Models and Algorithms for the Product Pricing with Single-Minded Customers Requesting Bundles

Víctor Bucarey[1,2], Sourour Elloumi[3,4], Martine Labbé[1,2], and Fränk Plein[1,2]

[1]Département d'Informatique, Université Libre de Bruxelles, Brussels, Belgium.
[2]Inria Lille-Nord Europe, Villeneuve d'Ascq, France.
{vbucarey, mlabbe, fplein}@ulb.ac.be
[3]UMA-Ensta Paris, Paris, France.
[4]CEDRIC-Cnam, Paris, France.
sourour.elloumi@ensta-paristech.fr

October 30, 2020

Abstract

We analyze a product pricing problem with single-minded customers, each interested in buying a bundle of products, if the total price is less than or equal to their budget. The objective of the seller is to maximize the total revenue and we assume that supply is unlimited for all products. We contribute to a missing piece of literature by giving some mathematical formulations for this single-minded bundle pricing problem. We first present a mixed-integer nonlinear program with bilinear terms in the objective function and the constraints. By applying classical linearization techniques, we obtain two different mixed-integer linear reformulations. Inspired by a reformulation-linearization technique framework, we derive valid inequalities leading to a tighter linear reformulation. We then discuss a Benders decomposition to project strong cuts from the tightest model onto a light and fast model. In another attempt to exploit the tighter linear relaxation, we discuss heuristics based on the developed mixed-integer linear formulations to quickly find good solutions. We conclude this work with extensive numerical experiments to assess the quality of the mixed-integer linear formulations, as well as the performance of the Benders decomposition and the heuristics.
Keywords: Pricing Problems, Integer Programming Formulations, Benders Decomposition, Heuristics

1 Introduction

Pricing problems generally aim to determine the best price policy for a determined set of products to maximize the revenue of a company. They therefore exhibit a hierarchical interaction between a seller and their customers. In the Single-Minded Bundle Pricing Problem (SMBPP), the seller decides product prices under a model of single-minded customer behavior, determined by a bundle and a budget. A bundle is a subset of products, that a given customer wants to purchase, and they are willing to pay at most their budget. The single-mindedness of a client translates into purchasing the bundle if and only if the corresponding total price does not exceed the budget. The objective of the seller is to maximize the total revenue.

The SMBPP is an instance of pricing problem, where a reservation price customer behavior is assumed, as opposed to a multinomial-logit approach that is often found in the revenue management literature, see, e.g., Hanson and Martin 1996; Aydin and Ryan 2000; Pierson et al. 2013. However, there is a link between both models exhibited by Mayer and Gönsch 2012. Under the model of reservation price, the clients' behavior is determined by the optimization of a certain utility function. Rusmevichientong et al. 2006 introduce maximum and minimum utility functions, as well as a maximum rank utility function. In the SMBPP, clients can be seen to maximize a utility function given by the difference of their budget minus the total price of their bundle. Talluri and Van Ryzin 2006 give an overview of various other client behavior models.

For limited product supplies, in addition to the seller deciding a best price policy, an assignment of bundles of products to the requesting customers needs to be determined. From that perspective, the notion of envy-free assignment is discussed by Guruswami et al. 2005. Clients might also be willing to purchase bundles multiple times with a budget that is a function of the number of received bundles. For this variant, Zhang et al. 2020 have recently proposed offline and online algorithms. In this work, we focus on a simplified setting with unlimited supplies and where customers only request their bundle once. One possible example is the pricing of digital goods. Another interesting application is to price different features of goods or services instead of products. In this context, a bundle would be the set of characteristics a client requires for his purchase. Note that when supplies are unlimited, every assignment is automatically envy-free. Fiat and Wingarten 2009 study a problem similar to the SMBPP with the exception that prices are fixed for subsets of products, as opposed to each individual product. They show that under the assumptions of unlimited supply, this problem can be solved in polynomial time.

Other problems that have attracted great interest in the literature and that share a similar combinatorial structure with the SMBPP are combinatorial auctions. See, for instance, Ledyard 2007 and the references therein. Bidders request a bundle of products and their bids are the maximum price they are willing to pay. The goal of auctioneers is however not to price their goods, but rather to determine an envy-free assignment of a limited supply to bidders that maximizes their revenue.

The SMBPP considered in this work, i.e., pricing individual products with unlimited supply and customers having a budget for a single bundle, has been studied in the literature for its complexity-theoretical aspects. Grigoriev et al. 2008 show that the SMBPP is NP-hard, even when bundles have size 2. From their analysis, the authors then devise a PTAS in some special cases. Guruswami et al. 2005 and Briest and Krysta 2006 give two different proofs that the SMBPP is APX-hard and Balcan and Blum 2006 present a 4-approximation algorithm. Further research on the hardness of approximation has been carried out by Khandekar et al. 2009, improving the best-known approximation bounds. The SMBPP is also a central topic of the PhD thesis of Van Loon 2009. The author shows, among others, that the SMBPP can be solved in polynomial time, if the average budget, i.e., budget divided by bundle size, is the same for all clients. However, if there is only a slight variation in average budgets, the problem is again shown to be NP-hard.

To the best of our knowledge, the SMBPP has not yet been tackled using exact approaches based on mathematical optimization. Mathematical formulations for pricing problems have attracted the attention of the literature in the recent years, see for instance recent work by Calvete et al. 2019. This work contributes to the literature by proposing linear and nonlinear formulations for the SMBPP. The problem has been previously studied by Plein 2017 in his Master's thesis and the results of this paper are an extension of the results obtained therein. To get stronger formulations, valid inequalities and a polyhedral study are presented. Also, a Benders reformulation is presented to increase the scalability of our formulations. We study a Benders decomposition to generate cuts to strengthen light and fast formulations. Moreover, we propose and study two families of heuristics based on the LP relaxation of the formulations stated in this work. An extensive computational study is presented to shed light on the difficulty of the SMBPP depending on the size of the bundles and the client budget distribution.

The structure of the paper is as follows. The problem statement and notation is introduced in Section 2. In Section 3, we introduce basic notations and give new mixed-integer nonlinear programming (MINLP) and mixed-integer linear (MILP) formulations for the SMBPP. Furthermore, based on an RLT-like approach, we obtain valid inequalities that lead to a new tighter MILP formulation. In Section 4, we present an algorithmic discussion. First, in Section 4.1 presents a Benders reformulation of the tightest MILP formulation. Then, in Section 4.2 we present two families of heuristic based in solving the LP-relaxation of the formulations proposed. In Section 5, we carry out an extensive computational study and summarize the main results. Finally, we present our conclusions and discuss future work in Section 6.

2 Problem Statement

Let $N = \{1, \dots, n\}$ be a set of products with unlimited supply and let $M = \{1, \dots, m\}$ be a set of clients. We consider a reservation price model with single-minded customers, i.e., each client $j \in M$ is entirely represented by a bundle $S^j \subseteq N$ and a budget $b^j > 0$. Client bundles can be encoded in a characteristic matrix $S \in \mathbb{R}^{n \times m}$ with

$$S_i^j := \begin{cases} 1, & \text{if product } i \in S^j, \\ 0, & \text{otherwise,} \end{cases}$$

for product $i \in N$ and client $j \in M$. On the other hand, all budget are collected in a vector $b \in \mathbb{R}^m_{>0}$. Under single-minded customer behavior, a client purchases his bundle whenever its total price is less than or equal to his budget. Here, the total price of a bundle is given by the sum of the prices of the individual products that constitute it.

The *single-minded bundle pricing problem (SMBPP)* consists of determining prices $p_i \geq 0$, for each product $i \in N$, that maximize the total revenue obtained by selling bundles to clients. The total bundle price is denoted by

$$p(S^j) := \sum_{i \in S^j} p_i.$$

A client purchases his bundle if and only if the bundle price does not exceed his budget, so when $p(S^j) \leq b^j$. The SBMPP which maximizes the total revenue is

$$\max_{p \geq 0} \sum_{j \in M} \text{Revenue}_j(p)$$

where $\text{Revenue}_j(p)$, is the revenue obtained from client $j \in M$, which takes value $p(S^j)$ if $p(S^j) \leq b^j$ and 0 otherwise.

Example 1. Take for instance the following input data

$$S = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \qquad\qquad b = \begin{pmatrix} 2 & 3 & 4 \end{pmatrix}$$

We then need to solve the SMBPP with two products $N = \{1, 2\}$ and three clients $M = \{1, 2, 3\}$. A first client desires to purchase both products and is willing to pay at most 2 price units in total. A second client seeks to purchase product 1 for at most 3 price units, whereas the third client requests product 2 at a budget of 4. The single-mindedness implies, for instance, that the first client purchases his bundle if and only if its total price $p(S^1) = p_1 + p_2$ is at most 2. In this example, the optimal solution is obtained by setting $p_1 = 3$ and $p_2 = 4$. Then, client 1 does not purchase, and clients 2 and 3 purchase, resulting in a total profit of 7.

3 Formulations

In this section, we first derive a mixed-integer nonlinear programming (MINLP) formulation for the SMBPP involving products of price variables and buying decision variables. Next, we discuss bounds on the prices. We then present two different mixed-integer linear programming (MILP) reformulations obtained by linearizing the nonlinear terms in two different ways.

3.1 MINLP Formulation

We start by giving a MINLP formulation that is directly derived from the problem description. We introduce binary variables $x_j \in \{0, 1\}$ for every client's purchase decision, i.e. $x_j = 1$ if and only if customer $j \in M$ purchases his bundle S^j.

$$\max_{p,x} \quad \sum_{j \in M} \sum_{i \in S^j} p_i x_j \tag{1a}$$

$$\text{s.t.} \quad x_j (\sum_{i \in S^j} p_i - b^j) \leq 0, \quad j \in M, \tag{1b}$$

$$(1 - x_j)(\sum_{i \in S^j} p_i - b^j) \geq 0, \quad j \in M, \tag{1c}$$

$$p_i \geq 0, \quad i \in N, \tag{1d}$$

$$x_j \in \{0, 1\}, \quad j \in M. \tag{1e}$$

The objective is to maximize the total profit given by $\sum_{j\in M} p(S^j)x_j = \sum_{j\in M}\sum_{i\in S^j} p_i x_j$. Constraints (1b) ensure that if the total bundle price is greater than the budget, i.e., $p(S^j) > b^j$, then client $j \in M$ cannot purchase, i.e., $x_j = 0$. Conversely, Constraints (1c) ensure that, in a profit maximizing solution, $x_j = 1$ if $p(S^j) \le b^j$. Note that by optimality, Constraints (1c) are always satisfied, because as soon as $p(S^j) \le b^j$, it is more profitable to set $x_j = 1$. We will thus often use the following formulation

$$\text{(NLM)} \quad \max_{p_i, x_j} \quad \sum_{j\in M}\sum_{i\in S^j} p_i x_j \tag{2a}$$
$$\text{s.t.} \quad (1b), (1d), (1e).$$

However, there exist feasible points for this formulation where clients do not purchase their bundle if $p\left(S^j\right) \le b^j$. Nonetheless, an optimal solution of this relaxed formulation has the same value as an optimal solution of the initial formulation. Furthermore, observe that by optimality, variables x_j always take a binary value even when (NLM) is solved over $[0, 1]$. Finally, the problem is always bounded since the maximum revenue cannot be larger than $\sum_{j\in M} b^j$. Furthermore, the revenue corresponding to an optimal solution is at least $\max_{j\in M} b^j$.

Remark 2 (Problem decomposition and reduction)**.** Observe that an instance of the SMBPP, as described in Section 2, can be represented by a bipartite graph where S plays the role of adjacency matrix. In other words, for a given instance of the SMBPP we can build a bipartite graph with one set of nodes for clients and products, and edges between client j and product i if and only if $S_i^j = 1$. It is not hard to see that the SMBPP can be decomposed, solving one smaller SMBPP instance for each connected component in the graph. For example, consider the following instance:

$$S = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} \qquad b = \begin{pmatrix} 10 & 50 & 20 & 15 \end{pmatrix}.$$

The corresponding bipartite graph is shown in Figure 1. The SMBPP for this instance can be solved by splitting the problem into two sub-problems: The first considering clients 1 and 4 and product 1 (dashed edges), and the second considering clients 2 and 3 and products 2 and 3 (solid edges). However, all the instances considered in this article are composed of a single connected component.

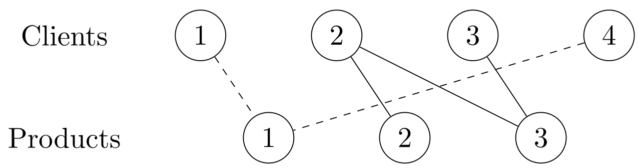

Figure 1: Bipartite graph representation of a SMBPP instance.

Also, the set of products can be reduced, whenever there exists a subset of products $\tilde{N} \subseteq N$ that appear only together in client bundles. In that case, these products can be considered as a single product with $p_{\tilde{N}} := \sum_{i\in\tilde{N}} p_i$. For example, in the following instance, matrix S indicates that products 2 and 3 appear together in bundles of clients 1 and 2, then they can be treated as one single product. The resulting matrix S' has one row less.

$$S = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \qquad S' = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

3.2 Implicit Upper Bound on Prices

To tackle the nonlinear and nonconvex model (NLM), we propose to linearize the product terms of continuous variables p_i by binary variables x_j. To that end, we first need an upper bound on the price variables. Let us denote by U_i the upper bound on variable p_i for $i \in N$. Although, prices p_i are not bounded a priori, it is not hard to observe that we can set

$$U_i := \max_{j \in M} \left\{ b^j : i \in S^j \right\}.$$

If we fix $p_i > U_i$, any client $j \in M$ with bundle $S^j \ni i$ is unable to purchase. Hence, product i is never bought. Consequently, $p_i \leq U_i$ holds in any optimal solution. We give an example to convince the reader that these bounds are tight in general.

Example 3. Consider the instance of 1 product and 2 clients given by the following input:

$$S = \begin{bmatrix} 1 & 1 \end{bmatrix} \qquad b = \begin{pmatrix} 1 & 10 \end{pmatrix}$$

Both clients want to purchase the same product but with different budgets. It is straightforward to see that an optimal solution yields a revenue of 10 and is obtained by setting $p = 10 = U$, so that $x_1 = 0$ and $x_2 = 1$.

For a subset of products $S \subseteq N$, denote by $U(S)$ the upper bound on the total price $p(S) := \sum_{i \in S} p_i$ of the subset S. As a natural extension, we define $U(S) := \sum_{i \in S} U_i$.

3.3 Aggregated and Disaggregated MILP Formulations

To obtain a first linear reformulation, we linearize the objective function and the constraints of (NLM) by replacing the products of total bundle price and buying decision using McCormick inequalities (McCormick 1976). To that end, we introduce new variables $r_j := p(S^j)x_j = \sum_{i \in S^j} p_i x_j$ for every $j \in M$ and obtain a first MILP formulation,

$$\begin{aligned}
(\mathrm{LM}_1) \quad \max_{p,x,r} \quad & \sum_{j \in M} r_j & \text{(3a)}\\
\text{s.t.} \quad & r_j \leq b^j x_j, \quad j \in M, & \text{(3b)}\\
& r_j \leq p(S^j), \quad j \in M, & \text{(3c)}\\
& r_j \geq p(S^j) - U(S^j)(1 - x_j), \quad j \in M, & \text{(3d)}\\
& r_j \geq 0, \quad j \in M, & \text{(3e)}\\
& p_i \geq 0, \quad i \in N, &\\
& x_j \in \{0, 1\}, \quad j \in M. &
\end{aligned}$$

Here, Constraints (3b) are the linearized Constraints (1b). In particular, they dominate the additional McCormick inequalities $r_j \leq U(S^j)x_j$, which are redundant. For all clients $j \in M$, the new variables r_j can be interpreted as $\mathrm{Revenue}_j(p)$ and represent the aggregation of quadratic terms $p_i x_j$ for all $i \in S^j$, i.e., $\sum_{i \in S^j} p_i x_j$.

In a second reformulation, we can be more precise in our linearization of (NLM) and replace all individual products $p_i x_j$ for any $i \in S^j$ and $j \in M$. Therefore, we introduce variables

$$s_{ij} := p_i x_j, \quad j \in M, i \in S^j. \tag{4}$$

We obtain a disaggregated formulation as opposed to aggregated formulation (LM_1).

$$\begin{aligned}
(\mathrm{LM}_2) \quad \max_{p,x,s} \quad & \sum_{j \in M} \sum_{i \in S^j} s_{ij} & \text{(5a)}\\
\text{s.t.} \quad & \sum_{i \in S^j} s_{ij} \leq b^j x_j, \quad j \in M, & \text{(5b)}\\
& s_{ij} \leq p_i, \quad i \in S^j, j \in M, & \text{(5c)}\\
& s_{ij} \geq p_i - U_i(1 - x_j), \quad i \in S^j, j \in M, & \text{(5d)}\\
& s_{ij} \geq 0, \quad i \in S^j, j \in M, & \text{(5e)}\\
& p_i \geq 0, \quad i \in N, &\\
& x_j \in \{0, 1\}, \quad j \in M. &
\end{aligned}$$

We only point out that for $j \in M$ and $i \in S^j$, Constraints (5b) imply that $s_{ij} \leq b^j x_j$ and hence, $s_{ij} \leq U_i x_j$ are redundant by definition of U_i. An in-depth polyhedral analysis of both formulations is presented in the Master's thesis Plein 2017. Here, we only remark that if we denote by $v(\cdot)$ the optimal value of the linear relaxation of a given formulation, then $v(\text{LM}_2) \leq v(\text{LM}_1)$. For a given feasible point (p, x, s) of the linear relaxation of (LM_2), we can always construct a feasible point (p, x, r) of the linear relaxation of (LM_1) with the same objective value by setting

$$r_j = \sum_{i \in S^j} s_{ij}, \quad j \in M.$$

Note also that $v(\text{LM}_2) < v(\text{LM}_1)$ may hold, as we show with the following example.

Example 4. Consider an instance of 2 products and 2 clients given by

$$S = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \qquad b = \begin{pmatrix} 10 & 20 \end{pmatrix}.$$

The optimal solution of the linear relaxation of (LM_1) is obtained by setting $x = (\frac{1}{2}, 1)$ and $p = (20, 0)$ yielding a revenue of $v(\text{LM}_1) = 25$. On the other hand, $v(\text{LM}_2) = 20$. The optimal prices are the same, but the fractional solution of (LM_1) is cut off by constraints $s_{11} \geq p_1 - 20(1 - x_1)$, $s_{11} + s_{12} \leq 10x_1$ and the fact that $s_{12} \geq 0$.

Finally, note that an additional linearization has been studied in Plein 2017. Only the constraints of (NLM) have been linearized, resulting in a MINLP formulation with bilinear objective function and linear constraints. Compared to the linear formulations studied in this paper, the MINLP model did not perform well, which is why it has been omitted in the present work.

3.4 Valid Inequalities and a New Formulation

The two linear formulations presented thus far are both based on linearizations of (NLM). In particular, by definition of U_i, their linear relaxations are less tight when products appear in more bundles, i.e., when the density of the characteristic matrix S increases. Furthermore, observe that buying decisions of clients $j, k \in M$ are not explicitly linked. The only relation between two clients with intersecting bundles is established through the corresponding prices. We intend to derive valid inequalities that create an explicit link using reformulation-linearization techniques (RLT) (Sherali and Adams 2013). These RLT-like valid inequalities are given by

$$(p(S^k) - b^k)(x_k - x_j) \leq 0, \quad j, k \in M, \, j \neq k. \tag{6a}$$

$$(p(S^k) - b^k)(x_k + x_j - 1) \leq 0, \quad j, k \in M, \, j \neq k, \tag{6b}$$

We now show that they are derived from (NLM) and are thus valid for the SMBPP.

Proposition 5. Constraints (6) are valid.

Proof. Consider two clients $j, k \in M$ with $j \neq k$. For client k, we multiply the corresponding Constraint (1b) by $(1 - x_j) \geq 0$ and Constraint (1c) by $-x_j \leq 0$ and sum up both resulting inequalities. We then obtain

$$(p(S^k) - b^k)x_k(1 - x_j) - (p(S^k) - b^k)(1 - x_k)x_j = (p(S^k) - b^k)(x_k - x_j) \leq 0,$$

which is exactly Constraint (6a) for clients j and k. Constraint (6b) is obtained similarly, by multiplying Constraint (1b) by $x_j \geq 0$ and Constraint (1c) by $-(1 - x_j) \leq 0$ and again summing up the resulting inequalities. □

In Constraints (6) appear explicitly the buying decisions of two clients $j \neq k \in M$. Although, redundant for (NLM), we can add these constraints to the linearized formulations to take advantage of this explicit link. To that end, let us extend the definition (4) of variables s_{ij} to all $i \in N$, $j \in M$.

With these additional variables s_{ik}, $i \notin S^k$, we can then linearize Constraints (6). Thus, we obtain the following formulation:

$$
\begin{aligned}
(\mathrm{LM}_3) \quad & \max_{p,x,s} && \sum_{j\in M}\sum_{i\in S^j} s_{ij} && (7a)\\
& \text{s.t.} && \sum_{i\in S^j} s_{ij} \le b^j x_j, \quad j \in M, && (7b)\\
& && s_{ij} \le p_i, \quad i \in N,\, j \in M, && (7c)\\
& && s_{ij} \le U_i x_j, \quad i \in N,\, j \in M, && (7d)\\
& && s_{ij} \ge p_i - U_i(1-x_j), \quad i \in N,\, j \in M, && (7e)\\
& && \sum_{i\in S^k} (s_{ik} - s_{ij}) \le b^k (x_k - x_j), \quad j,k \in M,\, j \ne k, && (7f)\\
& && \sum_{i\in S^k} (s_{ik} + s_{ij} - p_i) \le b^k (x_k + x_j - 1), \quad j,k \in M,\, j \ne k, && (7g)\\
& && s_{ij} \ge 0, \quad i \in N,\, j \in M, && (7h)\\
& && p_i \ge 0, \quad i \in N,\\
& && x_j \in \{0,1\}, \quad j \in M.
\end{aligned}
$$

This model is clearly an extension of (LM_2). We also investigated applying a complete RLT step to (NLM), which however does not lead to any new constraints for the linear formulations, and has therefore been omitted. Furthermore, Constraints (6) could have been linearized by introducing, in addition to the variables $r_j := p(S^j)x_j$ for $j \in M$ of (LM_1), variables $r_{kj} := p(S^k)x_j$ for $k, j \in M$ with $k \ne j$. However, it can be shown that the resulting formulation fails to exploit the additional information of (6) and thus results in the same linear relaxation bounds as (LM_1).

In Section 3.3, we already observed that Constraints (7d) are redundant for $i \in S^j$. This observation can be extended to $i \notin S^j$.

Proposition 6. Constraints (7d) are redundant in (LM_3).

Proof. For every optimal solution of (LM_3) in which we omit Constraints (7d) and that satisfies $s_{ij} > U_i x_j$, for $i \notin S^j$, we can build a new feasible point with the same objective function value by setting $s_{ij} = U_i x_j$ for those variables.

For a client j_0 and a product i_0 such that $i_0 \notin S^{j_0}$, if $s_{i_0 j_0} > U_{i_0} x_{j_0}$, we build a new solution with $\tilde{s}_{i_0 j_0} = U_{i_0} x_{j_0}$. We show that the constructed point is feasible. First, the left-hand side of (7c) and (7g) and thus the constraints remain satisfied. Given that $\tilde{s}_{i_0 j_0} = U_{i_0} \ge p_{i_0}$ when $x_{j_0} = 1$ and $\tilde{s}_{i_0 j_0} = 0 \ge p_{i_0} - U_{i_0}$ when $x_{j_0} = 0$, Constraints (7e) are also satisfied. For a fixed client $k \in M$, if $i_0 \notin S^k$, Constraint (7f) is trivially satisfied. Otherwise, $\tilde{s}_{i_0 j_0}$ satisfies

$$
\tilde{s}_{i_0 j_0} \ge b^k x_{j_0} \ge b^k x_{j_0} - \left(\sum_{i\in S^k} s_{ik} - b^k x_k \right) - \sum_{i \in S^k \setminus \{i_0\}} s_{i j_0},
$$

where the first inequality comes from the fact that $b^k \le U_{i_0}$ when $i_0 \in S^k$ and the second inequality is derived by (7b) and $s_{ik} \ge 0$. Then, Constraints (7f) are also satisfied. Finally, since $i_0 \notin S^{j_0}$, the objective function does not change and the result follows. □

The new variables s_{ij} for $i \notin S^j$ allow to create a relation to a client k for whom $i \in S^k$ by means of Constraints (7f) and (7g). The formulation (LM_3) thus explicitly links clients with intersecting bundles. However, (LM_3) contains $\mathcal{O}(nm)$ variables and $\mathcal{O}(m^2)$ constraints, as opposed to the $\mathcal{O}(n+m)$ variables and $\mathcal{O}(m)$ constraints of (LM_1) or the $\mathcal{O}(nm)$ variables and $\mathcal{O}(m)$ constraints of (LM_2). In particular, the size of (LM_3) grows quadratically in the number of clients, as opposed to a linear factor for the two other formulations.

To summarize, (LM_1) and (LM_2) are two different linear reformulations of the MINLP formulation (NLM), and it has been shown in Section 3.3 that $v(\mathrm{LM}_2) \le v(\mathrm{LM}_1)$. Finally, ($\mathrm{LM}_3$) is obtained from

(LM_2) by adding linearized valid inequalities, therefore improving on the linear relaxation bound of (LM_2). Thus, the linear relaxations of all three formulations satisfy $v(\mathrm{LM}_3) \leq v(\mathrm{LM}_2) \leq v(\mathrm{LM}_1)$.

4 Algorithmic Discussion

In this section, we present two approaches that allow to exploit (LM_1) and (LM_3). First, we discuss a Benders decomposition, generating valid cutting planes from (LM_3) that can be added to (LM_1) on-the-fly. Second, we present two families of heuristics using both formulations.

4.1 Benders Decomposition

Based on (LM_3), we now develop a Benders decomposition to find valid feasibility cuts to add to (LM_1). To be able to decompose the Benders sub-problems by clients, we first add to (LM_3) the variables r_j from (LM_1), as well as the coupling between variables r_j and s_{ij}

$$r_j = \sum_{i \in S^j} s_{ij}, \quad j \in M. \tag{8}$$

Note that (8) is already used in Section 3.3 to map solutions of (LM_2) to solutions of (LM_1).

The objective function of (LM_3) can then be rewritten as $\max \sum_{j \in M} r_j$. For fixed $\bar{x}, \bar{p}$ and $\bar{r}$, it remains to solve the feasibility problem given by Constraints (8), (7c), and (7e)–(7g) in the variables s_{ij}, which can now be decomposed by clients. Consequently, we obtain a Benders sub-problem for every client $j \in M$, denoted by (SP_j) having the following form:

$$
\begin{aligned}
(\mathrm{SP}_j) \quad \max \quad & 0 \\
& \sum_{i \in S^j} s_{ij} = \bar{r}_j, && && [\alpha_j] \\
& s_{ij} \leq \bar{p}_i, && i \in N, && [\beta_{ij}] \\
& s_{ij} \geq \bar{p}_i - U_i(1 - \bar{x}_j), && i \in N, && [\gamma_{ij}] \\
& \sum_{i \in S^k} (s_{ik} - s_{ij}) \leq b^k(\bar{x}_k - \bar{x}_j), && k \in M,\ j \neq k, && [\delta_{jk}] \\
& \sum_{i \in S^k} (s_{ik} + s_{ij}) \leq \sum_{i \in S^k} \bar{p}_i + b^k(\bar{x}_k + \bar{x}_j - 1), && k \in M,\ j \neq k, && [\epsilon_{jk}] \\
& s_{ij} \geq 0, && i \in N,
\end{aligned}
$$

where the dual variables corresponding to each group of constraints are shown in brackets. Its dual is given by

$$
\begin{aligned}
(\mathrm{DSP}_j) \quad \min_{(\alpha,\beta,\gamma,\delta,\epsilon)_j} \quad & \alpha_j \bar{r}_j + \sum_{i \in N} \beta_{ij} \bar{p}_i - \sum_{i \in N} \gamma_{ij} [\bar{p}_i - U_i(1 - \bar{x}_j)] \\
& - \sum_{k \in M:\, k \neq j} \delta_{kj} \left[\bar{r}_k - b^k(\bar{x}_k - \bar{x}_j)\right] \\
& + \sum_{k \in M:\, k \neq j} \epsilon_{kj} \left[b^k(\bar{x}_k + \bar{x}_j - 1) + \sum_{i \in S^k} \bar{p}_i - \bar{r}_k \right] \\
\text{s.t.} \quad & \alpha_j + \beta_{ij} - \gamma_{ij} - \sum_{\substack{k \in M:\, k \neq j \\ i \in S^k}} \delta_{kj} + \sum_{\substack{k \in M:\, k \neq j \\ i \in S^k}} \epsilon_{kj} \geq 0, \quad i \in S^j, \\
& \beta_{ij} - \gamma_{ij} - \sum_{\substack{k \in M:\, k \neq j \\ i \in S^k}} \delta_{kj} + \sum_{\substack{k \in M:\, k \neq j \\ i \in S^k}} \epsilon_{kj} \geq 0, \quad i \notin S^j, \\
& \alpha_j \in \mathbb{R},\ \beta_{ij}, \gamma_{ij}, \delta_{kj}, \epsilon_{kj} \geq 0, \quad i \in N,\ k \in M, j \neq k.
\end{aligned}
$$

Since (SP_j) is a feasibility problem and since (DSP_j) is always feasible ($\alpha = \beta = \gamma = \delta = \epsilon = 0$ is a feasible point), the optimal value $v(\text{DSP}_j)$ is either 0 or unbounded. In the latter case, given an extreme ray of (DSP_j), a feasibility cut can be derived. The master problem of the Benders reformulation is

$$
\begin{aligned}
\text{(MP)} \quad \max_{p,x,r} \quad & \sum_{j \in M} r_j \\
\text{s.t.} \quad & r_j \leq b^j x_j, \quad j \in M \\
& p_i \geq 0,\ r_j \geq 0,\ x_j \in \{0,1\}, \quad i \in N,\ j \in M, \\
& \alpha_j r_j + \sum_{i \in N} \beta_{ij} p_i - \sum_{i \in N} \gamma_{ij} [p_i - U_i(1 - x_j)] \\
& - \sum_{k \in M:\, k \neq j} \delta_{kj} [r_k - b^k (x_k - x_j)] \\
& + \sum_{k \in M:\, k \neq j} \epsilon_{kj} \left[b^k (x_k + x_j - 1) + \sum_{i \in S^k} p_i - r_k \right] \geq 0, \\
& (\alpha, \beta, \gamma, \delta, \epsilon)_j \text{ extreme ray of } (\text{DSP}_j),\ j \in M. \qquad (9)
\end{aligned}
$$

Constraints (9) can be generated using the classical Benders approach. The relaxed master problem, denoted (RMP) and obtained by dropping most of Constraints (9), is solved to retrieve optimal values $\bar{x}$, $\bar{p}$, and $\bar{r}$ and yields an upper bound for the original problem. The incumbent is introduced into each (DSP_j). If (DSP_j) is unbounded, a cut corresponding to an extreme ray is added to (RMP).

However, here our goal is to tighten the linear relaxation of (LM_1), which is the fastest formulation, but has weak LP bounds (supported by computational results in Section 5.2). We use cuts generated by applying the Benders decomposition to the tighter formulation (LM_3) and strengthen the linear relaxation of (LM_1) iteratively. This procedure is summarized in Algorithm 1.

Algorithm 1 Adding Benders feasibility cuts from (LM_3) to (LM_1)

```
Require: N, M, S, b
  set exists = True
  while exists do
    solve linear relaxation of (LM_1)
    retrieve x̄, p̄ and r̄
    set exists = False
    for j ∈ M do
      solve (DSP_j)
      if (DSP_j) unbounded then
        add feasibility cut (9) to (LM_1)
        set exists = True
      end if
    end for
  end while
```

Observe that at the end of Algorithm 1, all violated feasibility cuts from the Benders decomposition of (LM_3) have been added to (LM_1). In particular, the optimal solution $\bar{x}$, $\bar{p}$ and $\bar{r}$ of the final linear relaxation has value $v(\text{LM}_3)$.

4.2 LP-based Heuristics

In this section, we discuss two heuristics based on the LP relaxation of LM_1 and LM_3. As we will see in Section 5.2, (LM_1) is the lightest and fastest model in terms of computation time, whereas (LM_3) produces the best linear relaxation, making both formulations the ideal starting points for the heuristics.

Whenever client buying decisions x are fixed, we can determine the corresponding optimal prices by solving an LP. In particular, we can easily determine prices corresponding to the optimal selling strategy. It could therefore be useful to partition clients into two sets, i.e., those clients that purchase and those that do not, and then optimize the prices corresponding to that fixed x. To obtain this partition, we use information of the linear relaxation to fix client buying decisions and then optimize prices to maximize the total revenue.

The first heuristic, denoted by $\mathsf{HS}l$, begins by solving the linear relaxation of (LM_l) for $l = 1, 3$, retrieving an optimal fractional solution x. For a fixed sequence of thresholds $T \subseteq [0, 1]$, the heuristic computes the partition of clients $M_1 := \{j \in M : x_j \geq t\}$ and $M_0 := M \setminus M_1$ for all $t \in T$. Then, fixing $x_j = 1$ for $j \in M_1$ and $x_j = 0$ for $j \in M_0$, the corresponding prices are optimized by solving

$$\text{(Pricing)} \quad \max_{p \geq 0} \sum_{j \in M_1} p(S^j) \quad \text{s.t.} \quad p(S^j) \leq b^j, \; j \in M_1.$$

Given the optimal prices, the buying decisions are updated so that all clients purchase when their budget allows it. Finally, the solution with highest revenue among all the thresholds is returned. The pseudo-code of the heuristic is given in Algorithm 2.

Algorithm 2 Heuristic $\mathsf{HS}l$

Require: $N, M, S, b, T = \{t_1, t_2, \ldots t_{|T|}\} \subseteq [0, 1]$
  solve linear relaxation of (LM_l), retrieve x
  for $t \in T$ **do**
    set $M_1 = \{j \in M : x_j \geq t\}$
    solve Pricing(M_1), retrieve p^*
    for $j \in M$ **do**
      if $p^*(S^j) \leq b^j$ **then**
        set $x_j^t = 1$
      else
        set $x_j^t = 0$
      end if
    end for
  end for
  set $x^* = \arg\max_t\{\sum_{j \in M} p^*(S^j) x_j^t\}$
  return $x^*, p^*, v^* = \sum_{j \in M} p^*(S^j) x_j^*$

The second heuristic, denoted by $\mathsf{HR}l$, uses again the linear relaxation of (LM_l) to partition the clients. It, however, then turns to solving a reduced MILP obtained from the lightest formulation (LM_1) to determine the remaining decision variables. After retrieving the optimal fractional x of the linear relaxation of (LM_l), clients are partitioned into three sets M_0, M_f and M_1. More precisely, we define $M(x)$ as the sequence of ordered non-decreasingly with respect to their x value, i.e.,

$$M(x) := \{j_1, \ldots, j_m : x_{j_1} \leq \ldots \leq x_{j_m}\}, \tag{10}$$

and we let

$$M_0 := \left\{j_k \in M(x) : k \leq \frac{m}{3}\right\}, \quad M_f := \left\{j_k \in M(x) : \frac{m}{3} < k \leq \frac{2m}{3}\right\}, \quad M_1 = M \setminus (M_0 \cup M_f). \tag{11}$$

Next, we fix $x_j = 1$ for $j \in M_1$ and $x_j = 0$ for $j \in M_0$, while the remainder is optimized by (LM_1) after fixing variables. In particular, the upper bound on prices can now be tightened by setting

$$\tilde{U}(p_i) := \begin{cases} \min\{b_j : j \in M_1\}, & \text{if } j \in M_1,\, i \in S^j, \\ U(p_i), & \text{otherwise,} \end{cases} \qquad \tilde{U}(S^j) := \sum_{i \in S^j} \tilde{U}(p_i), \; j \in M. \tag{12}$$

The reduced MILP for (LM_1) is denoted by (LM_1-H) and reads

$$
\begin{aligned}
(\text{LM}_1\text{-H}) \quad \max_{p,x,r} \quad & \sum_{j\in M_1} p(S^j) + \sum_{j\in M_f} r_j \\
\text{s.t.} \quad & r_j \le \min\{b^j, \tilde{U}(S^j)\}x_j, \quad j \in M_f, \\
& r_j \ge p(S^j) - \tilde{U}(S^j)(1-x_j), \quad j \in M_f, \\
& p(S^j) \le b^j, \quad j \in M_1, \\
& r_j \ge 0,\ p_i \ge 0,\ x_j \in \{0,1\}, \quad j \in M_f.
\end{aligned}
$$

Note that the McCormick inequality might not be redundant after tightening the upper bounds on the prices, which justifies the first constraint of (LM_1-H). Finally, the client buying decisions are recomputed after retrieving the optimal prices. The pseudo-code for HRl is given in Algorithm 3.

Algorithm 3 Heuristic HRl

```
Require: N, M, S, b
  solve linear relaxation of (LM_l), retrieve x
  construct M(x), M_0, M_f, M_1 by (10) and (11)
  update bounds Ũ(S^j) by (12)
  solve (LM_1-H), retrieve p*
  for j ∈ M do
    if p*(S^j) ≤ b^j then
      set x*_j = 1
    else
      set x*_j = 0
    end if
  end for
  return x*, p*, v* = Σ_{j∈M} p*(S^j) x*_j
```

5 Computational Study

In this section, we report the computational results of our experiments. First, we discuss how our instances were generated. We then discuss the performance of the formulations of Section 3, the Benders decomposition of Section 4.1 and finally the heuristics of Section 4.2. All experiments are implemented using the Python interface of CPLEX 12.9 MILP solver and carried out on a machine with an Intel i7 quad-core processor at 4.00 GHz and 32 GB of memory. CPLEX was run in single-threaded mode and all CPLEX parameters were set to their default values.

5.1 Instances

A SMBPP instance of consists n products and m clients. Each client is encoded by a positive real representing his budget and a subset of products representing his bundle. First of all, we define the density of the instance matrix S by

$$d := \frac{\sum_{i\in N}\sum_{j\in M} S_i^j}{nm}.$$

Then, we can discuss how the formulation sizes, in terms of number of variables and constraints, evolve with the instance size. In particular, the size of (LM_2) depends on the density d, whereas the sizes of (LM_1) and (LM_3) don't. The different formulation sizes are summarized in Table 1.

To test the performance of our formulations, as well as the stability of the Benders decomposition, we randomly generated families with different parameters and containing 10 instances each. We considered instances with $m \in \{25, 50, 100, 150\}$ clients, $n \in \{25, 50, 75\}$ products and density $d \in \{0.1, 0.2, 0.4\}$. We denote these instances families by $\text{U}(n, m, d)$ and will refer to them by *uniform instances*. The client bundles are generated as follows: for client $j \in M$ and product $i \in N$, we set $S_i^j = 1$ independently with

Formulation	Variables	Constraints
(LM_1)	$n + 2m$	$3m$
(LM_2)	$n + m + dnm$	$m + 2dnm$
(LM_3)	$n + m + nm$	$m + 2nm + 2m(m - 1)$

Table 1: Number of variables and constraints for each formulation.

probability d. The corresponding budgets are drawn uniformly in $[1, 1000]$. If an empty bundle S^j or unused product i is detected, we sample a random product i or client j, respectively, and set $S^j_i = 1$.

Furthermore, to measure the effect of budget disparsity, we generated instances with rich and poor clients. We consider fixed parameters $m = 100$, $n = 25$ and $d = 0.2$ and the matrices S are generated as previously. However, for $(m_1, m_2) \in \{(25, 75), (50, 50), (75, 25)\}$, budgets are drawn uniformly in $[1, 500]$ for m_1 clients and in $[1000, 5000]$ for m_2 clients. We denote these families by $\mathrm{RP}(m_1, m_2)$ and call them *rich-poor instances* in the following.

All instance data can be downloaded at https://github.com/vbucarey/smbpp.

5.2 Performance of (LM_1), (LM_2) and (LM_3).

First, we compared the quality of the three MILP formulations. Thus, we ran all the uniform instances with a time limit of 1 hour, while keeping default CPLEX settings. Table 2 shows the percentages of instances solved to optimality, grouped by density, whereas Table 3 shows the number of instances solved for different parameter combinations. In total, (LM_1) can solve 64.4% of the considered instances to optimality, while (LM_2) and (LM_3) solve only 55.8% and 44.9%, respectively.

Density	0.1	0.2	0.4	All
LM_1	75.3	62.7	55.3	64.4
LM_2	69.3	52.0	46.0	55.8
LM_3	53.3	42.0	39.3	44.9

Table 2: Percentage of solved instances within a time limit of 1 hour.

		d =0.1			d =0.2			d =0.4		
m	n	LM_1	LM_2	LM_3	LM_1	LM_2	LM_3	LM_1	LM_2	LM_3
25	25	10	10	10	10	10	10	10	10	10
25	50	10	10	10	10	10	10	10	10	10
25	75	10	10	10	10	10	10	10	10	10
50	25	10	10	10	10	10	10	10	10	10
50	50	10	10	10	10	10	10	10	10	10
50	75	10	10	10	10	10	9	10	10	7
75	25	10	10	10	10	10	4	10	9	2
75	50	10	10	2	9	5	0	7	0	0
75	75	10	10	0	6	0	0	3	0	0
100	25	10	10	8	9	3	0	3	0	0
100	50	9	4	0	0	0	0	0	0	0
100	75	0	0	0	0	0	0	0	0	0
150	25	4	0	0	0	0	0	0	0	0
150	50	0	0	0	0	0	0	0	0	0
150	75	0	0	0	0	0	0	0	0	0

Table 3: Number of solved instances within a time limit of 1 hour.

Observe that the number of solved instances decreases with higher density for all three formulations. This trend is partially linked to weak upper bounds $U(p_i)$ and $U(S^j)$ for products $i \in N$ and clients $j \in M$ used in the linearizations. The higher the density, the larger these upper bounds must be to be valid. Consequently, the linear relaxation of all models produce weaker bounds, as they involve big-M constraints using these bounds. Furthermore, for low density, the SMBPP has an inherent decomposable structure, which can be exploited by the solver. It also becomes apparent that for higher problem sizes, none of the formulations are able to solve the instances within the time limit of 1 hour. (LM_1) is the most scalable formulation and can still solve instances with 150 clients, 25 products and a density of 0.1. On the other hand, for the same density of 0.1 formulations (LM_2) and (LM_3) cannot solve any instances with more than 100 clients. This difficulty can mainly be explained by the larger model size of the latter two formulations, as presented in Table 1. Even though (LM_3) produces strong linear relaxation bounds, these better bounds do not benefit in terms of solution time, due to the large size of the model in terms of variables and constraints.

This observation is also supported by Figure 2. For instances that are optimally solved by all three formulations, it shows the cumulative distributions of running time, time spent for the LP relaxation, LP gap and branch-and-bound nodes. Here, LP gap is the relative gap between the value of the LP relaxation v^{LP} and the optimum v^*, i.e.,

$$\text{LP gap} := \frac{v^{\mathrm{LP}} - v^*}{v^*}.$$

We observe that (LM_1) solves the fastest, both with respect to the total running time and the time spent

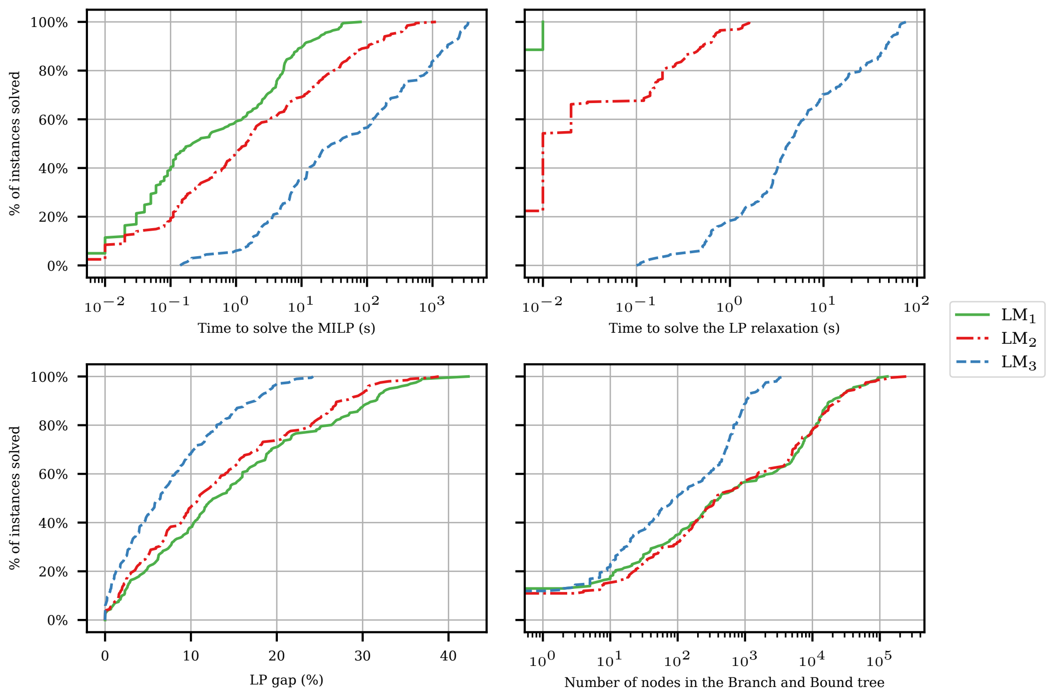

Figure 2: Cumulative distributions of different measures for instances solved by all three formulations.

to solve the LP relaxation. In contrast, (LM_3) is slowest although it needs to solves the smallest number of branch-and-bound nodes. It has the tightest LP relaxation, thus justifying the smaller number of nodes. However, due to its large size, it takes a long time to solve every node, i.e., every linear relaxation. Finally, (LM_2) produces only a slightly better LP relaxation compared to (LM_1), while the number of branching nodes remains comparable. With regard to its time performance it can be situated in between (LM_1) and (LM_3).

To further understand these results, we also compared the LP gap and the root gap of all three formulations with respect to density, product and client numbers. Similarly to the LP gap, we define the root gap as the relative gap between the value after root processing by CPLEX and the optimal solution. As a base case, we use $d = 0.2$, $n = 50$ and $m = 50$ and only one parameter at a time is varied. The results of this experiment averaged over the corresponding uniform instances are represented in Figure 3. First of all, we observe that for increasing density both gaps increase for all formulations. This observation is closely linked to the evolution of the upper bounds on the prices, discussed previously.

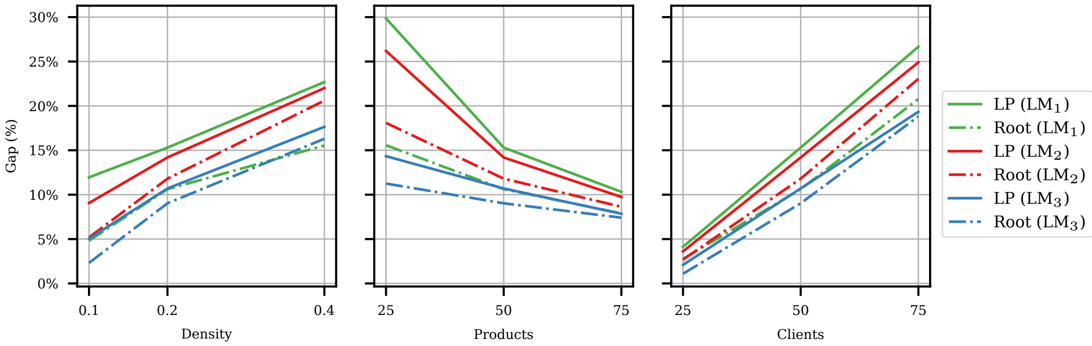

Figure 3: LP and root gaps for instances with $n = m = 50$ and $d = 0.2$.

Similarly, when increasing the number of products while having constant density and number of clients, every individual product appears in fewer bundles. As a consequence, we can set tighter bounds on the prices and therefore obtain tighter gaps. An analogous interpretation applies when decreasing the number of clients.

On the other hand, observe the effect of CPLEX root processing on the different formulations. The solver can significantly improve the gap of (LM_1), in some cases outperforming the linear relaxation of the tighter (LM_3). For the instances $\mathrm{U}(50, 50, 0.4)$, the average root gap for (LM_1) is even better than for (LM_3). In fact, we observed that CPLEX finds already good quality feasible points at the root node using its default heuristics on (LM_1). The main difficulty therefore mostly lies in improving the dual bound in the branch-and-bound tree.

Finally, we ran all formulations on the rich-poor instances and summarize the results in Table 4. A general observation is that more instances are solved than in the corresponding uniform instance family. (LM_1) solves all the rich-poor instances. Even (LM_3) can solve additional instances, especially when there are a lot of low-budget clients. If there are a lot of low budgets, the upper bounds used in the linearization can be chosen tighter, thus resulting in tighter LP relaxations. Furthermore, due to the structure of the SMBPP, some buying decisions can be easily deduced from each other. Consider, for instance, that low-budget clients decide to buy, then rich clients interested in same products can also purchase. This simple observation reduces the number of different combinations of buying decisions and thus the nodes to be explored in the branch-and-bound tree.

	(LM_1)	(LM_2)	(LM_3)
$\mathrm{RP}(25, 75)$	10	7	1
$\mathrm{RP}(50, 50)$	10	10	5
$\mathrm{RP}(75, 25)$	10	10	8
$\mathrm{U}(25, 100, 0.2)$	9	3	0

	(LM_1)	(LM_2)	(LM_3)
$\mathrm{RP}(25, 75)$	39.68	36.76	21.31
$\mathrm{RP}(50, 50)$	38.46	34.29	18.48
$\mathrm{RP}(75, 25)$	39.25	32.46	17.97
$\mathrm{U}(25, 100, 0.2)$	40.76	38.20	21.71

Table 4: Rich and poor experiment with corresponding uniform instance family for reference. Left: Number of instances solved. Right: Average LP gap (%).

5.3 Benders Decomposition Performance

We now discuss the performance of our Benders implementation. Cutting procedures such as the Benders decomposition of Section 4.1 are prone to instability, whenever a lot of cuts need to be separated. In a preliminary experiment, we compare in-out stabilization as proposed by Ben-Ameur and Neto 2007 with different parameters and an improved cut generation sub-problem discussed by Conforti and Wolsey 2018. For a given instance, Figure 4 shows the evolution of the upper bound over the Benders iterations, since only feasibility cuts are separated.

Traditional represents a Benders implementation without stabilization, In-Out α use in-out stabilization for different parameter values and CW is the method of Conforti and Wolsey 2018. The authors of the latter describe the utility of their procedure especially when only separating feasibility cuts, which is our case. Figure 4 confirmed this claim in our preliminary test and we also observed a drastic decrease in the

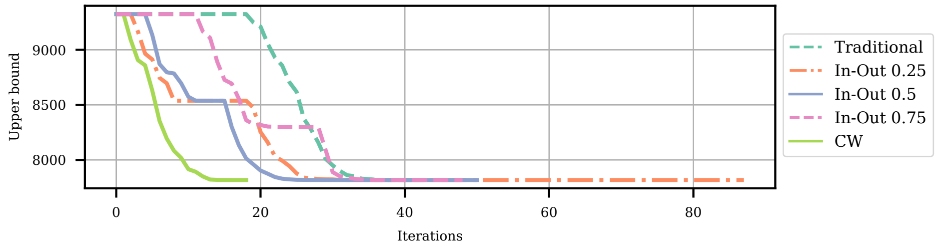

Figure 4: Upper bound over Benders iterations for different stabilization techniques.

number of generated cuts. Since it is the most efficient stabilization method in our case, we will only discuss this implementation. More precisely, we solve (LM_1) as a master problem and separate cuts by solving a separation problem for every client $j \in M$ using the method of Conforti and Wolsey 2018.

In another preliminary experiment, we compared our implementation against the automatic Benders in CPLEX in two versions: the full version (CPLEX decides the decomposition to apply) and the annotated one (the user provides a valid decomposition). Both of them were outperformed. Furthermore, since generating cuts further down in the branch-and-bound tree did not prove beneficial in our preliminary studies, we limit our analysis to a cut-and-branch algorithm using our implementation. We further limit the cut separation at the root node to k rounds over all clients $j \in M$. We consider $k \in \{1, 5, 10\}$ and denote the corresponding algorithm by CnB(k). The described framework is implemented using CPLEX callbacks, where we unregister the callback after the root node and finish the resolution using default CPLEX parameters. Afterwards we unregister the callback and keep the model object in order to take advantage of the warm-start and pre-processing of CPLEX. In this experiment, we only consider larger instances with $m \in \{100, 150\}$ clients with a time limit of 1 hour.

Density	0.1	0.2	0.4	All
(LM_1)	35.00	13.33	6.66	18.33
CnB(1)	36.66	15.00	6.66	19.44
CnB(5)	36.66	13.33	6.66	18.88
CnB(10)	36.66	15.00	6.66	19.44

Table 5: Percentage of solved instances within a time limit of 1 hour.

Table 5 presents the percentage of solved instances within the time limit of 1 hour for (LM_1) and CnB(k). We observe that neither variant of our Benders implementation can significantly scale up the number of solved instances compared to (LM_1). More precisely, CnB(1) and CnB(10) could solve 2 more instances, whereas CnB(5) could even only 1 additional instance. As discussed in Section 4.1, a full addition of Benders cuts results in a root relaxation equivalent to (LM_3). However, Figure 3 also showed that (LM_1) is significantly tightened by CPLEX at the root, resulting eventually in a tighter root gap for larger instances compared to (LM_3). As a consequence, our Benders implementations are not more efficient than (LM_1), which runs fast and has the smallest number of variables and constraints.

5.4 Heuristics Performance

As a last experiment, we assess the performance of the heuristics presented in Section 4.2. In Section 5.2, we already mentioned that CPLEX finds good quality feasible points using its default heuristics applied to (LM_1). Thus, we only study our heuristics as standalone methods to quickly find feasible points close to the best known solution produced by CPLEX.

We begin our analysis with a few example runs of HSl for $l = 1, 3$. We aim to understand the behavior of the heuristics in terms of the sequence of thresholds. To do so, we select 9 instances from $U(n, m, d)$ with $n = 25$ products and variable size of the set of clients m and bundle density d. Figure 5 shows the values of both heuristics for different threshold values and for different uniform instances. Here, Best indicates the value of the best-known solution value found by any of our formulations after the time limit of 1 hour. It is represented as a solid line for instances solved to optimality, otherwise it is represented as a dotted one. In the majority of tested instances, HS3 outperforms HS1, in the sense that it finds a

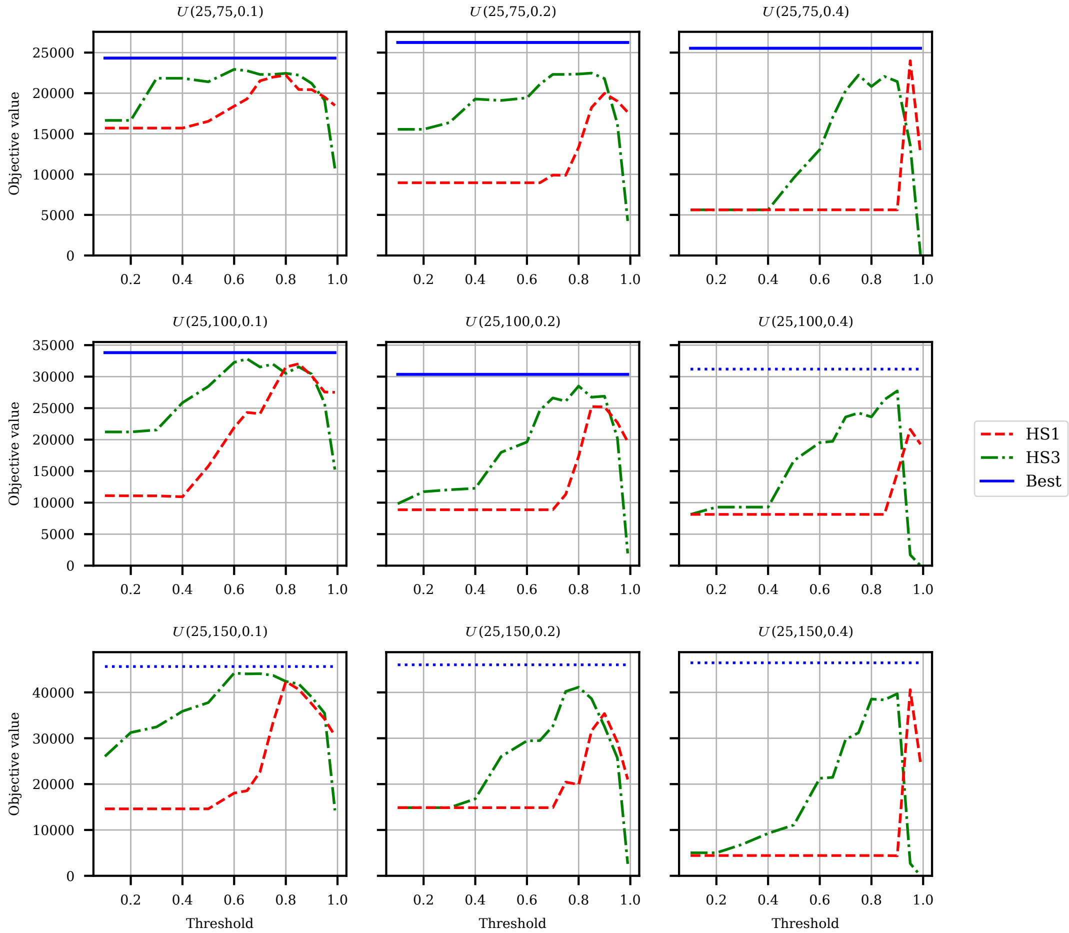

Figure 5: Impact of the threshold on the solution quality HSl for $l = 1, 3$.

better overall solution. We also note that only in 2 cases HS1 found a better solution. While HS1 finds good solutions only for thresholds close to 1, HS3 finds good solutions for smaller values already. In other words, HS3 is more robust in the sense that it finds good solutions for a wider range of threshold values than HS1. Thus, it would be possible to examine a wider grid of threshold and still find good solutions.

In a second step, we aim to measure the compromise between solution quality and solution time for both heuristics. We define two threshold sequences

$$T_0 = \{0, 0.1, 0.2, 0.3, \ldots, 0.9, 0.99\}$$
$$T_1 = \{0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, \ldots, 0.95, 0.99\} \supset T_0.$$

and let HSlk be the heuristic HSl for $l = 1, 3$ using the sequence T_k for $k = 0, 1$. We furthermore define the gap of a heuristic as the relative error to the best-known solution after 1 hour. More precisely, if we let v^{H} be the value returned by a heuristic and v^{Best} the best-known solution value, then

$$\mathrm{Gap} := \frac{v^{\mathrm{Best}} - v^{\mathrm{H}}}{v^{\mathrm{Best}}}.$$

Note that for instances where the best-known solution is not optimal, the gap of a heuristic could be negative, i.e., the heuristic finds a better solution than any of our formulations after 1 hour. However, the latter never occurred in our experiments, since at best our heuristics returned the best-known solution values. We ran all heuristics over the uniform instances. The average gaps and running times are given in Table 6.

Heuristic	HR1	HR3	HS10	HS11	HS30	HS31
Gap (%)	18.35	16.33	35.86	21.32	12.99	10.24
Time (s)	9.93	34.74	0.16	0.25	16.42	16.45

Table 6: Average gap and time of the heuristics

Note that for exact methods the best average running time was 1447.282 seconds, while the heuristics run in less than a minute. HS3k produce the best solutions on average, while HS1k admit solutions furthest away from the best-known solution value. There is however a notable improvement when considering the thinner grid of thresholds, i.e., HSl1 for $l = 1, 3$. Solutions returned by heuristics HRl are of better quality than HS10 and HS11 but are dominated in quality by heuristics HS30 and HS31. On the other hand, with respect to running time the methods making use of (LM_3) are slower than their (LM_1) counter-parts. This trend is mainly related to the high time necessary to solve the linear relaxation of (LM_3), as we have observed previously in Figure 2. Furthermore, HRl might have a higher running time for some instances, since they still need to solve a reduced MILP, as opposed to only linear relaxations in HSlk. On average, HR3 is the most time-consuming heuristic, while not producing exceptionally good solutions. More visually, Figure 6 shows the cumulative distribution of both gap and running time of the heuristics over all uniform instances. We observe that HS31 finds the best quality solutions at the cost of a slightly slower running time. HR3, however, has roughly the same running time, but its gap is notably worse. Finally, HR1 presents a good compromise between running time and solution quality, with overall gaps below 50%.

6 Concluding Remarks

In this paper, we presented three novel MILP formulations for the single-minded bundle pricing problem. After obtaining a mixed-integer non-linear model, we derived models (LM_1) and (LM_2) by different linearizations. A stronger model, (LM_3), was obtained by adding RLT-like valid inequalities, that significantly strengthen the gap of the LP relaxation.

Our results showed that the main bottleneck is solving the tighter but significantly heavier LP relaxation of (LM_3) in large instances. In a first attempt to circumvent this bottleneck, we discussed a Benders decomposition with a strengthened cut generation LP using the results of Conforti and Wolsey 2018. The idea was to exploit the light model (LM_1) and tighten its LP relaxation, using Benders cut from (LM_3). However, we could not observe the desired scale up in our computational experiments.

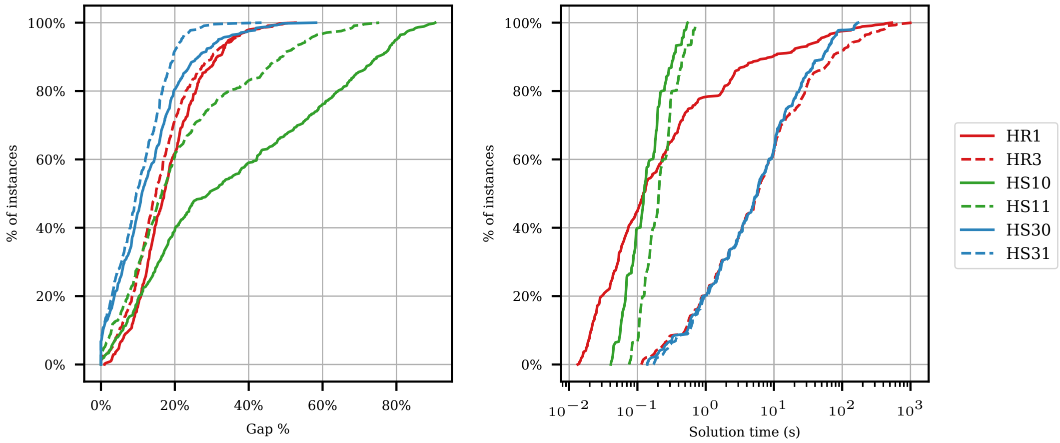

Figure 6: Cumulative distributions for HRl and HSlk for $l = 1, 3$ and $k = 0, 1$: Left: Gap (%) to the best-known solution. Right: Solution time (s).

Finally, in an attempt to exploit the tighter linear relaxation of (LM$_3$), we presented two LP-based heuristics to find good solutions fast. Here, we could obtain significantly better results when using (LM$_3$) to obtain a first fractional solution.

There are two possible ways to improve the results presented in this paper. The main bottleneck is still the large size of (LM$_3$) resulting from the linearization of the RLT-like valid inequalities. One possible improvement is to find different valid inequalities that link clients' buying decisions and that do not require additional linearizations. Secondly, the Benders decomposition could be sped up if there were a combinatorial algorithm to generate cuts, whereas we had to solve a cut generation LP for every client. On the other hand, the computational study of the SMBPP could still be extended. Another possible experiment is to consider the influence of the average budgets, i.e., budget divided by bundle size, on our formulations and resolution approaches.

Acknowledgments

Victor Bucarey and Martine Labbé have been partially supported by the Fonds de la Recherche Scientifique - FNRS under Grant(s) no PDR T0098.18. Fränk Plein has been supported by his F.R.S.-FNRS research fellowship. We would also like to thank Domenico Salvagnin for his valuable comments and discussion about this work.

References

Aydin, Goker and Jennifer K Ryan (2000). "Product line selection and pricing under the multinomial logit choice model." In: *Proceedings of the 2000 MSOM conference*. Citeseer.

Balcan, Maria-Florina and Avrim Blum (2006). "Approximation algorithms and online mechanisms for item pricing." en. In: *Proceedings of the 7th ACM conference on Electronic commerce - EC '06*. Ann Arbor, Michigan, USA: ACM Press, pp. 29–35. DOI: 10.1145/1134707.1134711. URL: http://portal.acm.org/citation.cfm?doid=1134707.1134711.

Ben-Ameur, Walid and José Neto (2007). "Acceleration of cutting-plane and column generation algorithms: Applications to network design." In: *Networks: An International Journal* 49.1, pp. 3–17.

Briest, Patrick and Piotr Krysta (2006). "Single-minded unlimited supply pricing on sparse instances." In: *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*. Society for Industrial and Applied Mathematics, pp. 1093–1102.

Calvete, Herminia I, Concepción Domínguez, Carmen Galé, Martine Labbé, and Alfredo Marin (2019). "The Rank Pricing Problem: models and branch-and-cut algorithms." In: *Computers & operations research* 105, pp. 12–31.

Conforti, Michele and Laurence A Wolsey (2018). “Facet separation with one linear program.” In: *Mathematical Programming*, pp. 1–20.
Fiat, Amos and Amiram Wingarten (2009). “Envy, Multi Envy, and Revenue Maximization.” In: *Internet and Network Economics*. Ed. by Stefano Leonardi. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 498–504. DOI: 10.1007/978-3-642-10841-9_48.
Grigoriev, Alexander, Joyce van Loon, Maxim Sviridenko, Marc Uetz, and Tjark Vredeveld (2008). “Optimal bundle pricing with monotonicity constraint.” In: *Operations research letters* 36.5, pp. 609–614.
Guruswami, Venkatesan, Jason D Hartline, Anna R Karlin, David Kempe, Claire Kenyon, and Frank McSherry (2005). “On profit-maximizing envy-free pricing.” In: *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, pp. 1164–1173.
Hanson, Ward and Kipp Martin (1996). “Optimizing multinomial logit profit functions.” In: *Management Science* 42.7, pp. 992–1003.
Khandekar, Rohit, Tracy Kimbrel, Konstantin Makarychev, and Maxim Sviridenko (2009). “On hardness of pricing items for single-minded bidders.” In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Springer, pp. 202–216.
Ledyard, John O (2007). “Optimal combinatoric auctions with single-minded bidders.” In: *Proceedings of the 8th ACM conference on Electronic commerce*. ACM, pp. 237–242.
Mayer, Stefan and Jochen Gönsch (2012). “Consumer choice modelling in product line pricing: reservation prices and discrete choice theory.” In: *Operations Research Proceedings 2011*. Springer, pp. 547–552.
McCormick, Garth P (1976). “Computability of global solutions to factorable nonconvex programs: Part I—Convex underestimating problems.” In: *Mathematical programming* 10.1, pp. 147–175.
Pierson, Margaret P, Gad Allon, and Awi Federgruen (2013). “Price Competition Under Mixed Multinomial Logit Demand Functions.” In: *Management Science* 59, p. 8.
Plein, Fränk (2017). “Analysis of a Problem in Product Pricing, Mathematics.” en. MA thesis. Brussels: Université Libre de Bruxelles. URL: http://hdl.handle.net/2013/ULB-DIPOT:oai:dipot.ulb.ac.be:2013/260879 (visited on 01/07/2018).
Rusmevichientong, Paat, Benjamin Van Roy, and Peter W Glynn (2006). “A nonparametric approach to multiproduct pricing.” In: *Operations Research* 54.1, pp. 82–98.
Sherali, Hanif D. and Warren P. Adams (2013). “Reformulation–Linearization Techniques for Discrete Optimization Problems.” In: *Handbook of Combinatorial Optimization*. Ed. by Panos M. Pardalos, Ding-Zhu Du, and Ronald L. Graham. New York, NY: Springer New York, pp. 2849–2896. DOI: 10.1007/978-1-4419-7997-1_45. URL: https://doi.org/10.1007/978-1-4419-7997-1_45.
Talluri, Kalyan T and Garrett J Van Ryzin (2006). *The theory and practice of revenue management*. Vol. 68. Springer Science & Business Media.
Van Loon, Joyce (2009). “Algorithmic Pricing.” PhD thesis. Universitaire Pers Maastricht.
Zhang, Yong, Francis YL Chin, Sheung-Hung Poon, Hing-Fung Ting, Dachuan Xu, and Dongxiao Yu (2020). “Offline and online algorithms for single-minded selling problem.” In: *Theoretical Computer Science*.