Electronic companion for "Exact first-choice product line optimization" by D. Bertsimas and V. V. Mišić

EC.1. Proofs

EC.1.1. Proof of Proposition 1

To prove that $\mathcal{F}_{Utility} \subseteq \mathcal{F}_{MZ}$, let $(\mathbf{x}, \mathbf{y})$ be an optimal solution to the relaxation of problem (4). Observe that constraints (5b), (5c) and (5g) are automatically satisfied by $(\mathbf{x}, \mathbf{y})$ as these constraints are also found in problem (4). Therefore, we only need to verify constraints (5d), (5e) and (5f).

To verify constraint (5d), let us start from constraint (4d) where the right-hand side product is j:

$$\sum_{j'=0}^{n} u^k_{j'} y^k_{j'} \geq (u^k_j - L^k) x_j + L^k.$$

Since the utilities are non-negative and $x_j \leq 1$, we have that

$$(u^k_j - L^k) x_j + L^k = u^k_j x_j + L^k (1 - x_j) \geq u^k_j x_j,$$

which allows us to assert that

$$\sum_{j'=0}^{n} u^k_{j'} y^k_{j'} \geq u^k_j x_j. \tag{EC.1}$$

The left-hand side can be upper-bounded as

$$u^k_i x_i + \sum_{\substack{j'=0 \\ j' \neq i}}^{n} U^k y^k_{j'} \geq \sum_{j'=0}^{n} u^k_{j'} y^k_{j'},$$

since $y^k_j \leq x_j$ for all j, $U^k \geq u^k_j$ for all j, and all y^k_j are nonnegative. Combining this with (EC.1), we get that

$$u^k_i x_i + \sum_{\substack{j'=0 \\ j' \neq i}}^{n} U^k y^k_{j'} \geq u^k_j x_j.$$

We can re-arrange this to obtain

$$\begin{aligned} u^k_i x_i &\geq u^k_j x_j - \sum_{\substack{j'=0 \\ j' \neq i}}^{n} U^k y^k_{j'} \\ &= u^k_j x_j - U^k \cdot (1 - y^k_i), \end{aligned}$$

where the second step follows by the fact that $\sum_{j'=0}^{n} y^k_{j'} = 1$. This establishes that $(\mathbf{x}, \mathbf{y})$ satisfies the constraint

$$u^k_i x_i \geq u^k_j x_j - U^k \cdot (1 - y^k_i),$$

as required. Similar reasoning can be used to establish that constraints (5e) and (5f) also hold. □

EC.1.2. Proof of Theorem 1

EC.1.2.1. Proof of Part (a) Let $(\mathbf{x},\mathbf{y})$ be an optimal solution to the relaxation of problem (2). We need to establish that $(\mathbf{x},\mathbf{y})$ is feasible for problem (3). We begin by observing that constraints (3b), (3c) and (3f) are automatically satisfied, since these constraints also exist in problem (2). Therefore, we only need to focus on constraints (3d) and (3e).

To establish constraint (3d), observe that $(\mathbf{x},\mathbf{y})$ satisfies constraint (2d), so for each $i \in \{1,\dots,n\}$, we have

$$\sum_{j:\sigma^k(j)>\sigma^k(i)} y_j^k \leq 1 - x_i.$$

Observe that for any j with $\sigma^k(j) > \sigma^k(i)$, we have that

$$y_j^k \leq \sum_{j':\sigma^k(j')>\sigma^k(i)} y_{j'}^k$$

which, combined with constraint (2d), lets us assert that

$$y_j^k \leq 1 - x_i,$$

for every $i \in \{1,\dots,n\}$ and every j with $\sigma^k(j) > \sigma^k(i)$. This establishes that $(\mathbf{x},\mathbf{y})$ satisfies constraint (3d); similar reasoning allows us to establish that constraint (3e) is also satisfied. □

EC.1.2.2. Proof of Part (b) Let $(\mathbf{x},\mathbf{y})$ be an optimal solution to the relaxation of problem (2). We need to establish that $(\mathbf{x},\mathbf{y})$ is feasible for problem (4). We begin by observing that constraints (4b), (4c) and (4f) are automatically satisfied, since these constraints also exist in problem (2). Therefore, we only need to focus on constraints (4d) and (4e).

Before we begin, let us establish a useful identity for $(\mathbf{x},\mathbf{y})$. Constraint (2d) is

$$\sum_{j:\sigma^k(j)>\sigma^k(i)} y_j^k \leq 1 - x_i,$$

which can be re-arranged to obtain

$$x_i \leq 1 - \sum_{j:\sigma^k(j)>\sigma^k(i)} y_j^k.$$

Combining this last inequality with constraint (2b), we obtain

$$x_i \leq \sum_{j:\sigma^k(j)\leq\sigma^k(i)} y_j^k. \tag{EC.2}$$

Now, to show that constraint (4d) holds, we have

$$\begin{aligned}
(u_i^k - L^k)x_i + L^k &\leq (u_i^k - L^k) \sum_{j:\sigma^k(j)\leq\sigma^k(i)} y_j^k + L^k \\
&= (u_i^k - L^k) \sum_{j:\sigma^k(j)\leq\sigma^k(i)} y_j^k + L^k \sum_{j=0}^{n} y_j^k \\
&= (u_i^k - L^k) \sum_{j:\sigma^k(j)\leq\sigma^k(i)} y_j^k + L^k \sum_{j:\sigma^k(j)\leq\sigma^k(i)} y_j^k + L^k \sum_{j:\sigma^k(j)>\sigma^k(i)} y_j^k \\
&= \sum_{j:\sigma^k(j)\leq\sigma^k(i)} u_i^k y_j^k + \sum_{j:\sigma^k(j)>\sigma^k(i)} L^k y_j^k
\end{aligned}$$

$$
\begin{aligned}
&\leq \sum_{j:\sigma^k(j)\leq\sigma^k(i)} u_j^k y_j^k + \sum_{j:\sigma^k(j)>\sigma^k(i)} u_j^k y_j^k \\
&= \sum_{j=0}^{n} u_j^k y_j^k
\end{aligned}
$$

where the first step follows by identity (EC.2) and the fact that $(u_i^k - L^k)$ is nonnegative (recall the definition of L^k as $L^k = \min_{0\leq i\leq n} u_i^k$); the second step follows by constraint (2b); the third step follows by splitting the second sum from the second step; the fourth step follows by putting together the first two sums in the third step, and then moving the coefficient in front of each sum to inside its respective sum; the fifth step follows by the fact that each y_j^k is nonnegative, each u_j^k is nonnegative, that

$$u_j^k \geq u_i^k$$

whenever $\sigma^k(j) \leq \sigma^k(i)$, and that $L^k \leq u_j^k$ for all j; and the final step by simple algebra. Similar reasoning can be used to establish that constraint (4e) holds. □

EC.1.2.3. Proof of Part (c) Let $(\mathbf{x}, \mathbf{y})$ be an optimal solution to the relaxation of problem (2). We need to establish that $(\mathbf{x}, \mathbf{y})$ is feasible for problem (5). As in parts (a) and (b), constraints (5b), (5c) and (5g) are automatically satisfied as these constraints are also found in problem (2). We thus only need to establish that constraints (5d), (5e) and (5f) hold.

We begin by considering constraint (5d). We proceed in two cases.

Case 1: $u_j^k < u_i^k$. In this case, we have

$$
\begin{aligned}
u_j^k x_j - U^k(1-y_i^k) &\leq u_j^k - U^k(1-y_i^k) \\
&\leq u_j^k - u_i^k(1-x_i) \\
&= u_j^k - u_i^k + u_i^k x_i \\
&\leq u_i^k x_i,
\end{aligned}
$$

where the first step follows by the fact that u_j^k is nonnegative and x_j is upper bounded by 1; the second step follows by the fact that $U^k \geq u_i^k$ for all i and the fact that $x_i \leq y_i^k$, which can be re-arranged to assert that $(1-y_i^k) \geq (1-x_i)$; the third step by algebra; and the fourth by the assumption of this case.

Case 2: $u_j^k > u_i^k$. Recall that by constraint (2d), we have

$$y_j^k \leq \sum_{j':\sigma^k(j')>\sigma^k(i)} y_i^k \leq 1 - x_i$$

for any $\sigma^k(j) > \sigma^k(i)$; re-arranging, we get that whenever $\sigma^k(j) > \sigma^k(i)$, we have

$$x_i \leq 1 - y_j^k. \tag{EC.3}$$

Now, observe that, since $u_j^k > u_i^k$, it must be that $\sigma^k(j) < \sigma^k(i)$. We therefore have

$$
\begin{aligned}
u_j^k x_j - U^k(1-y_i^k) &\leq u_j^k x_j - U^k x_j \\
&\leq 0 \\
&\leq u_i^k x_i,
\end{aligned}
$$

where the first step follows by inequality (EC.3); the second step follows by the fact that $U^k \geq u_j^k$ for all j; and the final step by the fact that both u_i^k and x_i are nonnegative.

This establishes constraint (5d). Similar reasoning can be used to establish constraints (5e) and (5f). □

EC.1.3. Proof of Proposition 2

The proof of Proposition 2 follows almost immediately from the proof of Part (c) of Theorem 1. As in part (c) of Theorem 1, the only constraints that need to be established are (5d), (5e) and (5f); just as in part (c) of Theorem 1, we will only focus on (5d).

Constraint (5d) can be established in the same way: case 1 follows through without any modifications, and for case 2, the key inequality that is needed is

$$y_j^k \leq 1 - x_i$$

for any j such that $\sigma^k(j) > \sigma^k(i)$. For a feasible solution of our problem (2), this was established as an implication of constraint (2d). For a feasible solution of problem (3), it is even more straightforward because this inequality is actually a constraint of problem (3) (specifically constraint (3d)) and so is automatically available to us.

Thus, the same steps used in the proof of part (c) of Theorem 1 are applicable here, which establishes the result. $\square$

EC.1.4. Proof of Proposition 3

We first prove that $\mathcal{F}_{BFSS} \nsubseteq \mathcal{F}_{Utility}$. We will prove this through a counterexample. Consider an instance with $n = 5$ and $K = 1$, and the ranking σ defined by $\sigma(i) = i - 1$ for $i \in \{1, \ldots, 5\}$ and $\sigma(0) = 5$ (we drop the k superscript for convenience). Set $\mathbf{C} = [\mathbf{0}^T], \mathbf{d} = [0]$ (i.e., the constraint $\mathbf{Cx} \leq \mathbf{d}$ is vacuous). Let the utility function be given by $u_i = 6 - i$ for $i \in \{1, \ldots, 5\}$ and $u_0 = 0$, so that $L = 0$.

We can see that the following is a feasible solution of $\mathcal{F}_{BFSS}$:

$$\begin{aligned} x_1 &= 0.5, \quad & y_1 &= 0, \\ x_2 &= 0.5, \quad & y_2 &= 0, \\ x_3 &= 0.5, \quad & y_3 &= 0, \\ x_4 &= 0.5, \quad & y_4 &= 0.5, \\ x_5 &= 0.5, \quad & y_5 &= 0.5, \\ & & y_0 &= 0. \end{aligned}$$

However, this solution does not belong to $\mathcal{F}_{Utility}$. To see this, observe that $\sum_{j=0}^{n} u_j \cdot y_j = 1.5$. However, for $i = 1$, we have

$$\begin{aligned} (u_i - L)x_i + L &= (5 - 0) \times 0.5 + 0 \\ &= 2.5 \\ &\nleq \sum_{j=0}^{n} u_j y_j = 1.5, \end{aligned}$$

i.e., constraint (4d) of problem (4) is violated. Thus, the candidate solution $(\mathbf{x}, \mathbf{y})$ is not in $\mathcal{F}_{Utility}$. This shows that in general, $\mathcal{F}_{BFSS}$ is not contained in $\mathcal{F}_{Utility}$.

We now prove that $\mathcal{F}_{Utility} \nsubseteq \mathcal{F}_{BFSS}$. Consider the same instance as above. The following solution is a feasible solution of $\mathcal{F}_{Utility}$:

$$\begin{aligned} x_1 &= 0.8, \quad & y_1 &= 0.5, \\ x_2 &= 0.5, \quad & y_2 &= 0.5, \\ x_3 &= 0, \quad & y_3 &= 0, \\ x_4 &= 0, \quad & y_4 &= 0, \\ x_5 &= 0, \quad & y_5 &= 0, \\ & & y_0 &= 0. \end{aligned}$$

The solution is not, however, a feasible solution of $\mathcal{F}_{BFSS}$. To see this, observe that we should have $y_2 \leq 1 - x_1$, by virtue of constraint (3d), but in the above solution, $y_2 = 0.5 \not\leq 1 - x_1 = 1 - 0.8 = 0.2$. Therefore, in general, $\mathcal{F}_{Utility} \not\subseteq \mathcal{F}_{BFSS}$. $\square$

EC.1.5. Proof of Proposition 4

When $K = 1$ and the constraint $\mathbf{Cx} \leq \mathbf{d}$ is removed from the formulation, the feasible region $\mathcal{F}$ of the LO relaxation of problem (2) is the set of $(\mathbf{x}, \mathbf{y})$ that satisfy the following set of constraints:

$$x_i + \sum_{j:\sigma(j)>\sigma(i)} y_j \leq 1, \quad \forall i \in \{1, \ldots, n\}, \tag{EC.4}$$

$$\sum_{j:\sigma(j)>\sigma(0)} y_j \leq 0, \tag{EC.5}$$

$$-x_i + y_i \leq 0, \quad \forall i \in \{1, \ldots, n\}, \tag{EC.6}$$

$$x_i \leq 1, \quad \forall i \in \{1, \ldots, n\}, \tag{EC.7}$$

$$\sum_{j=0}^{n} y_j \leq 1, \tag{EC.8}$$

$$-\sum_{j=0}^{n} y_j \leq -1, \tag{EC.9}$$

$$\mathbf{x} \geq \mathbf{0}, \tag{EC.10}$$

$$\mathbf{y} \geq \mathbf{0}. \tag{EC.11}$$

Note that in the above system, we have re-arranged the inequalities so that all variables are on one side. Note also that constraint (2e) is re-expressed as an inequality and the unit sum constraint (2b) is expressed as two inequalities. In matrix form, the above system can be re-written as

$$\mathbf{A} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \leq \mathbf{b}, \tag{EC.12}$$

$$\mathbf{x}, \mathbf{y} \geq \mathbf{0}. \tag{EC.13}$$

To show that $\mathcal{F}$ is integral, we will first show that the matrix $\mathbf{A}$ is totally unimodular.

To prove that $\mathbf{A}$ is totally unimodular, we will use the following characterization of total unimodularity (see Bertsimas and Weismantel 2005):

Proposition EC.1 **(Corollary 3.2 from Bertsimas and Weismantel 2005)**. *A matrix* $\mathbf{A}$ *is totally unimodular if and only if each collection* Q *of rows of* $\mathbf{A}$ *can be partitioned into two parts so that the sum of the rows in one part minus the sum of the rows in the other is a vector with entries only in* $\{0, +1, -1\}$.

For notational convenience, instead of working with rows, we will work in terms of algebraic expressions involving $\mathbf{x}$ and $\mathbf{y}$. There are four types of expressions which we denoted by A, B, C and D:

$$\text{A}(i),\ i \in \{1, \ldots, n\}: \quad x_i + \sum_{j:\sigma(j)>\sigma(i)} y_j \tag{EC.14}$$

$$\text{A}(0): \quad \sum_{j:\sigma(j)>\sigma(0)} y_j \tag{EC.15}$$

$$\text{B}(i), i \in \{1, \ldots, n\}: \quad -x_i + y_i \tag{EC.16}$$

$$\text{C}(i), i \in \{1, \ldots, n\}: \quad x_i \tag{EC.17}$$

$$\text{D}(1): \qquad \sum_{j=0}^{n} y_j \tag{EC.18}$$

$$\text{D}(2): \qquad -\sum_{j=0}^{n} y_j \tag{EC.19}$$

Instead of working with a collection of rows Q, we will assume that of each type of expression (A, B, C and D) we are given a collection of expressions:

$$\begin{aligned} S_{\text{A}} &\subseteq \{0,1,\ldots,n\}, \\ S_{\text{B}} &\subseteq \{1,\ldots,n\}, \\ S_{\text{C}} &\subseteq \{1,\ldots,n\}, \\ S_{\text{D}} &\subseteq \{1,2\}. \end{aligned}$$

To establish the equivalent condition in Proposition EC.1, we will show that given $S_{\text{A}}, S_{\text{B}}, S_{\text{C}}, S_{\text{D}}$, we can partition the expressions into two groups R_+, R_- such that the difference of the sums of the expressions in each group will yield an expression

$$\sum_{e \in R_+} e - \sum_{e \in R_-} e = \sum_{i=1}^{n} q_i x_i + \sum_{j=0}^{n} w_j y_j,$$

where each q_i and each w_j is in $\{0, +1, -1\}$. This will establish that $\mathbf{A}$ is totally unimodular.

We provide a constructive procedure for generating a valid partition. This procedure proceeds in four steps.

Step 1. Sort the indices $i \in S_{\text{A}}$ according to σ. Specifically, obtain the ordering $i_1, i_2, \ldots, i_{|S_{\text{A}}|}$, such that

$$\sigma(i_1) < \sigma(i_2) < \cdots < \sigma(i_{|S_{\text{A}}|}).$$

Now, for each $i_j \in S_{\text{A}}$:

- If j is odd, put expression $\text{A}(i_j)$ in R_+. If additionally $i_j \in S_{\text{B}}$, put $\text{B}(i_j)$ in R_+.
- If j is even, put expression $\text{A}(i_j)$ in R_-. If additionally $i_j \in S_{\text{B}}$, put $\text{B}(i_j)$ in R_-.

If we evaluate $\sum_{e \in R_+} e - \sum_{e \in R_-} e$, we will obtain an expression of the following form:

$$\sum_{i \in I_+} (+1) x_i + \sum_{i \in I_-} (-1) x_i + \sum_{j \in J} (+1) y_j,$$

where $J \subseteq \{0,1,\ldots,n\}$, and where $I_+, I_- \subseteq \{1,\ldots,n\}$. Note that by the above procedure, the resulting coefficient of each x_i is either 0 (if the corresponding i was in both S_{A} and S_{B}, or not in S_{A}), +1 (if the corresponding $i = i_j$ was in S_{A} and not in S_{B}, and j was odd) or -1 (if the corresponding $i = i_j$ was in S_{A} and not in S_{B}, and j was even). Note that I_+ and I_- do not intersect – each $i \in I_+$ corresponds to an i_j for an odd j, and each $i \in I_-$ corresponds to an i_j for an even j.

Lastly, note that the coefficients on the y_j's so far are either 0 or +1. This is assured because we have sorted the i's from most to least preferred in terms of σ, and so the y_j's that participate in $\text{A}(i_1), \text{A}(i_2), \ldots, \text{A}(i_{|S_{\text{A}}|})$ form a nested sequence. In particular, for odd $|S_{\text{A}}|$, we have

$$\begin{aligned} &\text{A}(i_1) - \text{A}(i_2) + \text{A}(i_3) - \text{A}(i_4) + \cdots + \text{A}(|S_{\text{A}}|) \\ &= x_{i_1} - x_{i_2} + x_{i_3} - x_{i_4} + \cdots + x_{i_{\text{A}}} + \\ &\quad + \sum_{j:\sigma(j) > \sigma(i_1)} y_j - \sum_{j:\sigma(j) > \sigma(i_2)} y_j + \sum_{j:\sigma(j) > \sigma(i_3)} y_j - \sum_{j:\sigma(j) > \sigma(i_4)} + \cdots + \sum_{j:\sigma(j) > \sigma(i_{|S_{\text{A}}|})} y_j \\ &= x_{i_1} - x_{i_2} + x_{i_3} - x_{i_4} + \cdots + x_{i_{\text{A}}} \\ &\quad + \sum_{j:\sigma(i_2) \geq \sigma(j) > \sigma(i_1)} y_j + \sum_{j:\sigma(i_4) \geq \sigma(j) > \sigma(i_3)} y_j + \cdots + \sum_{j:\sigma(j) > \sigma(i_{|S_{\text{A}}|})} y_j. \end{aligned}$$

and for even $|S_{\mathrm{A}}|$ we end up with (intermediate steps omitted)

$$\begin{aligned}
&\mathrm{A}(i_1) - \mathrm{A}(i_2) + \mathrm{A}(i_3) - \mathrm{A}(i_4) + \cdots + \mathrm{A}(|S_{\mathrm{A}}|-1) - \mathrm{A}(|S_{\mathrm{A}}|) \\
&= x_{i_1} - x_{i_2} + x_{i_3} - x_{i_4} + \cdots + x_{i_{|S_{\mathrm{A}}|-1}} - x_{i_{|S_{\mathrm{A}}|}} \\
&\quad + \sum_{j:\sigma(i_2)\geq\sigma(j)>\sigma(i_1)} y_j + \sum_{j:\sigma(i_4)\geq\sigma(j)>\sigma(i_3)} y_j + \cdots + \sum_{j:\sigma(i_{|S_{\mathrm{A}}|})\geq\sigma(j)>\sigma(i_{|S_{\mathrm{A}}|-1})} y_j .
\end{aligned}$$

By the definition of $i_1, i_2, \ldots, i_{|S_{\mathrm{A}}|}$, all of the intervals of the form $\{i' : \sigma(i_{j+1}) \geq \sigma(i') > \sigma(i_j)\}$ are disjoint and do not intersect, so all y_j's have coefficients of 0 or +1. Note that once we add the $\mathrm{B}(i)$ expressions for those $i \in S_{\mathrm{A}} \cap S_{\mathrm{B}}$ as described above (we add $\mathrm{B}(i_j)$ to R_+ if j is odd, and to R_- if j is even), this will only change whether the inequalities in each interval $\{i' : \sigma(i_{j+1}) \geq \sigma(i') > \sigma(i_j)\}$ are strict or non-strict. In the end, after we perform Step 1, all y_j's have a coefficient of 0 or +1.

Step 2. For $i \in S_{\mathrm{B}} \setminus S_{\mathrm{A}}$:

- If $i \in J$ (coefficient of y_j after Step 1 is +1), then add $\mathrm{B}(i)$ to R_-; otherwise,
- If $i \notin J$ (coefficient of y_j after Step 1 is 0), then add $\mathrm{B}(i)$ to R_+.

Observe that by definition of this step, each y_j still has a coefficient of either 0 or +1. Observe also that by taking this step, the coefficients of the x_i's remain in $\{0, +1, -1\}$. This is because the x_i's whose coefficients change in this step are disjoint from the x_i's whose coefficients changed in Step 1 – more specifically, $I_+, I_- \subseteq S_{\mathrm{A}}$, while the i's which are being set here are for those i's in $S_{\mathrm{B}} \setminus S_{\mathrm{A}}$.

If we evaluate $\sum_{e\in R_+} e - \sum_{e\in R_-} e$ after this step, we will thus obtain an expression of the following form:

$$\begin{aligned}
&\sum_{i\in I_+} (+1)x_i + \sum_{i\in I_-} (-1)x_i + \sum_{j\in J\setminus(S_{\mathrm{B}}\setminus S_{\mathrm{A}})} (+1)y_j + \sum_{j\in(S_{\mathrm{B}}\setminus S_{\mathrm{A}})\setminus J} (+1)y_j \\
&+ \sum_{i\in(S_{\mathrm{B}}\setminus S_{\mathrm{A}})\cap J} (+1)x_i + \sum_{i\in(S_{\mathrm{B}}\setminus S_{\mathrm{A}})\cap J^C} (-1)x_i .
\end{aligned}$$

Step 3. For $i \in S_{\mathrm{C}}$:

- If $i \in I_+$, add $\mathrm{C}(i)$ to R_-.
- If $i \in I_-$, add $\mathrm{C}(i)$ to R_+.
- If $i \in (S_{\mathrm{B}} \setminus S_{\mathrm{A}}) \cap J$, add $\mathrm{C}(i)$ to R_-.
- If $i \in (S_{\mathrm{B}} \setminus S_{\mathrm{A}}) \cap J^C$, add $\mathrm{C}(i)$ to R_+.
- If i is not in any of the above sets, add $\mathrm{C}(i)$ to R_+.

In this step, we are adding the $\mathrm{C}(i)$ expressions in accordance with the current sign of x_i in the expression after Step 2, so as to ensure that every x_i's coefficient remains in $\{0, +1, -1\}$. After this step, if we evaluate $\sum_{e\in R_+} e - \sum_{e\in R_-} e$, we obtain

$$\begin{aligned}
&\sum_{i\in I_+\setminus S_{\mathrm{C}}} (+1)x_i + \sum_{i\in I_-\setminus S_{\mathrm{C}}} (-1)x_i + \sum_{j\in J\setminus(S_{\mathrm{B}}\setminus S_{\mathrm{A}})} (+1)y_j + \sum_{j\in(S_{\mathrm{B}}\setminus S_{\mathrm{A}})\setminus J} (+1)y_j \\
&+ \sum_{i\in[(S_{\mathrm{B}}\setminus S_{\mathrm{A}})\cap J]\setminus S_{\mathrm{C}}} (+1)x_i + \sum_{i\in[(S_{\mathrm{B}}\setminus S_{\mathrm{A}})\cap J^C]\setminus S_{\mathrm{C}}} (-1)x_i + \sum_{i\in S_{\mathrm{C}}\setminus[I_+\cup I_-\cup(S_{\mathrm{B}}\setminus S_{\mathrm{A}})]} (+1)x_i . \qquad \text{(EC.20)}
\end{aligned}$$

Step 4. Finally, we are left with assigning the expressions in S_{D}. This step has four possible cases:

- If S_{D} is empty, then we are left with expression (EC.20).
- If $S_{\mathrm{D}} = \{1, 2\}$, then assign both $\mathrm{D}(1)$ and $\mathrm{D}(2)$ to R_+. Because $\mathrm{D}(1)$ and $\mathrm{D}(2)$ involve the same variables and have opposite sign, they cancel out, and again we are left with expression (EC.20).

• If $S_{\mathrm{D}} = \{1\}$, then assign D(1) to R_-. Intuitively, this assignment is safe because all y_j's up to this point have a coefficient of +1 or 0 and we are only subtracting 1 from the coefficient of every y_j. As a result, our expression becomes

$$\sum_{i \in I_+ \setminus S_{\mathrm{C}}} (+1)x_i + \sum_{i \in I_- \setminus S_{\mathrm{C}}} (-1)x_i + \sum_{j \in (S_{\mathrm{B}} \setminus S_{\mathrm{A}}) \cap J} (-1)y_j + \sum_{j \in [(S_{\mathrm{B}} \setminus S_{\mathrm{A}}) \cup J]^C} (-1)y_j$$
$$+ \sum_{i \in [(S_{\mathrm{B}} \setminus S_{\mathrm{A}}) \cap J] \setminus S_{\mathrm{C}}} (+1)x_i + \sum_{i \in [(S_{\mathrm{B}} \setminus S_{\mathrm{A}}) \cap J^C] \setminus S_{\mathrm{C}}} (-1)x_i + \sum_{i \in S_{\mathrm{C}} \setminus [I_+ \cup I_- \cup (S_{\mathrm{B}} \setminus S_{\mathrm{A}})]} (+1)x_i. \quad \text{(EC.21)}$$

• If $S_{\mathrm{D}} = \{2\}$, then assign D(2) to R_+. Again, this assignment is safe, because all y_j's have a coefficient of +1 or 0, and we are only adding -1 to every y_j's coefficient. We again end up with expression (EC.21).

After completing Step 4, we are left with expression (EC.20) or (EC.21). In both of these expressions, all variables have coefficients in $\{0, +1, -1\}$; moreover, our procedure assigns all expressions to either R_+ and R_- and leaves no expression unassigned.

As a result, invoking Proposition EC.1, we have that the matrix $\mathbf{A}$ is totally unimodular. We now use the following classical result in integer optimization:

Proposition EC.2 **(Theorem 3.1(b) of Bertsimas and Weismantel 2005)**. *Let $\mathbf{A} \in \mathbb{Z}^{m \times n}$ be an integer matrix. The matrix $\mathbf{A}$ is totally unimodular if and only the polyhedron $P(\mathbf{b}) = \{\mathbf{x} \in \mathbb{R}^n_+ \mid \mathbf{A}\mathbf{x} \leq \mathbf{b}\}$ is integral for all $\mathbf{b} \in \mathbb{Z}^m$ for which $P(\mathbf{b}) \neq \emptyset$.*

In our context, the vector $\mathbf{b}$ defining the system of inequalities (EC.12) is indeed integer. Also, the feasible region $\mathcal{F}$ is nonempty; this will be assured by a later result, Proposition 6, which asserts that for any $\mathbf{x} \in \{0,1\}^n$ and any given ranking σ^k, there exists $\mathbf{y}^k \in \mathbb{R}^{n+1}$ with $\mathbf{y}^k \geq \mathbf{0}$ such that $(\mathbf{x}, \mathbf{y}^k)$ satisfy constraints (2b), (2c), (2d) and (2e). Thus, we can apply Proposition EC.2 to assert that $\mathcal{F}$ is integral. This concludes the proof. □

EC.1.6. Proof of Proposition 5

We consider each model separately. In each case, we assume that the constraint $\mathbf{C}\mathbf{x} \leq \mathbf{d}$ is removed (or equivalently, we set $\mathbf{C} = [\mathbf{0}^T], \mathbf{d} = [0]$, so as to make the constraint $\mathbf{C}\mathbf{x} \leq \mathbf{d}$ vacuous).

Problem (3) **(Belloni et al. 2008)**: For this formulation, consider $n = 5$ and σ defined as

$$\sigma(2) = 0,$$
$$\sigma(1) = 1,$$
$$\sigma(0) = 2,$$
$$\sigma(3) = 3,$$
$$\sigma(5) = 4,$$
$$\sigma(4) = 5.$$

It can be shown that $\mathbf{x} = (1/2, 1/2, 0, 0, 0)$, $\mathbf{y} = (1/2, 1/2, 0, 0, 0, 0)$ (where $\mathbf{y}$ is indexed from 0 to n) is an extreme point of $\mathcal{F}_{BFSS}$. Since this solution is fractional, $\mathcal{F}_{BFSS}$ cannot be integral in general.

Problem (4): For this formulation, consider $n = 5$ and σ defined as

$$\sigma(3) = 0,$$
$$\sigma(5) = 1,$$
$$\sigma(0) = 2,$$
$$\sigma(2) = 3,$$
$$\sigma(1) = 4,$$
$$\sigma(4) = 5.$$

For this ranking, consider the following utilities for the products:

$$u_0 = 3,$$
$$u_1 = 1,$$
$$u_2 = 2,$$
$$u_3 = 5,$$
$$u_4 = 0,$$
$$u_5 = 4.$$

It can be shown that $\mathbf{x} = (1, 1, 1/3, 0, 3/4)$, $\mathbf{y} = (0, 0, 2/3, 1/3, 0, 0)$ is a fractional extreme point of $\mathcal{F}_{Utility1}$.

Problem (5) (McBride and Zufryden 1988): For this formulation, since Proposition 1 shows that $\mathcal{F}_{MZ} \supseteq \mathcal{F}_{Utility1}$, the same choice of σ and utilities $u_0, \ldots, u_n$ as for problem (4) can be used to establish that $\mathcal{F}_{MZ}$ is not integral in general.

EC.1.7. Proof of Proposition 6

We will prove the result by showing that, in fact, the proposed $\mathbf{y}^k$ is the *only* solution of problem (7). First, letting $i^* = \arg\min_{j \in S \cup \{0\}} \sigma^k(j)$, observe that by constraints (7d) and (7e), we have that for $j \in \{0, 1, \ldots, n\}$ with $\sigma^k(j) > \sigma^k(i^*)$, it must be that $y^k_j = 0$. Observe also that for $j \in \{1, \ldots, n\}$ with $\sigma^k(j) < \sigma^k(i^*)$, it must be that $y^k_j = 0$, because it must be that $x_j = 0$ (if x_j were 1, then i^* could not be the index that minimizes $\sigma^k(j')$ for $j' \in S \cup \{0\}$).

These two observations imply that $y^k_i = 0$ for $i \in \{0, 1, \ldots, n\} \setminus \{i^*\}$, and thus by constraint (7b), it must be that $y^k_{i^*} = 1$. Since this completely specifies $\mathbf{y}^k$, it must be the only solution to the problem and hence the optimal solution of the problem. □

EC.1.8. Proof of Proposition 7

We will proceed in two steps: first, we will show that the proposed solution is feasible, and second, we will show that the objective value of the proposed solution is exactly π^*, which is the optimal objective of the primal problem.

Feasibility: Observe that both $\boldsymbol{\alpha}^k$ and $\boldsymbol{\beta}^k$ are nonnegative by their definition. Constraint (9c) is automatically satisfied because by their definition, γ^k and $\boldsymbol{\beta}^k$ are both nonnegative. Thus, we need only verify constraint (9b).

For $j \in S \setminus \{i^*\}$, we have

$$\begin{aligned}
\gamma^k + \sum_{i : \sigma^k(i) < \sigma^k(j)} \beta^k_i &= \gamma^k + \beta^k_{i^*} \\
&= \pi^* + \max_{i' \in S} \pi_{i'} - \pi^* \\
&= \max_{i' \in S} \pi_{i'} \\
&\geq \pi_j
\end{aligned}$$

where the first equality follows by definition of $\boldsymbol{\beta}^k$, the second by definition of $\boldsymbol{\beta}^k_{i^*}$ and γ^k, the third by simple algebra and the final inequality by the definition of the maximum. Since $\boldsymbol{\alpha}^k$ is nonnegative, it follows that

$$\gamma^k + \alpha^k_j + \sum_{i : \sigma^k(i) < \sigma^k(j)} \beta^k_i \geq \pi_j.$$

For $j \in \{1, \dots, n\} \setminus S$, we have

$$\begin{aligned} \gamma^k + \alpha_j^k + \sum_{i : \sigma^k(i) < \sigma^k(j)} \beta_i^k &\geq \gamma^k + \left(\pi_j - \gamma - \sum_{i : \sigma^k(i) < \sigma^k(j)} \beta_i^k \right) + \sum_{i : \sigma^k(i) < \sigma^k(j)} \beta_i^k \\ &= \pi_j \end{aligned}$$

where the inequality follows by the definition of α_j^k via the maximum function. Thus, constraint (9b) holds for $j \in \{1, \dots, n\} \setminus S$.

The above two cases fully verify constraint (9b) when $i^* = 0$. In the case that $i^* \neq 0$, we must check constraint (9b) for i^*. If $i^* \neq 0$, we have for i^* that

$$\begin{aligned} \gamma^k + \sum_{i : \sigma^k(i) < \sigma^k(i^*)} \beta_i^k &= \gamma^k \\ &= \pi^* \\ &= \pi_{i^*}, \end{aligned}$$

where the first equality follows by definition of $\boldsymbol{\beta}$, the second by definition of γ^k and the third by definition of π^*. Since $\boldsymbol{\alpha} \geq \mathbf{0}$ by definition, constraint (9b) must hold for $j = i^*$. This concludes our proof of the feasibility of $(\boldsymbol{\alpha}^k, \boldsymbol{\beta}^k, \gamma)$.

Objective value: For $j \in S \setminus \{i^*\}$, we have that

$$\begin{aligned} \alpha_j^k &= \max \left\{ \pi_j - \gamma^k - \sum_{i : \sigma^k(i) < \sigma^k(j)} \beta_i^k, 0 \right\} \\ &= \max\{ \pi_j - \max_{i' \in S} \pi_{i'}, 0 \} \\ &= 0 \end{aligned}$$

and if $i^* \neq 0$, then for i^* we have that

$$\begin{aligned} \alpha_{i^*}^k &= \max \left\{ \pi_{i^*} - \gamma^k - \sum_{i : \sigma^k(i) < \sigma^k(j)} \beta_i^k, 0 \right\} \\ &= \max\{ \pi_{i^*} - \pi_{i^*} - 0, 0 \} \\ &= 0. \end{aligned}$$

Thus, we have $\alpha_j^k = 0$ for any $j \in S$, or equivalently, any j with $x_j = 1$. Similarly, for $j \in \{1, \dots, n\} \setminus S$ or equivalently, for any $j \in \{1, \dots, n\}$ with $x_j = 0$, we have that $\beta_j^k = 0$. Thus, when we consider the objective value of the solution, we get

$$\begin{aligned} \gamma^k + \sum_{j=1}^n \alpha_j^k x_j + \sum_{i=1}^n \beta_i^k (1 - x_i) &= \pi^* + 0 + 0 \\ &= \pi^* \end{aligned}$$

which is exactly the optimal value of the primal problem. Thus, it follows that $(\boldsymbol{\alpha}^k, \boldsymbol{\beta}^k, \gamma^k)$ is a dual optimal solution. $\square$

EC.1.9. Proof of Theorem 2

We proceed in three stages. First, we show that $\mathbf{y}$ is primal feasible. Second, we show that $(\boldsymbol{\alpha},\boldsymbol{\beta},\gamma)$ is dual feasible. Finally, we show that the two solutions satisfy complementary slackness, which establishes that they are optimal.

Primal feasibility. It is clear from the structure of the algorithm that constraints (13d) and (13c) are never violated in the algorithm; in addition, at each stage of the algorithm, the solution $\mathbf{y}$ satisfies the inequality $\sum_{j=0}^{n} y_j \leq 1$. Note also that since these constraints are never violated, the slacks associated with these inequality constraints are never negative. Therefore, whenever $y_{\tau(s)}$ is set to q^*, it is never set to a negative value; y_j must therefore always be nonnegative, for all j.

We only need to verify that $\sum_{j=0}^{n} y_j = 1$, i.e., the sum of the y_j variables is exactly one upon termination. To see this, we proceed in two cases:

1. **Case 1**: $B_{main} \neq \emptyset$. If B_{main} is not empty, then let $i^{**} = \arg\min_{i \in B_{main}} \sigma(i)$, i.e., it is the option in B_{main} that has the lowest rank. After the algorithm terminates, we know that the preference constraint (13d) for i^{**} is satisfied at equality, that is:

$$\sum_{j:\sigma(j)>\sigma(i^{**})} y_j = 1 - x_{i^{**}}.$$

Given this, we now ask: what happens when the algorithm checks i^{**}? By our assumption on i^{**}, it cannot be that there is a B event when we check i^{**}. Therefore, either a C event happens, in which case we are done because we will set $y_{i^{**}}$ so as to reach the unit sum; or it is neither a C nor a B event, in which case it must be an A event. This latter case cannot happen (if i^{**} is checked before $f(i^{**})$, then when $f(i^{**})$ is checked there should be a C event; if i^{**} is checked after $f(i^{**})$, then there should also be a C event). Therefore, it must be the case $\sum_{j=0}^{n} y_j = 1$ upon termination.

2. **Case 2**: $B_{main} = \emptyset$. If B_{main} is empty, then at each stage of the algorithm there is either an A event or a C event. If there is a C event, we are done. Note that if a C event does not occur by stage $s = n$, then it must occur at stage $s = n$, because $\tau(n) = 0$ is the no-purchase option, $x_{\tau(n)}$ is 1, and $1 - \sum_{j=0}^{n} y_j$ at stage $s = n$ is at most 1.

Dual feasibility. Before we show that $(\boldsymbol{\alpha},\boldsymbol{\beta},\gamma)$ is dual feasible, let us present two useful results regarding the dual phase of the algorithm:

Observation 1. In Algorithm 2, observe that when we sort $B_{main} = \{i_1, i_2, \ldots, i_{|B_{main}|}\}$ in increasing order of σ (i.e., $\sigma(i_1) < \sigma(i_2) < \cdots < \sigma(i_{|B_{main}|})$), then we also have that $\pi_{f(i_1)} \leq \pi_{f(i_2)} \leq \cdots \leq \pi_{f(i_{|B_{main}|})}$. The reason for this is that, by definition of the arg min in the clause for B events in Algorithm 1, we always take the one with the lowest value of σ. Therefore, as we progress through the algorithm, the options i^* that we add to B_{main} are such that their values of σ are decreasing. (They cannot be increasing, because of the way we have defined the arg min.) Since Algorithm 1 checks the options in decreasing order of π_i, it must be that $\pi_{f(i_1)} \leq \pi_{f(i_2)} \leq \cdots \leq \pi_{f(i_{|B_{main}|})}$.

Observation 2. By the definition of Algorithm 2, we have that

$$\gamma + \sum_{t'=1}^{t} \beta_{i_{t'}} = \pi_{f(i_t)}.$$

Having defined the two observations, let us now check dual feasibility. The dual constraints are:

$$\gamma + \alpha_i + \sum_{j:\sigma(j)<\sigma(i)} \beta_j \geq \pi_i, \quad \forall i \in \{0,1,\ldots,n\} \tag{EC.22}$$

$$\alpha_i \geq 0, \quad \forall i \in \{0,1,\ldots,n\}, \tag{EC.23}$$

$$\beta_j \geq 0, \quad \forall j \in \{0,1,\ldots,n\}. \tag{EC.24}$$

Let us begin by checking (EC.24). The set of i's breaks into two cases:

1. **Case 1**: $i \in B_{main}$. If $i \in B_{main}$, then $i = i_t$ for some $t \in \{1, \ldots, |B_{main}|\}$. For $t = 1$, we have

$$\beta_{i_1} = \pi_{f(i_1)} - \gamma,$$

which is nonnegative because $\gamma = \pi_C$, and option $f(i_1)$ must have been checked before option C. For $t > 1$, we have:

$$\begin{aligned}
\beta_{i_t} &= \pi_{f(i_t)} - \gamma - \sum_{t'=1}^{t-2} \beta_{i_{t'}} - \beta_{i_{t-1}} \\
&= \pi_{f(i_t)} - \gamma - \sum_{t'=1}^{t-2} \beta_{i_{t'}} - \left[\pi_{f(i_{t-1})} - \gamma - \sum_{t'=1}^{t-2} \beta_{i_{t'}} \right] \\
&= \pi_{f(i_t)} - \pi_{f(i_{t-1})},
\end{aligned}$$

which is nonnegative because option $f(i_t)$ is checked before option $f(i_{t-1})$, and so its profit must be at least that of $f(i_{t-1})$ (see Observation 1). We therefore have $\beta_i \geq 0$ for $i \in B_{main}$.

2. **Case 2**: $i \notin B_{main}$. By the way that β is initialized, $\beta_i = 0$ for these i's.

Having checked (EC.24), let us check (EC.23). For $i \notin A$, α_i is initialized to zero, so the condition is satisfied. For $i \in A$, we have that

$$\alpha_i = \pi_i - \gamma - \sum_{j:\sigma(j) < \sigma(i)} \beta_j,$$

and by the structure of the β_j's, we know that the last sum can be written as

$$\alpha_i = \pi_i - \gamma - \sum_{t'=1}^{t} \beta_{i_{t'}},$$

for some $t \in \{1, \ldots, |B_{main}|\}$. Note that by Observation 2, we have that

$$\gamma + \sum_{t'=1}^{t} \beta_{i_{t'}} = \pi_{f(i_t)}.$$

Therefore, α_i simplifies to

$$\alpha_i = \pi_i - \pi_{f(i_t)}.$$

So now the question is whether $f(i_t)$ was checked after i or not (i.e., is $\pi_i \geq \pi_{f(i_t)}$). If $f(i_t)$ was checked before i, then we know that the constraint $\sum_{j:\sigma(j)>\sigma(i_t)} y_j \leq 1 - x_{i_t}$ became tight. But recall that it is also the case that $\sigma(i_t) < \sigma(i)$. Therefore, if $f(i_t)$ were checked before i, the aforementioned constraint becoming tight would mean that when checking i, q_2 would have been equal to q^* (a B event would have occurred) and Algorithm 1 would *not* have added i to A. Therefore, it must be that $\pi_i - \pi_{f(i_t)} \geq 0$, so that $\alpha_i \geq 0$.

Having checked (EC.23), let us now check the last dual constraint (EC.22). We can break the set of i's into four mutually exclusive and collectively exhaustive cases.

1. **Case 1**: i's for which $\pi_i \leq \pi_C$. In this case, we have

$$\begin{aligned}
\gamma + \alpha_i + \sum_{j:\sigma(j)<\sigma(i)} \beta_j &\geq \gamma + 0 + 0 \\
&= \pi_C \\
&\geq \pi_i,
\end{aligned}$$

where the first inequality holds by the non-negativity of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ verified above. Thus, the constraint holds for this case.

2. **Case 2**: The set of i's for which a B event occurs ($q^* = q_2$) and $i = f(i_t)$ for some $i_t \in B_{main}$. In this case, we can use Observation 2 to assert that

$$\begin{aligned}\gamma + \alpha_i + \sum_{j:\sigma(j)<\sigma(i)} \beta_j &= \gamma + 0 + \sum_{t'=1}^{t} \beta_{i_{t'}} \\ &= \pi_{f(i_t)},\end{aligned}$$

which clearly satisfies the constraint.

3. **Case 3**: The set of i's for which a B event occurs ($q^* = q_2$), but $i \neq f(i_t)$ for all $i_t \in B_{main}$. Let $i_t \in B_{main}$ be the minimizer in the B event clause in Algorithm 1 when i is checked. Specifically, we have:

$$i_t = \arg \min_{p:\sigma(p)<\sigma(i)} \left\{ 1 - x_p - \sum_{j:\sigma(j)>\sigma(p)} y_j \right\}.$$

Since $i \neq f(i_t)$ for any $i_t \in B_{main}$, this means that i_t was already added to B_{main} in an earlier iteration of the Algorithm 1. Therefore, $\pi_{f(i_t)} \geq \pi_i$. Also, observe that $\sigma(i_1) < \cdots < \sigma(i_t) < \sigma(i)$, by the definition of i_t above. Therefore, we have

$$\begin{aligned}\gamma + \alpha_i + \sum_{j:\sigma(j)<\sigma(i)} \beta_j &\geq \gamma + \sum_{t'=1}^{t} \beta_{i_{t'}} \\ &= \pi_{f(i_t)} \\ &\geq \pi_i,\end{aligned}$$

where the first inequality follows because $\sigma(i_t) < \sigma(i)$; $i_1, \ldots, i_t$ are a subset of the sum condition on the left-hand expression. This establishes the constraint.

4. **Case 4**: $i \in A$. By the definition of α_i, we have

$$\begin{aligned}\gamma + \alpha_i + \sum_{j:\sigma(j)<\sigma(i)} \beta_j &= \gamma + (\pi_i - \gamma - \sum_{j:\sigma(j)<\sigma(i)} \beta_j) + \sum_{j:\sigma(j)<\sigma(i)} \beta_j \\ &= \pi_i,\end{aligned}$$

which verifies the constraint.

Complementary slackness. The complementary slackness conditions are:

$$\gamma \cdot (1 - \sum_{j=0}^{n} y_j) = 0 \tag{EC.25}$$

$$\beta_i \cdot (1 - x_i - \sum_{j:\sigma(j)>\sigma(i)} y_j) = 0 \tag{EC.26}$$

$$\alpha_i \cdot (x_i - y_i) = 0 \tag{EC.27}$$

$$y_i \cdot (\gamma + \alpha_i + \sum_{j:\sigma(j)<\sigma(i)} \beta_j - \pi_i) = 0. \tag{EC.28}$$

Equation (EC.25) is automatically satisfied, because the y solution produced by our algorithm is feasible and satisfies the unit sum condition.

To see that equation (EC.26) holds, observe that for $i \notin B_{main}$ produced by the algorithm, $\beta_i = 0$ and the condition holds. Therefore, we only need to check $i \in B_{main}$. For $i \in B_{main}$, observe that i is only added to B_{main} by the algorithm whenever the inequality $\sum_{j:\sigma(j)>\sigma(i)} y_j \leq 1 - x_i$ becomes tight the very first time. Therefore, for $i \in B_{main}$, the y produced by our algorithm will satisfy $(1 - x_i - \sum_{j:\sigma(j)>\sigma(i)} y_j) = 0$, and the condition holds.

To see that equation (EC.27) holds, observe that for $i \notin A$, α_i by default is set to zero and the condition holds. Therefore, we only need to check $i \in A$. For $i \in A$, observe that i is only added to A when the inequality $y_i \leq x_i$ becomes tight. Therefore, for $i \in A$, the y produced by our algorithm will satisfy $x_i - y_i = 0$, and thus the condition will hold.

Finally, to see that equation (EC.28) holds, we proceed carefully in several steps. First, observe that in Algorithm 1, if a C event occurs, then $C = \tau(s)$, and the loop is terminated. For i with $\pi_i \leq \pi_C$, observe that $y_i = 0$ (since the loop was terminated at option C, and all y_j's for j's with lower profit than C were initialized to zero).

This leaves i's for which $\pi_i > \pi_C$. This remaining set of i's can be partitioned into three mutually exclusive and collectively exhaustive cases, which we now treat.

1. **Case 1**: A B event occurred for i ($q^* = q_2$) and $i = f(i_t)$ for some $i_t \in B_{main}$. This is the case when we first encounter the option i_t as the arg min of expression q_2. In this case, by the way that we have specified the dual solution, we have that

$$\begin{aligned} \gamma + \alpha_i + \sum_{j:\sigma(j)<\sigma(i)} \beta_j - \pi_i &= \gamma + \sum_{t'=1}^{t} \beta_{i_{t'}} - \pi_i \\ &= \pi_i - \pi_i \\ &= 0, \end{aligned}$$

and thus equation (EC.28) must hold.

2. **Case 2**: A B event occurred for i ($q^* = q_2$) and $i \neq f(i_t)$ for all $i_t \in B_{main}$. This is the case when $q^* = q_2$, but the i_t which is the arg min that leads to q_2 has already been encountered. Since i_t has already been encountered, it is the case that $\pi_{f(i_t)} \geq \pi_i$. This is the case because Algorithm 1 scans through the options in decreasing order of profit.

Since i_t was already added to B_{main} by Algorithm 1 when option $f(i^*)$ was tested, the constraint $\sum_{j:\sigma(j)>\sigma(i_t)} y_j \leq 1 - x_{i_t}$ must have become binding after option $f(i^*)$. Therefore y_i must have been set to zero. Since y_i is zero, equation (EC.28) must hold.

3. **Case 3**: $i \in A$. By the definition of α_i, we have that:

$$\begin{aligned} \gamma + \alpha_i + \sum_{j:\sigma(j)<\sigma(i)} \beta_j - \pi_i &= \gamma + \left[\pi_i - \gamma - \sum_{j:\sigma(j)<\sigma(i)} \beta_j \right] + \sum_{j:\sigma(j)<\sigma(i)} \beta_j - \pi_i \\ &= 0, \end{aligned}$$

so that again, equation (EC.28) holds.

Since we have established that the two solutions are feasible for their respective problems and satisfy complementary slackness, this concludes the proof. □

EC.2. Additional theoretical results for Benders decomposition

EC.2.1. Solving the integer Benders problem using classical constraint generation

In this section, we present a constraint generation procedure for solving the Benders problem (10). For the purpose of defining our constraint generation procedure, let us suppose that for each customer type k, we have a set of dual solutions $\bar{A}_k \subseteq A_k$ that have been generated so far. We define the restricted master problem as

$$\underset{\mathbf{x},\mathbf{t}}{\text{maximize}} \quad \sum_{k=1}^{K} \lambda^k t_k \tag{EC.29a}$$

$$\text{subject to} \quad t_k \leq \gamma^k + \sum_{i=1}^{n} \alpha_i^k \cdot x_i + \sum_{i=1}^{n} \beta_i^k \cdot (1 - x_i), \quad \forall\, k \in \{1, \dots, K\},\ (\boldsymbol{\alpha}^k, \boldsymbol{\beta}^k, \gamma^k) \in \bar{A}_k, \tag{EC.29b}$$

$$\mathbf{C}\mathbf{x} \leq \mathbf{d}, \tag{EC.29c}$$

$$x_i \in \{0, 1\}, \quad \forall\, i \in \{1, \dots, n\}. \tag{EC.29d}$$

We define our classical constraint generation procedure as Algorithm 3.

Algorithm 3 Classical constraint generation algorithm for solving integer Benders formulation

Set $\bar{A}_k = \emptyset$ for all customer types $k \in \{1, \dots, K\}$.
Solve the restricted master problem (EC.29) to obtain a solution $(\mathbf{x}, \mathbf{t})$.
For each $k \in \{1, \dots, K\}$:
  Determine primal subproblem solution $\mathbf{y}^k$ using equation (8) and dual subproblem solution $(\boldsymbol{\alpha}^k, \boldsymbol{\beta}^k, \gamma^k)$ using equations (11a) – (11c).
while $\max_{k=1,\dots,K} (t_k - \boldsymbol{\pi}^T \mathbf{y}^k) > 0$ **do**
  For each $k \in \{1, \dots, K\}$ with $\boldsymbol{\pi}^T \mathbf{y}^k < t_k$:
    Set $\bar{A}_k \leftarrow \bar{A}_k \cup \{(\boldsymbol{\alpha}^k, \boldsymbol{\beta}^k, \gamma^k)\}$.
  Solve restricted master problem (EC.29) with $\bar{A}_1, \dots, \bar{A}_K$ to obtain $(\mathbf{x}, \mathbf{t})$.
  For each $k \in \{1, \dots, K\}$:
    Determine primal subproblem solution $\mathbf{y}^k$ using equation (8) and dual subproblem solution $(\boldsymbol{\alpha}^k, \boldsymbol{\beta}^k, \gamma^k)$ using equations (11a) – (11c).
end while
return Optimal solution $(\mathbf{x}, \mathbf{t})$ of problem (10).

A standard result for constraint generation algorithms is that the solution returned by the algorithm is optimal. For completeness, we prove below that the solution produced by Algorithm 3 is in fact optimal.

PROPOSITION EC.3. *Let* $(\mathbf{x}, \mathbf{t})$ *be the solution obtained upon termination of Algorithm 3. Then* $(\mathbf{x}, \mathbf{t})$ *is an optimal solution of problem* (10).

Proof of Proposition EC.3: To prove this, let $(\mathbf{x}^*, \mathbf{t}^*)$ be an optimal solution of problem (10) and let $Z_1 = \sum_{k=1}^{K} \lambda^k t_k^*$ be the optimal objective value of the master solution. Let $Z_2 = \sum_{k=1}^{K} \lambda^k t_k$ be the objective value of the solution generated by Algorithm 3.

Upon termination of Algorithm 3, $(\mathbf{x}, \mathbf{t})$ is the optimal solution of the restricted master problem with constraints (EC.29b) enforced at the sets $\bar{A}_1, \dots, \bar{A}_K$ of dual subproblem solutions that were generated over the execution of Algorithm 3. Since $\bar{A}_k \subseteq A_k$ (the set of generated dual solutions for a given customer type k is a subset of *all* dual feasible solutions for a customer type k), then $(\mathbf{x}^*, \mathbf{t}^*)$ must be a feasible solution for problem (EC.29). Since problems (10) and (EC.29) share the same objective function in terms of $\mathbf{x}$ and $\mathbf{t}$, and since $(\mathbf{x}, \mathbf{t})$ is an optimal solution of (EC.29) it follows that $Z_1 \leq Z_2$.

Now, upon termination of the algorithm, we have that

$$\max_{k=1,\dots,K} (t_k - \boldsymbol{\pi}^T \mathbf{y}^k) \leq 0,$$

which is equivalent to

$$t_k \leq \boldsymbol{\pi}^T \mathbf{y}^k, \quad \forall\, k \in \{1, \dots, K\},$$

where $\mathbf{y}^k$ is the solution obtained from equation (8). By Proposition 6, each $\mathbf{y}^k$ is guaranteed to be an optimal solution of the primal subproblem for customer type k at the current solution $\mathbf{x}$.

Similarly, by Proposition 7, each $(\boldsymbol{\alpha}^k, \boldsymbol{\beta}^k, \gamma^k)$ is guaranteed to be an optimal solution of the dual subproblem for customer type k at $\mathbf{x}$. Therefore, by strong duality, the primal subproblem objective value $\boldsymbol{\pi}^T \mathbf{y}^k$ is equal to the dual subproblem objective value $\gamma^k + \sum_{i=1}^n \alpha_i^k x_i + \sum_{j=0}^n \beta_j^k (1 - x_j)$. We thus have:

$$t_k \leq \gamma^k + \sum_{i=1}^n \alpha_i^k x_i + \sum_{j=1}^n \beta_j^k (1 - x_j), \quad \forall\, k \in \{1, \dots, K\},$$

Since $(\boldsymbol{\alpha}^k, \boldsymbol{\beta}^k, \gamma^k)$ is the optimal dual subproblem solution at $\mathbf{x}$, the above is equivalent to

$$t_k \leq \min \left\{ \bar{\gamma}^k + \sum_{i=1}^n \bar{\alpha}_i^k x_i + \sum_{j=1}^n \bar{\beta}_j (1 - x_j) \;\middle|\; (\bar{\boldsymbol{\alpha}}^k, \bar{\boldsymbol{\beta}}^k, \bar{\gamma}^k) \in A_k \right\}, \quad \forall\, k \in \{1, \dots, K\}.$$

This last inequality is exactly equivalent to

$$t_k \leq \bar{\gamma}^k + \sum_{i=1}^n \bar{\alpha}_i^k x_i + \sum_{j=1}^n \bar{\beta}_j^k (1 - x_j), \quad \forall\, (\bar{\boldsymbol{\alpha}}^k, \bar{\boldsymbol{\beta}}^k, \bar{\gamma}^k) \in A_k,\ k \in \{1, \dots, K\},$$

which is exactly constraint (10b). Thus, $(\mathbf{x}, \mathbf{t})$ must be a feasible solution of the master problem (10). Since problems (10) and (EC.29) share the same objective function, we must have that $Z_2 = \sum_{k=1}^K \lambda^k t_k \leq Z_1$.

Since we have established that $Z_1 \leq Z_2$ and $Z_1 \geq Z_2$, it must be that $Z_1 = Z_2$, and that $(\mathbf{x}, \mathbf{t})$ attains the optimal objective value in problem (10). Since $(\mathbf{x}, \mathbf{t})$ is a feasible solution of problem (10), it thus follows that it is optimal. □

EC.2.2. Solving the LO relaxation of the Benders problem using classical constraint generation

Analogously to the integer problem, we can solve the LO relaxation of the Benders formulation using a classical constraint generation approach. We define the restricted master problem as:

$$\underset{\mathbf{x}, \mathbf{t}}{\text{maximize}} \quad \sum_{k=1}^K \lambda^k t_k \tag{EC.30a}$$

$$\text{subject to} \quad t_k \leq \gamma^k + \sum_{i=1}^n \alpha_i^k \cdot x_i + \sum_{i=1}^n \beta_i^k \cdot (1 - x_i), \quad \forall\, k \in \{1, \dots, K\},\ (\boldsymbol{\alpha}^k, \boldsymbol{\beta}^k, \gamma^k) \in \bar{A}_k, \tag{EC.30b}$$

$$\mathbf{C}\mathbf{x} \leq \mathbf{d}, \tag{EC.30c}$$

$$0 \leq x_i \leq 1, \quad \forall\, i \in \{1, \dots, n\}. \tag{EC.30d}$$

Algorithm 4 provides the classical constraint generation procedure for solving the LO relaxation of the Benders problem. This algorithm is almost exactly the same as Algorithm 3; the only difference is that instead of problem (EC.29), we solve problem (EC.30), and instead of using equations (11a) – (11c) to find the primal and dual subproblem solutions, we instead apply Algorithms 1 and 2.

Like Algorithm 3, Algorithm 4 is guaranteed to obtain the optimal solution of problem (12). We formalize this as the proposition below; we omit the proof, since it is almost identical to that of Proposition EC.3.

PROPOSITION EC.4. *Let $(\mathbf{x}, \mathbf{t})$ be the solution obtained upon termination of Algorithm 4. Then $(\mathbf{x}, \mathbf{t})$ is an optimal solution of problem* (12).

Algorithm 4 Classical constraint generation algorithm for solving relaxation of Benders formulation

Set $\bar{A}_k = \emptyset$ for all customer types $k \in \{1, \ldots, K\}$.
Solve the restricted master problem (EC.30) to obtain a solution $(\mathbf{x}, \mathbf{t})$.
For each $k \in \{1, \ldots, K\}$:
  Run Algorithms 1 and 2 with $\mathbf{x}$ to obtain a primal solution $\mathbf{y}^k$ and a dual solution $(\boldsymbol{\alpha}^k, \boldsymbol{\beta}^k, \gamma^k)$.
while $\max_{k=1,\ldots,K}(t_k - \boldsymbol{\pi}^T \mathbf{y}^k) > 0$ **do**
  For each $k \in \{1, \ldots, K\}$ with $\boldsymbol{\pi}^T \mathbf{y}^k < t_k$:
    Set $\bar{A}_k \leftarrow \bar{A}_k \cup \{(\boldsymbol{\alpha}^k, \boldsymbol{\beta}^k, \gamma^k)\}$.
  Solve restricted master problem (EC.30) with $\bar{A}_1, \ldots, \bar{A}_K$ to obtain $(\mathbf{x}, \mathbf{t})$.
  For each $k \in \{1, \ldots, K\}$:
    Run Algorithms 1 and 2 with $\mathbf{x}$ to obtain a primal solution $\mathbf{y}^k$ and a dual solution $(\boldsymbol{\alpha}^k, \boldsymbol{\beta}^k, \gamma^k)$.
end while
return Optimal solution $(\mathbf{x}, \mathbf{t})$ of problem (10).

EC.2.3. Finite convergence of constraint generation

Algorithms 3 and 4 are guaranteed to provide optimal solutions upon termination to the integer problem and the LO relaxation problem, respectively. In this section, we establish that both algorithms are guaranteed to terminate in finitely many iterations. The first result that we establish is that the primal and dual subproblem solutions produced by Algorithms 1 and 2 are guaranteed to be extreme points of their respective subproblems. As with our definitions of Algorithms 1 and 2, we develop the result in terms of a ranking σ and drop the index k to lighten notation.

THEOREM EC.1. *Let $\sigma : \{0, 1, \ldots, n\} \to \{0, 1, \ldots, n\}$ and $\mathbf{x} \in [0, 1]^n$. Let $\mathbf{y}$ be the solution produced by Algorithm 1, and $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma)$ be the solution produced by Algorithm 2. Then $\mathbf{y}$ is an extreme point of the primal problem* (13) *and $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma)$ is an extreme point of the dual problem* (14).

Proof of Theorem EC.1: We prove the result separately for the primal and dual solutions.

Primal solution is an extreme point: To prove this, we will proceed directly from the definition of an extreme point. A point $\mathbf{z}$ in a polyhedron P is said to be an extreme point if there do *not* exist two points, $\mathbf{z}^1, \mathbf{z}^2 \neq \mathbf{z}$ and a real number $\theta \in (0, 1)$ such that $\mathbf{z} = \theta \mathbf{z}^1 + (1 - \theta)\mathbf{z}^2$.

We will prove this by contradiction. Let us suppose that $\mathbf{y}$ is not an extreme point. Then there exist $\mathbf{y}^1, \mathbf{y}^2 \neq \mathbf{y}$ and a scalar $\theta \in (0, 1)$ such that $\mathbf{y} = \theta \mathbf{y}^1 + (1 - \theta)\mathbf{y}^2$. Let τ be the same ordering used in Algorithm 1. Define the index s^* as

$$s^* = \min\{s \in \{0, 1, \ldots, n\} \mid y^1_{\tau(s^*)} \neq y_{\tau(s^*)} \text{ or } y^2_{\tau(s^*)} \neq y_{\tau(s^*)}\},$$

i.e., it is the first stage of Algorithm 1 at which a coordinate of $\mathbf{y}$ differs from $\mathbf{y}^1$ or $\mathbf{y}^2$. Note that the set defining the minimum cannot be empty, because otherwise both $\mathbf{y}^1$ and $\mathbf{y}^2$ would be equal to $\mathbf{y}$. At s^*, we must have that both $y^1_{\tau(s^*)} \neq y_{\tau(s^*)}$ and $y^2_{\tau(s^*)} \neq y_{\tau(s^*)}$ (both must be different from $y_{\tau(s^*)}$, because if only one is distinct, then their convex combination could not be equal to $y_{\tau(s^*)}$). Note also that s^* cannot happen after the C event in Algorithm 1 (if s^* is after the C event, then $y_{\tau(s^*)} = 0$, and since $\mathbf{y}^1$ and $\mathbf{y}^2$ must be nonnegative, this would again imply that both $\mathbf{y}^1$ and $\mathbf{y}^2$ would have to be equal to $\mathbf{y}$).

Without loss of generality, let us assume that $y^1_{\tau(s^*)} < y_{\tau(s^*)} < y^2_{\tau(s^*)}$. We now argue that $\mathbf{y}^2$ cannot be feasible. Observe that, as we proceed from $s = 0$ to $s = s^* - 1$, the coordinates of $\mathbf{y}$ and $\mathbf{y}^2$ are set the same way. Algorithm 1 always sets each coordinate to the largest possible value it can be set that maintains the feasibility of all constraints. Thus, at $s = s^*$, $y_{\tau(s^*)}$ is set to the largest feasible value based on the current values of all the variables; if $y^2_{\tau(s^*)} > y_{\tau(s^*)}$, then that implies that $\mathbf{y}^2$ cannot be feasible:

- If an A event occurred, then this would mean that $y^2_{\tau(s^*)} > x_{\tau(s^*)}$;
- If a C event occurred, then this would mean that $\sum_{s=0}^{s^*} y^2_{\tau(s^*)} > \sum_{s=0}^{s^*} y_{\tau(s^*)} = 1$, which implies that $\sum_{j=0}^{n} y_j^2 > 1$; and
- If a B event occurred, with i^* as the corresponding preference inequality that became tight, then this would mean that

$$1 - x_{i^*} = \sum_{\substack{s=0:\\ \sigma(\tau(s))<\sigma(i^*)}}^{s^*} y_{\tau(s)} < \sum_{\substack{s=0:\\ \sigma(\tau(s))<\sigma(i^*)}}^{s^*} y^2_{\tau(s)},$$

which implies that $\sum_{j:\sigma(j)<\sigma(i^*)} y_j^2 > 1 - x_{i^*}$.

We thus have a contradiction, and it must be that $\mathbf{y}$ is an extreme point of the primal problem.

Dual solution is an extreme point: To prove this, we will use the equivalence between extreme points and basic feasible solutions (see Theorem 2.3 of Bertsimas and Tsitsiklis 1997), and show that $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma)$ is a basic feasible solution. A feasible solution $\mathbf{z}$ of a polyhedron $P = \{\mathbf{z} \in \mathbb{R}^m \mid \mathbf{Az} \leq \mathbf{b}\}$ is a basic feasible solution if there are m linearly independent active constraints at $\mathbf{z}$.

Consider the following system of equations:

$$\gamma + \alpha_C + \sum_{j:\sigma(j)<\sigma(C)} \beta_j = \pi_C, \tag{EC.31}$$

$$\gamma + \alpha_{f(i_t)} + \sum_{j:\sigma(j)<\sigma(f(i_t))} \beta_j = \pi_{f(i_t)}, \quad \forall\, t \in \{1, \ldots, |B_{main}|\}, \tag{EC.32}$$

$$\gamma + \alpha_i + \sum_{j:\sigma(j)<\sigma(i)} \beta_j = \pi_i, \quad \forall\, i \in A, \tag{EC.33}$$

$$\alpha_i = 0, \quad \forall i \in \{0, 1, \ldots, n\} \setminus A, \tag{EC.34}$$

$$\beta_j = 0, \quad \forall j \in \{0, 1, \ldots, n\} \setminus \{i_1, \ldots, i_{|B_{main}|}\}. \tag{EC.35}$$

We note that there are $1 + |B_{main}| + |A| + (n + 1 - |A|) + (n + 1 - |B_{main}|) = 2n + 3$ equations, which is exactly the number of variables in the dual problem. These equations are constraints from problem (14) that are made to hold at equality. We now show that this system of equations implies a unique solution, establishing that the system of equations is linearly independent, and we show that the solution produced by our algorithm coincides with this solution.

First, observe that equations (EC.34) imply that $\alpha_i = 0$ for all $i \notin A$. This is also true at the end of Algorithm 2, because all α_i's are initially set to zero, and the only ones that are potentially changed from zero are those with $i \in A$. Similarly, equations (EC.34) – (EC.35) imply that $\beta_j = 0$ for any $j \notin B_{main}$, which is also true at the end of Algorithm 2.

Second, let us establish that γ must be equal to π_C. To do so, we observe the following:

- In Algorithm 1, suppose a C event occurs at stage s. Then the corresponding q^* must satisfy $q^* > 0$. If this is not the case, then $q^* = 0$, which would mean that the equality $\sum_{j=0}^{n} y_j = 1$ became true before the current stage s. However, a C event would then have to have been triggered before stage s, which cannot be true.
- Using the above observation, we now show that for any $i_t \in B_{main}$, we must have $\sigma(i_t) \geq \sigma(C)$. To see this, suppose that there is an $i_t \in B_{main}$ such that $\sigma(i_t) < \sigma(C)$. This would imply that the preference inequality for option i_t became tight before the C event was triggered. However, since the preference inequality became tight and C is less preferred than i_t, this would mean that q^* would have to be zero in the iteration in which the C event is triggered, which is not possible.

These observations, together with the fact that $\beta_j = 0$ for any $j \notin \{i_1, \ldots, i_{|B_{main}|}\}$ (this is a consequence of equations (EC.35)) and the fact that $\alpha_C = 0$ (this follows from (EC.34) and the fact that $C \notin A$, since at most one type of event can occur at each stage), equation (EC.31) implies that:

$$\begin{aligned} \gamma + \alpha_C + \sum_{j:\sigma(j)<\sigma(C)} \beta_j &= \gamma + 0 + 0 \\ &= \gamma \\ &= \pi_C. \end{aligned}$$

At the end of Algorithm 2, γ is also set to π_C.

Third, we handle $\beta_{i_1}, \dots, \beta_{i_{|B_{main}|}}$. Observe that by using equations (EC.34) and (EC.35), together with the fact that $f(i_1), f(i_2), \dots, f(i_{|B_{main}|}) \notin A$ (this is true because at most one type of event can occur at each stage), we can simplify equations (EC.32) to

$$\begin{aligned} \gamma + \beta_{i_1} &= \pi_{f(i_1)} \\ \gamma + \beta_{i_1} + \beta_{i_2} &= \pi_{f(i_2)} \\ \gamma + \beta_{i_1} + \beta_{i_2} + \beta_{i_3} &= \pi_{f(i_3)} \\ &\vdots \\ \gamma + \beta_{i_1} + \beta_{i_2} + \dots + \beta_{i_{|B_{main}|}} &= \pi_{f(i_{|B_{main}|})} \end{aligned}$$

Notice that the unique solution to this system of equations is exactly given by the first loop of Algorithm 2. Since our algorithm also sets γ to π_C, it follows that any solution to (EC.31) – (EC.35) must match the solution created by our algorithm for $\beta_{i_1}, \dots, \beta_{i_{|B_{main}|}}$.

Finally, we handle α_i for $i \in A$. The corresponding equations are given by (EC.33), which uniquely determine α_i for each $i \in A$ to be

$$\alpha_i = \pi_i - \gamma - \sum_{j:\sigma(j)<\sigma(i)} \beta_j.$$

These are exactly the same values set by the second loop of Algorithm 2.

We have thus established that the unique solution to (EC.31) – (EC.35) must exactly coincide with the solution produced by our algorithm. This establishes that the solution $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma)$ of Algorithm 2 is a basic feasible solution or equivalently, an extreme point of the dual problem. □

A corollary of this theorem is that the primal and dual subproblem solutions for the integer problem, which are defined in equations (8) and equations (11a) – (11c) respectively, are also extreme points.

Corollary EC.1. *Let* $\mathbf{x} \in \{0,1\}^n$ *and* $k \in \{1, \dots, K\}$. *Let* $\mathbf{y}$ *be the primal subproblem solution specified by equation* (8) *and* $(\boldsymbol{\alpha}^k, \boldsymbol{\beta}^k, \gamma^k)$ *be the dual subproblem solution specified by equations* (11a) – (11c). *Then* $\mathbf{y}$ *is an extreme point of the primal problem* (13) *and* $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma)$ *is an extreme point of the dual problem* (14).

Proof of Corollary EC.1: It can be shown that the primal solution specified equations (8) and the dual solution specified by equations (11a) – (11c) coincide with the primal and dual solutions produced by Algorithms 1 and 2, respectively, when $\mathbf{x}$ is binary. The result then follows from Theorem EC.1. □

With these results, we are now ready to prove that both constraint generation procedures terminate in finitely many iterations.

Proposition EC.5. *Algorithms 3 and 4 terminate in finitely many iterations.*

Proof of Proposition EC.5: Let us consider Algorithm 4 for the LO relaxation; the proof for Algorithm 3 follows in the same way. By Algorithm 4, we generate at most one dual subproblem solution $(\boldsymbol{\alpha}^k, \boldsymbol{\beta}^k, \gamma^k)$ corresponding to a violated constraint for each customer type k. For a given k, once the subproblem solution $(\boldsymbol{\alpha}^k, \boldsymbol{\beta}^k, \gamma^k)$ is added to $\bar{A}_k$, any solution of the restricted master problem in subsequent iterations must satisfy constraint (10b) at $(\boldsymbol{\alpha}^k, \boldsymbol{\beta}^k, \gamma^k)$; thus, the solution $(\boldsymbol{\alpha}^k, \boldsymbol{\beta}^k, \gamma^k)$ will not be generated again in subsequent iterations. This means that for a given k, each iteration of Algorithm 4 generates a distinct solution $(\boldsymbol{\alpha}^k, \boldsymbol{\beta}^k, \gamma^k)$. By Theorem EC.1, this implies that for a given k, each iteration generates a distinct extreme point of the polyhedron A_k. By standard results in linear optimization theory (see Corollary 2.1 of Bertsimas and Tsitsiklis 1997), the polyhedron A_k of feasible dual subproblem solutions for customer type k has finitely many extreme points. Since there are finitely many customer types k, Algorithm 4 must terminate in finitely many iterations.

The proof for Algorithm 3 proceeds in the same way, with the difference that Corollary EC.1 guarantees that each dual subproblem solution is an extreme point. □

EC.3. Additional results

EC.3.1. Formulation comparisons with Toubia et al. (2003) data

In this section, we report on additional experiments to compare the solvability and strength of the different formulations, using instances derived from the Toubia et al. (2003) data. For these experiments, we tested $n \in \{20, 50, 100\}$ and $K = 100$. For each (n, K), we created 20 instances as follows. We sampled n products randomly without replacement from the full set of 3584 products (as described in Section 5.3), and sampled K customers without replacement from the full set of 330 customers. The utilities for this random subset of products and the no-purchase option for this random subset of customers were calculated in exactly the same way as in Section 5.3. The marginal profit of each product was also computed the same way. For each instance, we tested the constrained formulation with the constraint $\sum_i x_i \leq b$ for $b \in \{5, 10\}$, as well as the unconstrained formulation.

Table EC.1 compares the average integrality gap of the relaxations of the four MIO formulations of the first-choice PLD problem, while Table EC.2 compares the average solution time (for full optimality) for the four MIO formulations. From these tables, we observe the same behavior as with the synthetic instances, namely that our formulation (problem (2)) produces the tightest LO relaxation bounds and requires the least amount of time to solve to full optimality.

Table EC.1 Comparison of LO bounds for instances derived from Toubia et al. (2003) data.

n	K	b	G_{BM}	G_{BFSS}	$G_{Utility}$	G_{MZ}
20	100	5	2.71	7.15	15.83	23.23
20	100	10	2.10	7.36	14.68	21.81
20	100	–	2.10	7.36	14.68	21.81
50	100	5	3.44	7.51	21.63	40.94
50	100	10	3.01	8.96	18.75	33.72
50	100	–	2.43	9.29	17.90	32.32
100	100	5	3.69	6.94	20.22	35.47
100	100	10	3.28	7.99	16.24	27.36
100	100	–	2.69	8.51	14.76	24.95

Table EC.2 Comparison of MIO solution times for instances derived from Toubia et al. (2003) data.

n	K	b	T_{BM}	T_{BFSS}	$T_{Utility}$	T_{MZ}
20	100	5	0.40	0.70	1.72	1.40
20	100	10	0.37	0.70	1.22	1.22
20	100	–	0.37	0.76	1.14	1.21
50	100	5	1.44	2.33	11.38	7.95
50	100	10	1.37	3.66	13.91	7.56
50	100	–	1.04	3.36	8.88	6.13
100	100	5	5.12	9.39	88.01	119.27
100	100	10	5.51	18.58	147.75	146.73
100	100	–	3.79	39.65	215.83	126.19

EC.3.2. Additional Benders results without warm-starting

In this set of additional experiments, we test out our Benders method from Section 4 (the LO relaxation phase and the integer phase) on the same real PLD instance from Section 5.3. The method was tested in exactly the same way as in Section 5.3, except that we do not run the divide and conquer algorithm before the Benders method, and so we do not warm-start the Benders method with an initial integer solution. Table EC.3 shows the results without this warm-starting. From this table, we can see that for the smaller instances, the time required to prove optimality is in general larger – for example, for $\sum x_i = 5$, the warm-started approach from Section 5.3 could solve the problem in around 10 minutes, whereas without warm-starting, it takes about 17 minutes. Similarly, for the larger instances, the final optimality gap is larger without warm-starting than it is with warm-starting. For example, for $\sum x_i = 50$, we obtain a final optimality gap of 3.09% with warm-starting, whereas without it, we obtain a gap of 5.73%. The main takeaway from this experiment is that providing a high-quality integer solution to the Benders method can be very beneficial, allowing for the problem to be solved more quickly (in smaller instances) or allowing a better final optimality gap (in larger instances).

Table EC.3 Results of large-scale Benders experiment without warm-starting with D&C solution.

	LO Benders phase		MIO Benders phase			Final
Constraints	Bound	Time (s)	Obj.	Bound	Time (s)	Gap (%)
$\sum x_i = 2$	59.22	175.59	59.22	59.22	54.26	0.00
$\sum x_i = 3$	66.57	268.96	66.29	66.29	109.02	0.00
$\sum x_i = 4$	71.21	325.45	70.24	70.24	580.42	0.00
$\sum x_i = 5$	73.85	392.79	72.82	72.82	650.01	0.00
$\sum x_i = 10$	79.36	553.34	77.41	77.41	14934.88	0.01
$\sum x_i = 20$	82.65	914.75	77.18	81.81	21617.32	5.67
$\sum x_i = 50$	85.01	2572.22	79.64	84.48	21600.09	5.73