.265	-30.326	-0.20	-30.683	-1.38
50	100	-30.265	-30.326	-0.20	-30.481	-0.72
50	200	-30.265	-30.325	-0.20	-30.283	-0.06
100	50	-64.883	-64.898	-0.02	-65.396	-0.79
100	100	-64.883	-64.883	0.00	-66.031	-1.77
100	200	-64.883	-64.893	-0.02	-64.925	-0.06
200	50	-122.689	-122.735	-0.04	-122.690	0.00
200	100	-122.689	-122.920	-0.19	-122.739	-0.04
200	200	-122.689	-123.342	-0.53	-122.721	-0.03

Table 3: Comparing our decomposition method with the full MILP and PandasBiogeme in terms of β (N = population size, R = number of draws, Biog = PandasBiogeme, M = MILP, D = decomposition)

N	R	Beta-Biog	Beta-MILP	Gap [%]	Beta-D	Gap [%]
20	50	-1.558	-1.048	32.72	-0.97	37.71
20	100	-1.558	-1.143	26.62	-1.11	28.77
20	200	-1.558	-1.182	24.11	-2.16	-38.67
50	50	-1.41	-1.223	13.26	-0.935	33.64
50	100	-1.41	-1.223	13.26	-1.783	-26.46
50	200	-1.41	-1.223	13.22	-1.307	7.26
100	50	-0.948	-0.889	6.23	-0.612	35.48
100	100	-0.948	-0.943	0.53	-0.451	52.4
100	200	-0.948	-0.899	5.21	-0.85	10.33
200	50	-1.31	-1.39	-6.11	-1.322	-0.92
200	100	-1.31	-1.49	-13.73	-1.393	-6.33
200	200	-1.31	-1.021	22.04	-1.377	-5.11

Although Benders decomposition is an exact approach, our formulation contains mathematical aspects that may currently prevent the convergence to the real global optimum. As mentioned in the methodology, a possible explanation for the deviations is the fact that information about the master variables is maintained in the coefficient matrix of the dual. Other explanations include numerical issues, stemming for example from the linearization of the logarithm or the way certain solvers handle specific constraints.

3.2 Application to a continuous pricing problem

The suspicion that the log-linearization might influence the Benders cuts leads us to an immediate first attempt to fix it: apply the framework to a choice based optimization problem that does not rely on piece-wise linearization of the objective. A very important problem that displays this characteristic is the *continuous pricing problem (CPP)*. It can be written as follows:

$$\max_{p,\omega,U,H} \sum_n \sum_r \sum_i \frac{1}{R} \theta_{in} p_i \omega_{inr}$$

s.t.

$$\begin{aligned}
\sum_i \omega_{inr} &= 1 && (\mu_{nr}) \\
H_{nr} &= \sum_i U_{inr} \omega_{inr} && (\zeta_{nr}) \\
H_{nr} &\geq U_{inr} && (\alpha_{inr}) \\
U_{inr} &= \sum_{k \neq l} \beta_k x_{ink} + \beta_l p_i + \varepsilon_{inr} && (\kappa_{inr}) \\
\omega &\in \{0,1\} \\
p, U, H &\in \mathbb{R}
\end{aligned}$$

applying the exact same methodology as for the MSLE problem, we arrive at the following primal subproblem:

$$\max_{p,\omega,U,H} \sum_n \sum_r \sum_i \frac{1}{R} \theta_{in} p_i x_{inr}$$

s.t.

$$\begin{aligned}
\sum_i \omega_{inr} &= 1 && (\mu_{nr}) \\
\sum_k \beta_k x_{ink} - H_{nr} &\leq -\varepsilon_{inr} && (\alpha_{inr}) \\
H_{nr} - \sum_{ik} \eta_{inr} x_{ink} &\leq \sum_i \omega_{inr} \varepsilon_{inr} && (\zeta_{nr}) \\
\eta_{inr} &= p_i^{\text{fixed}} \omega_{inr} && (\lambda_{inr}) \\
\chi_{inr} + \omega_{inr} &= 1 && (\pi_{inr}) \\
\eta_{inr} + p_i^{\text{fixed}} \chi_{inr} &= p_i^{\text{fixed}} && (\lambda_{inr}) \\
p_i - \sum_i \eta_{inr} &= 0 && (\varphi_{nr}^{\beta}) \\
\omega, \chi &\in \mathbb{R}_{\geq 0} \\
p, \eta, H &\in \mathbb{R}
\end{aligned}$$

We can apply the same Netherlands mode choice data as in the MSLE problem to the CPP, with the slight change that the utilities are now defined as follows:

$$\begin{aligned}
U_{\text{car},\, n} &= \beta_{\text{time}} \cdot \text{traveltime}_{\text{car},\, n} + \beta_{\text{cost}} \cdot \text{cost}_{\text{car},\, n} + \varepsilon_{\text{car},\, n} \\
U_{\text{pt},\, n} &= \text{ASC}_{\text{pt}} + \beta_{\text{time}} \cdot \text{time}_{\text{pt},\, n} + \beta_{\text{cost}} \cdot \text{cost}_{\text{pt},\, n} + \varepsilon_{\text{pt},\, n}
\end{aligned}$$

We now first estimate the optimal β-parameters using Biogeme and then replace the attribute cost_{pt} by a variable p_{pt} which is to be optimized. The problem can be thus seen as the rail company deciding on the optimal price for their service. Running experiments with the above framework leads to the results shown in table 4. Clearly. the gaps are still existent, proving that the linearization of the log in the MSLE problem is not the (only) source of the issue.

Table 4: Comparing our decomposition method with the full MILP in terms of objective, price and runtime (N = population size, R = number of draws, obj = objective, P = price, M = MILP, NL = Nonlinear, D = decomposition, R = decomp along scen, T = time in sec.)

N	R	obj-MILP	obj-D	Gap [%]	P-MILP	P-D	Gap [%]	T-MILP	T-MILP (NL)	T-D	T-D (R)
20	50	216.407	209.196	3.33	28.475	30.764	-8.04	7	2	11	17
20	100	202.642	201.712	0.46	28.302	26.576	6.1	37	7	21	65
20	200	200.901	200.185	0.36	30.03	28.721	4.36	205	252	49	50
50	50	440.686	437.243	0.78	28.579	29.989	-4.94	55	24	27	20
50	100	431.088	426.669	1.03	28.99	27.778	4.18	241	235	62	51
50	200	429.605	429.108	0.12	28.574	28.655	-0.28	1022	2125	163	323
100	50	990.026	988.732	0.13	29.118	28.944	0.6	252	202	31	147
100	100	977.606	976.149	0.15	30.099	29.925	0.58	1224	595	69	315
100	200	978.589	976.932	0.17	30.106	30.185	-0.26	3039	3057	304	737
200	50	1906.696	1904.189	0.13	28.977	28.678	1.03	1144	766	65	351
200	100	1882.793	1877.641	0.27	29.277	30.052	-2.65	4104	3149	359	895
200	200	1873.964	1871.614	0.13	29.276	29.343	-0.23	10811	9672	690	1539

3.3 Large numbers of draws

We want to investigate the effects that a larger number of simulation draws has on the gap. This is interesting, among other reasons, because we can expect more complex advanced discrete choice models to require a higher number of draws to be accurately simulated (Pacheco, 2020). We solve the same problem instance as in the previous subsection, this time with up to R = 1000 draws. The number of participants has been kept at N = 50, in order to guarantee that the MILP still finds a feasible solution. The results (for the MSLE problem) can be seen in Table 5. It appears that a higher number of draws indeed has the effect of reducing the optimality gap, however, it has to be taken into account that very often the MILP will not converge in the given time limit of three hours for such large instances, making it more likely that we compare our output to solutions that are actually sub-optimal. On the other hand, by far the biggest part of computational time in the MILP is spent proving the lowerbound, i.e. optimality, meaning that the objective value of the solution often is already optimal, even if at the time limit there is a nonzero gap to be observed. Furthermore, for 800 draws and more, the MILP does not manage to find a feasible solution in three hours, indicating that our approach could be applied as a heuristic to find good feasible solution fast in large scale scenarios, which could then be used as warm start solutions for the MILP.

Table 5: Comparing our decomposition method with the full MILP in terms of *sLL* and runtime for large numbers of draws (N = population size, R = number of draws, sLL = simulated Log-Likelihood, M = MILP, D = decomposition, T = time in sec.)

N	R	sLL-M	sLL-D	Gap [%]	T-M	T-D
50	20	-29.417	-29.908	1.67	22	6
50	50	-29.294	-31.173	6.41	279	26
50	100	-28.885	-29.42	1.85	1375	42
50	150	-29.973	-30.092	0.4	2852	70
50	200	-30.091	-30.101	0.03	10800	131
50	250	-30.741	-30.775	0.11	10800	156
50	300	-30.837	-30.843	0.02	10800	133
50	400	-30.632	-30.638	0.02	10800	130
50	600	-30.479	-30.51	0.1	10800	289
50	800		-32.035		10800	319
50	1000		-30.523		10800	349

3.4 Decomposition as a warm start

Another interesting perspective is to investigate the effectiveness of using our fast, yet slightly suboptimal decomposition algorithm to generate a good warm start solution for the MILP. This allows to still speed up computation, while maintaining guaranteed optimality. The results (for the MSLE problem) can be seen in Table 6. We observe varied outcomes: For smaller instances (N $\leq$ 100) most instances show a speed-up of around 50%, however, outliers exist where the warm start solution influences the computational time negatively. For larger instances, in almost all cases the warm start solution does not seem to lead to a significant speed-up. Interesting to observe is also that we can have slightly better or worse solutions when solving the MILP with a warm start versus solving it without, even though the Branch & Bound algorithm used by Gurobi to solve the MILP guarantees optimality in both cases. This is an indication that the problem might be highly susceptible to numerical issues. This is insofar illuminating that the gaps we achieve with our decomposition method might be small enough to be ignored, especially with higher numbers of draws, as seen in subsection 3.3. In general the utility of using our method as a warm start solution is not clearly ascertainable, and thus requires further investigation.

Table 6: Comparing our decomposition method as a warm start with the full MILP in terms of *sLL* and runtime (N = population size, R = number of draws, sLL = simulated Log-Likelihood, M = MILP, D = decomposition, M+ = MILP with warm start, T = time in sec.)

N	R	sLL-M	sLL-D	Gap [%]	sLL-M+	Gap[%]	T-M	T-D	T-M+	rel. Speed-up [%]
50	50	-29.685	-29.842	-0.53	-29.685	0	578	38	290	49.87
50	100	-29.956	-30.954	-3.33	-29.956	0	2444	70	1308	46.48
50	150	-30.295	-30.356	-0.2	-30.295	0	1759	106	1846	-4.95
50	200	-30.352	-30.414	-0.2	-30.352	0	5074	131	10800	-112.86
100	50	-64.812	-65.555	-1.15	-64.886	-0.11	4181	60	630	84.93
100	100	-63.995	-64.118	-0.19	-63.918	0.12	10800	177	3890	63.98
100	150	-64.524	-64.551	-0.04	-64.524	0	10800	379	5184	52
100	200	-65.669	-66.48	-1.23	-65.309	0.55	10800	122	10800	0
200	50	-124.5	-124.5	0	-125.343	-0.68	6341	157	2988	52.88
200	100	-122.97	-122.823	0.12	-131.467	-6.91	10800	319	10800	0
200	150	-124.1	-123.973	0.11	-124.104	0	10800	464	10800	0
200	200	-124.34	-123.786	0.45	-124.468	-0.1	10800	467	10800	0

4 Plan for future research

4.1 Modeling framework

This section formally introduces the groundwork on the methodology that is going to be investigated in future research.

Choice based optimization problems like MSLE complicate the direct application of decomposition methods due to an inherent non-linearity of the constraints modeling the choice of the alternative with the highest utility. There are multiple ways to tackle non-linearity in MILPs, for example with big-M reformulations, piece-wise linearization or the modeling of the problem as a (mixed-integer) convex quadratic program. An additional, original, linearization technique has been developed by the authors, however, it is more appropriate to characterize it as a "quasi"-linearization, as the product of two variables is defined with the use of fixed constants, depending on the variable value. It only makes sense to use the quasi linearization technique in the context of knapsack Benders (and extensions thereof). We furthermore give a brief summary on column generation, as it as promising alternative to Benders decomposition. Table 7 below shows the compatibility of different decomposition methods with various linearization techniques.

Table 7: Compatibility between decomposition methods and linearization methods

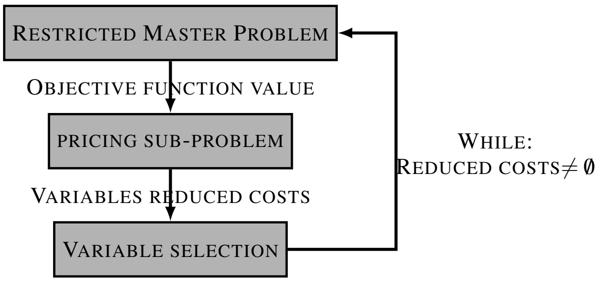
	Big-M	Quasi	Piece-wise	MICQP
Benders (knapsack)	×	✓	✓	?
Benders (standard)	✓	×	✓	✓
Column generation	✓	×	✓	✓
Benders + Col. gen.	✓	✓	✓	✓

It is important to note that there are possible issues arising from the piece-wise approximation of products involving relaxed integral variables in the context of total unimodularity. Furthermore it is difficult to predict the compatibility of knapsack Benders and MICQP, as no previous theory exists to rely on. What follows are short descriptions of the methods that are to be explored.

4.2 Decomposition methods

Column generation Combinatorial optimization problems featuring complicating constraints can be addressed by a *column generation* approach (Desaulniers *et al.*, 2006). Such an approach iteratively adds the decision variables involved in the complicating constraints, employing duality theory to determine whether there are more decision variables that can be added (i.e. enter the basis) and provide an improved solution (i.e. they have negative reduced costs). This is achieved by solving a pricing sub-problem, in which the objective function is the reduced cost of the new candidate variable subject to problem-specific constraints. After solving the pricing sub-problems, if there are multiple variables that can potentially enter the basis, the variable exhibiting the most negative reduced cost can be added. Instead, if there are no variables that can be added to the basis, an optimal solution has been found and the algorithm can be terminated at an early stage. A flowchart describing the several steps for column generation is shown in Figure 6. Column generation is especially interesting for choice-based problems, as many of the complicating choice variables are set to zero, while only the variable corresponding to the choice with the highest utility is set to be equal to tone. This means all other choice variables will remain non-basic and can technically be excluded. It is furthermore possible, that the performance of a column generation approach could further be improved by grouping certain variables together, instead of treating them all individually. A similar "smoothing" approach showed to improve the performance of for example Lagrangean decomposition in the same context, see Pacheco *et al.* (2018). The extension of column generation methods to mixed-integer convex quadratic problems is often referred to as Simplicial Decomposition (SD) or the Frank-Wolfe algorithm (see Bettiol, 2019). For a review on column generation, we refer to Lübbecke (2010).

Figure 6: Flowchart for column generation



4.3 Linearization methods

In this subsection we briefly describe the different methods anticipated to be capable of linearizing the nonlinear choice-constraints, i.e.

$$H_{nr} = \sum_i U_{inr}\omega_{inr} \tag{$\star$}$$

where H_{nr} represents the highest utility and ω the choice variables for every alternative i, individual n, and scenario r.

big-M linearization This is the most standard way of dealing with nonlinear constraints as the ones in ($\star$). It gets its name from utilizing a large enough constant M to deactivate constraints if needed. The nonlinear product is modeled with an auxiliary variable $\eta_{inr} = U_{inr}\omega_{inr}$:

$$\begin{aligned}
H_{nr} &= \sum_i \eta_{inr} \\
\eta_{inr} &\leq \omega_{inr}M \\
\eta_{inr} &\geq -\omega_{inr}M \\
\eta_{inr} &\leq U_{inr} + (1-\omega_{inr})M \\
\eta_{inr} &\geq U_{inr} - (1-\omega_{inr})M
\end{aligned}$$

This linearization method only works if ω is kept integral, thus making it incompatible with knapsack Benders.

Quasi linearization This linearization method only works in the context of knapsack Benders, where in the subproblem, the parameter to be optimized is fixed, and thus the utilities are as well. However, in order to derive Benders cuts, the fixing of variables to a given value has to happen in specific way, with the fixed value isolated on the rhs of the constraint. Technically, the fixed value should also not appear in the coefficient matrix, but so far we have not been able to come up with a reformulation that achieves this. The current approach, implemented for the maximum simulated likelihood estimation problem, where the parameter to be estimated is called β_k, is shown below. Note that, instead of modelling the product $U_{inr}\omega_{inr}$ we instead model the k products $\eta_{inrk} = \beta_k\omega_{inr}$, where k is the dimension of the optimization variable.

$$\begin{aligned}
\chi_{inr} + \omega_{inr} &= 1 \\
\eta_{inrk} + \beta_k^{\text{fixed}}\chi_{inr} &= \beta_k^{\text{fixed}} \\
\textstyle\sum_i \eta_{inrk} &= \beta_k
\end{aligned}$$

Piece-wise linearization For this method, we should first write the product $\beta_k\omega_{inr}$ in terms of quadratic expressions: we define two new variables η_{inrk} and ϕ_{inrk} in the following way:

$$\begin{aligned}\eta_{inrk} &= \tfrac{1}{2}(\beta_k + \omega_{inr})\\ \phi_{inrk} &= \tfrac{1}{2}(\beta_k - \omega_{inr})\end{aligned}$$

Now it holds that:

$$\beta_k\omega_{inr} = \eta_{inrk}^2 - \phi_{inrk}^2$$

We then need to come up with good bounds on $\beta_k \in [lb_k, ub_k]$, as for ω these bounds are already given by [0,1]. Then the classical way of piece-wise linearizing would be to first divide $[lb_k, ub_k]$ into m separate segments and define a_j, $(j \in \{0,1,\dots,m\})$ as the breakpoints, $a_0 < a_1 < \cdots < a_m$. Then, we can approximately linearize the non-linear function $f(x) = \bar{x}$ for $x \in [a_0, a_m]$ as follows:

$$\begin{aligned}\textstyle\sum_j \lambda_j a_j &= x\\ \textstyle\sum_j \lambda_j f(a_j) &= \bar{x}\\ \textstyle\sum_j \lambda_j &= 1\\ \lambda &\geq 0\end{aligned}$$

with the restriction that only two adjacent λ_j's are allowed to be nonzero. This is usually modeled using SOS2 constraints, however, this requires the use of $m+1$ binary variables, which, in the context of relaxed choice variables ω, would make little sense. However, in some scenarios these adjacency constraints can be redundant, for example if the function to be linearized is either convex and minimized by the objective, or concave and maximized (see Aimms, 2016). It is thus open to investigation if the approach can be made to work in a relaxation setting. Another approach worth investigating is to employ a method developed by Li and Yu (1999), which uses the fact that, if the slope of f is non-decreasing, which is the case for $f(x) = x^2$, its possible to write a piece-wise linearization using absolute values, without having to introduce auxiliary binary variables to deal with non-convex parts. First, a univariate mathematical function is formulated via a piece-wise linear sum of absolute expressions. Denote s_j, $(j \in \{0,1,\dots,m-1\})$ as the slopes of each line between a_j and a_{j+1} computed using the following equation:

$$s_j = \frac{f(a_{j+1}) - f(a_j)}{a_{j+1} - a_j}, \quad \forall j \in \{0,\dots,m-1\}$$

An equivalent piece-wise linear form of non-linear function $f(x)$ can then be reformulated as follows:

$$L(f(x)) = f(a_0) + s_0(x - a_0) + \sum_{j=1}^{m-1} \frac{s_j - s_{j-1}}{2}(|x - a_j| + x - a_j)$$

where the absolute expressions can be linearized as follows:

$$\begin{aligned}
L(f(x)) &= f(a_0) + s_0(x - a_0) + \sum_{j=1}^{m-1}(s_j - s_{j-1})(\sum_{h=0}^{j-1} d_h + x - a_j) \\
x + \sum_{j=0}^{m-2} d_j &\geq a_{m-1} \\
d_j &\leq a_{j+1} - a_j \quad \forall j \in \{0, \ldots, m-1\} \\
d &\geq 0
\end{aligned}$$

The overall advantage of a piece-wise linear approximation is that the accuracy of the linearization can be controlled using the number of breakpoints a_j, thus a clear trade-off between accuracy and efficiency is given. A clear downside is that it is very unclear whether the beneficial and crucial properties of the knapsack reduction still hold when the products with the integral variables are approximated. Positive in this regard is the fact that the approximation is most crucial at 0, where it can be made equal to the function by adding a breakpoint. Furthermore, there is the need for valid bounds on the parameters β to be estimated, however, if we manage to employ this method successfully in a relaxation context, adding more breakpoints simply means adding more continuous variables to a linear program, which is not expensive. This allows to approximate the square function over a large interval with high accuracy.

MICQP formulation Another way of dealing with nonlinearity in an MILP would be switch the paradigm completely and regard the problem as mixed-integer convex quadratic program. It is yet to be shown that choice-based optimization programs can be written in a convex way, and it might well depend on problems specifics. The intuition to why it might work is the following: since we can always shift the utilities without changing the optimal solution, we can wlog assume all utilities are negative. This means that in the product $U_{inr}\omega_{inr}$, again lets say in a relaxation setting where both variables are continuous, we are multiplying a negative continuous variable with a positive continuous variable, plus the constraint ($\star$) can be wlog written as

$$H_{nr} \leq \sum_i U_{inr}\omega_{inr}$$

since the lower bound is given by the constraints

$$H_{nr} \ \geq \ U_{inr}, \quad \forall i$$

This means we have a variable H_{nr} which is trying to stay below a sum of concave curves, which implies that the feasible region for this constraint is, in theory, convex.

5 Conclusion

In this paper, we develop a mixed integer linear program (MILP) for the simulated maximum likelihood estimation (MLSE) problem and construct a Benders decomposition approach to speed up the solution process. The methodology can be applied to any advanced discrete choice model and makes use of total unimodularity to keep the master problem linear in the decomposition, avoiding the typical bottleneck in efficiency for a Benders decomposition. The results on a binary logit discrete choice model show an average speed up of factor 35, with instances being solved up to 100 times faster. Small deviations in the optimal solution values between decomposition and full MILP are present. This issue will be tackled in future research by investigating different ways of dealing with non-linearity, before moving on to alternative decomposition approaches.

6 References

Aimms, B. (2016) Aimms modeling guide—integer programming tricks, *Pinedo, Michael. Scheduling: theory, algorithms, and systems. Haarlem, The Netherlands: AIMMS BV. Springer. Retrieved from https://download. aimms. com/aimms/download/manuals/AIMMS3OM_IntegerProgrammingTricks. pdf.*

Alenezy, E. J. (2020) Solving capacitated facility location problem using lagrangian decomposition and volume algorithm, *Advances in Operations Research*, **2020**.

Ben-Akiva, M. and M. Bierlaire (2003) Discrete choice models with applications to departure time and route choice, Operations Research and Management Science, 7–38.

Benders, J. F. (1962) Partitioning procedures for solving mixed-variables programming problems, *Numerische mathematik*, **4** (1) 238–252.

Bettiol, E. (2019) Column generation methods for quadratic mixed binary programming, Theses, Université Paris-Nord - Paris XIII.

Bierlaire, M. (1998) Discrete choice models, in *Operations research and decision aid methodologies in traffic and transportation management*, 203–227, Springer.

Bierlaire, M. (2020) A short introduction to pandasbiogeme, *A short introduction to PandasBiogeme*.

Bortolomiol, S., V. Lurkin, M. Bierlaire and C. Bongiovanni (2021) Benders decomposition for choice-based optimization problems with discrete upper-level variables, paper presented at the *21st Swiss Transport Research Conference*, no. CONF.

Boyer, V., B. Gendron and L.-M. Rousseau (2014) A branch-and-price algorithm for the multi-activity multi-task shift scheduling problem, *Journal of Scheduling*, **17** (2) 185–197.

Bront, J. J. M., I. Méndez-Díaz and G. Vulcano (2009) A column generation algorithm for choice-based network revenue management, *Operations Research*, **57** (3) 769–784.

CASE, N. M. C. (2017) Data collection.

Conejo, A. J., E. Castillo, R. Minguez and R. Garcia-Bertrand (2006) *Decomposition techniques in mathematical programming: engineering and science applications*, Springer Science & Business Media.

Cordeau, J.-F., G. Stojković, F. Soumis and J. Desrosiers (2001) Benders decomposition for simultaneous aircraft routing and crew scheduling, *Transportation science*, **35** (4) 375–388.

Desaulniers, G., J. Desrosiers and M. M. Solomon (2002) Accelerating strategies in column generation methods for vehicle routing and crew scheduling problems, in *Essays and surveys in metaheuristics*, 309–324, Springer.

Desaulniers, G., J. Desrosiers and M. M. Solomon (2006) *Column generation*, vol. 5, Springer Science & Business Media.

Dow, J. K. and J. W. Endersby (2004) Multinomial probit and multinomial logit: a comparison of choice models for voting research, *Electoral studies*, **23** (1) 107–122.

Feillet, D. (2010) A tutorial on column generation and branch-and-price for vehicle routing problems, *4or*, **8** (4) 407–424.

Fernández Antolín, A. (2018) Dealing with correlations in discrete choice models, *Technical Report*, EPFL.

Fernandez-Antolin, A., V. Lurkin and M. Bierlaire (2018) Maximum likelihood estimation of discrete and continuous parameters: an milp formulation, paper presented at the *ORBEL 32 Conference*, no. POST_TALK.

Fisher, M. L. (1981) The lagrangian relaxation method for solving integer programming problems, *Management science*, **27** (1) 1–18.

Gendron, B. (2019) Revisiting lagrangian relaxation for network design, *Discrete applied mathematics*, **261**, 203–218.

Goodfellow, I., Y. Bengio and A. Courville (2016) Machine learning basics, *Deep learning*, **1** (7) 98–164.

Gurobi Optimization, LLC (2021) Gurobi Optimizer Reference Manual, `https://www.gurobi.com`.

Hauschild, T. and M. Jentschel (2001) Comparison of maximum likelihood estimation and chi-square statistics applied to counting experiments, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, **457** (1-2) 384–401.

Heidari-Fathian, H. and S. H. R. Pasandideh (2018) Green-blood supply chain network design: Robust optimization, bounded objective function & lagrangian relaxation, *Computers & Industrial Engineering*, **122**, 95–105.

Koch, S., J. Gönsch and C. Steinhardt (2017) Dynamic programming decomposition for choice-based revenue management with flexible products, *Transportation Science*, **51** (4) 1046–1062.

Li, H.-L. and C.-S. Yu (1999) A global optimization method for nonconvex separable programming problems, *European Journal of Operational Research*, **117** (2) 275–292.

Lübbecke, M. E. (2010) Column generation, *Wiley encyclopedia of operations research and management science. Wiley, New York*, 1–14.

Myung, I. J. (2003) Tutorial on maximum likelihood estimation, *Journal of mathematical Psychology*, **47** (1) 90–100.

Pacheco, M. (2020) A general framework for the integration of complex choice models into mixed integer optimization, Ph.D. Thesis, École Polytechnique Fédérale de Lausanne.

Pacheco, M., M. Bierlaire, B. Gendron and S. S. Azadeh (2021) Integrating advanced discrete choice models in mixed integer linear optimization, *Transportation Research Part B: Methodological*, **146**, 26–49.

Pacheco, M., B. Gendron, V. Lurkin and S. S. A. M. Bierlaire (2018) A lagrangian relaxation technique for the demand-based benefit maximization problem, paper presented at the *Proceedings of the 18th Swiss Transport Research Conference (Ascona, Switzerland)*.

Papadakos, N. (2009) Integrated airline scheduling, *Computers & Operations Research*, **36** (1) 176–195.

Rahmaniani, R., S. Ahmed, T. G. Crainic, M. Gendreau and W. Rei (2020) The benders dual decomposition method, *Operations Research*, **68** (3) 878–895.

Rahmaniani, R., T. G. Crainic, M. Gendreau and W. Rei (2017) The benders decomposition algorithm: A literature review, *European Journal of Operational Research*, **259** (3) 801–817.

Shan, W., Z. Peng, J. Liu, B. Yao and B. Yu (2020) An exact algorithm for inland container transportation network design, *Transportation Research Part B: Methodological*, **135**, 41–82.

Sur, P. and E. J. Candès (2019) A modern maximum-likelihood theory for high-dimensional logistic regression, *Proceedings of the National Academy of Sciences*, **116** (29) 14516–14525.

Tiwari, R., S. Jayaswal and A. Sinha (2021) Alternate solution approaches for competitive hub location problems, *European Journal of Operational Research*, **290** (1) 68–80.

Train, K. E. (2009) *Discrete choice methods with simulation*, Cambridge university press.

Van Den Eeckhout, M., M. Vanhoucke and B. Maenhout (2021) A column generation-based diving heuristic to solve the multi-project personnel staffing problem with calendar constraints and resource sharing, *Computers & Operations Research*, **128**, 105163.

Yan, C., C. Barnhart and V. Vaze (2020) Choice-based airline schedule design and fleet assignment: A decomposition approach, *Available at SSRN 3513164*.

Yu, G., W. B. Haskell and Y. Liu (2017) Resilient facility location against the risk of disruptions, *Transportation research part B: methodological*, **104**, 82–105.