



**HAL**  
open science

# A Mixed-Integer Optimization Formulation for Buyers Formation

Sebastián Dávila-Gálvez, Martine Labbé, Vladimir Marianov, Fernando Ordoñez, Frédéric Semet

► **To cite this version:**

Sebastián Dávila-Gálvez, Martine Labbé, Vladimir Marianov, Fernando Ordoñez, Frédéric Semet. A Mixed-Integer Optimization Formulation for Buyers Formation. 2023. hal-04347339

**HAL Id: hal-04347339**

<https://inria.hal.science/hal-04347339v1>

Preprint submitted on 15 Dec 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# **Title: A Mixed-Integer Optimization Formulation for Buyers Formation**

## **Authors**

### **Sebastián Dávila-Gálvez (Corresponding author),**

Universidad de Santiago de Chile (USACH), Industrial Engineering Department, Laboratory for the Development of Sustainable Production Systems (LDSPS), Chile  
Av. Víctor Jara 3769, 9170124 Estación Central, Región Metropolitana  
sebastian.davila@usach.cl

### **Martine Labbé,**

Université d'Etat de Bruxelles, Department Libre de Bruxelles, Belgium  
Av. Franklin Roosevelt 50, 1050 Bruxelles, Belgium  
martine.labbe@ulb.be

### **Vladimir Marianov,**

Department of Electrical Engineering, Pontificia Universidad Católica de Chile and Instituto Sistemas Complejos de Ingeniería (ISCI), Chile  
Vicuña Mackenna 4860, Macul, Región Metropolitana  
marianov@ing.puc.cl

### **Fernando Ordónñez,**

Department of Industrial Engineering, Universidad de Chile, Chile  
Av. Beauchef 851, Santiago, Región Metropolitana  
fordon@dii.uchile.cl

### **Frédéric Semet,**

Univ. Lille, CNRS, Centrale Lille, Inria, UMR 9189 - CRISTAL, Lille, France  
Cité Scientifique, 59650 Villeneuve-d'Ascq, France  
frederic.semet@centralelille.fr

# A Mixed-Integer Optimization Formulation for Buyers Formation

Sebastián Dávila<sup>a</sup>, Martine Labbé<sup>c</sup>, Vladimir Marianov<sup>d</sup>, Fernando Ordóñez<sup>b</sup>, Frédéric Semet<sup>e</sup>

<sup>a</sup>Universidad de Santiago de Chile (USACH), Industrial Engineering Department, Laboratory for the Development of Sustainable Production Systems (LDSPS), Chile

<sup>b</sup>Department of Industrial Engineering, Universidad de Chile, Chile

<sup>c</sup>Université Libre de Bruxelles, Belgium

<sup>d</sup>Department of Electrical Engineering, Pontificia Universidad Católica de Chile and Instituto Sistemas Complejos de Ingeniería (ISCI), Chile

<sup>e</sup>Univ. Lille, CNRS, Centrale Lille, Inria, UMR 9189 - CRISTAL, Lille, France

---

## Abstract

Companies frequently offer wholesale prices for their products that decrease with the number of purchased items. However, single buyers may not be willing or able to purchase large quantities of a single item. Nevertheless, consumers can form groups to purchase at wholesale prices, obtaining bargaining power. This practice can be extended from single products to bundles. We propose a mathematical model to create groups of buyers wishing to purchase product bundles optimally. Mixed-integer programming formulations are presented for the cases of non-increasing price and step price functions. A Benders decomposition formulation is proposed for step price functions to solve large instances. Computational experiments show the performance of the method for synthetic instances.

*Keywords:* Coalition formation, MIP, Benders decomposition

---

## 1. Introduction

Digital marketplaces or e-marketplaces facilitate the interaction between purchasers and merchants worldwide. These systems enable merchants to provide goods or services to buyers without geographical restrictions, increasing demand. Buyers, in turn, have access to a wider variety of offers, allowing them to make better deals. The potential for increased commerce facilitated by digital marketplaces creates the opportunity to further stimulate commerce by simplifying the wholesale of products. Wholesale pricing refers to a discount on the unit price of a product when it is purchased in a large volume in a single [17]. Wholesale pricing has several benefits for the merchant or seller, such as lower marketing costs than retail sales, decreased need for logistics coordination, and increased market share. Although wholesale pricing offers the products at better prices for buyers, buyers are frequently unable or unwilling to purchase large volumes of products, reducing their bargaining power. To overcome this limitation, the buyer can attempt to form groups or clusters of customers wishing to buy the same product to take advantage of the wholesale price. In this situation, the price paid per unit depends on the number of buyers in a cluster.

Technology has made it easier to coordinate buyers with platforms that offer to bring them together to buy specific products, such as LivingSocial or Groupon [23]. Generally speaking, on these platforms, a coordinator negotiates wholesale prices with providers and collects information from buyers to obtain their reservation prices, preferences

*Preprint submitted to OMEGA*

*January 6, 2023*

for products, and the number of items they would be willing to purchase. With these data, it is possible to design the purchasing group and allocate the payoff to every member. The purchasers may accept the coalition or not. If some members decline, the coordinator determines the group again until an agreement is reached [30, 24].

We propose mathematical programming formulations for the problem of buyer coalition formation, where a coordinator, a set of buyers, and merchants, negotiate bundles of products. The product price depends on the size of the total purchase; every buyer wishes to buy at most one bundle and has a reservation price for every item; hence the coordinator aims at forming coalitions of buyers in which each buyer is allocated to a bundle, to achieve the maximum social surplus given by the total reward minus the total cost of the purchase. We develop a nonlinear mixed-integer model for this problem and show that it is an NP-hard optimization problem. We present equivalent mixed-integer linear programming models for the case in which the unit price of a product is a general non-increasing function of the sales volume and for the case in which the product price is a decreasing step function of the product quantity sold. The computational performance of these models is evaluated on large-scale instances.

### 1.1. Related work

Buyers often wish to purchase more than one product. Firms sell bundles with more than one product or service at a more attractive price than what is sold separately [10]. This way, they can extract the surplus from customers by incorporating substitutes and complementary products into these bundles. Additionally, they can offer a bundle containing low-demand products mixed with high-demand items, reducing the corresponding stockpile flow and inventory costs. In this context, [20] proposed a Combinatorial Coalition Formation (CCF) problem, in which customers have a reservation price for the bundle and can buy more than one unit per product. The authors artificially divide the reservation price among items and then construct sub-coalitions per item. If the set of sub-coalitions is compatible, i.e., each buyer belongs to all sub-coalitions or none, they complete a coalition. Otherwise, a reservation price is recalculated, and a new sub-coalition set is formed until a compatible set is found. They prove that in the case of a linearly decreasing price in the number of products, once this algorithm finds a set of compatible sub-coalitions, the found coalition is optimal. Moreover, they compute the core with the corresponding payoff for each buyer. In [18], a *Mixed-Integer Programming* (MIP) is proposed to find a core of the coalition. The problem is addressed by considering the linear relaxation and its dual, leading to two proposed algorithms: *CoreP* and *CoreD*. The first aims to solve the IP-Primal, constructing the optimal coalition value and checking if the total payoff can be divided in such a way as to make all members of the coalition stay in it, as this is the best possible solution for them. The second algorithm solves the Dual problem and checks if satisfaction of efficiency actually solves the IP. In such a case, it represents a core element.

In [21], a CCF problem is formulated in which the reservation prices of customers are for bundles, and the goods are complementary or substitutes, considering heterogeneous buyers' preferences. Therefore this CCF problem results in coalitions for bundles instead of products. They show that the CCF problem is NP-hard by a reduction of the Complete Set Partitioning problem. A greedy algorithm based on cost-sharing ideas is proposed.

The Coalition Structure Generation (CSG) problem focuses on partitioning the buyers into mutually disjoint coalitions so that the total reward is maximized [26]. Three essential characteristics of CSG are described in [5]: 1) the group of buyers is goal-directed and has a short life; 2) the coordination is only between members of the corresponding coalition, and 3) the organization structure is flat. The term coalition is used in a much broader sense. In the CSG, the concept of an agent is used to refer to entities that are grouped to form coalitions, expanding the concept of the coalition to a wide range of applications in different domains. For instance, [31] presents a coalition that forms teams to accomplish activities. Each activity needs specific skills, and each agent has a different skill level. They formulate a MIP to find the coalition and show an  $\epsilon$ -core as the solution. Their solution method takes advantage of the problem structure and can solve instances with 500 agents in less than one hour. Additional prior work has considered new variants of the Combinatorial Structure Generation (CSG) problem, such as including shipping costs [29], location decisions [25], limits on available stock [17], and measures of centrality in Social Networks such as EigenCentrality, closeness and PageRank [6]. In addition, specific constraints have been included, such as physical limitations, legal banishments, and social relationships [13] or social ridesharing [4, 3].

The CSG has been shown to be NP-hard by a reduction of the Complete Set Partitioning problem. Therefore, several heuristic solution approaches have been developed for this problem, such as [14], including genetic algorithms [30, 17, 22] and ant colony optimization [29]. Additionally, two principal approaches exist to solve the problem: dynamic programming and anytime algorithm (for more details see the survey of [26]).

On another hand, Wholesale prices are widely used in retail, which represents increasing discounts on the unit price of a product with an increased sales volume. Such discounts are discrete, and they are applied when the sales volume reaches certain thresholds. It is therefore natural to represent wholesale prices as a decreasing step function of the purchased volume. The price is constant within a range of volume, then it decreases and stays constant until a new threshold is reached. In general, a step function is a piece-wise linear function for which there exist three types of formulations in the literature: the multiple-choice model, the incremental cost formulation, and the convex combination model [8]. We note that it is possible to approximate any nonlinear function using a piece-wise linear function, and the accuracy depends on the size of its linear segments.

In order to solve generic problems with piece-wise linear functions [9] proposes using a special ordered set of type 2 (SOS2). A set of binary variables are said to be SOS2 when at most two adjacent variables are nonzero. In [33] the branch-and-cut algorithm for LPs with piece-wise linear continuous costs is extended to the lower semi-continuous case. They also extend the SOS2 formulation for LPs with piece-wise linear continuous costs to the lower semi-continuous case. The extension of such formulations to the multivariate nonseparable case is studied in [32]. [19] studies the polyhedral structure of piecewise linear optimization problems and derives strong valid inequalities.

Piecewise linear functions have been applied to many real-life applications. For instance, the multi-choice model has been used for the multi-commodity network flow problem [28], the transportation problem [7], the facility location problem [15], the capacitated location problem with freight cost discount [12], and the network design problem [11], to name a few. The incremental model, in turn, has been applied to facility location problems [16], capacitated local access network problems [27], and multi-commodity network design with piecewise cost [2]. A step cost function is used as edge weights to represent economies of scale in [27]. Two formulations are considered to represent these cost functions, based on the incremental and the multi-choice models.

### 1.2. Our contribution

The main objective of this paper is to define mathematical programming formulations for the CCF problem in online purchase decisions. Two different product price functions are considered. The first formulation uses a general non-increasing price function, which defines a nonlinear model. Using a binary variable that indicates whether a certain number of items of a given product are purchased or not, a mixed-integer linear programming problem is obtained. The second formulation uses a decreasing step function according to the number of products. The resulting formulation is related to the incremental and multi-choice model presented in [32].

Additionally, two strengthened versions are proposed. A Benders Decomposition (BD) approach for the *step price function* is shown to solve large-scale instances. Computational experiments showed that the best results were obtained when we used BD only at the root node.

In synthesis, we present MIP formulations of the combinatorial coalition formation problem for the first time and apply Benders decomposition. We provide complexity proof and apply known valid inequalities. We solve the problems using B&C and C&B and compare the methods on large instances.

### 1.3. Outline of paper

The paper is organized as follows: Section 2 presents the problem and describes the formulation. In addition, it shows that the CCF problem considered is NP-hard. The proposed solution method is described in Section 3. Computational

tests comparing the different formulations and solution methods are presented in Section 4. Finally, we present our conclusions and lines for future work in section 5.

## 2. Problem definition and formulations

This Section presents a formal description of the Combinatorial Coalition Formation (CCF) problem we consider and analyzes its computational complexity. We provide two different formulations for CCF. The first can be used with any non-increasing price function, while the second uses a step price function. Moreover, we introduce valid inequalities that strengthen the formulations.

### 2.1. Problem description

Suppose a market offers multiple a set  $\mathcal{K}$  of products with non-additive values, i.e., products may be complementary or substitute. Products can be purchased by acquiring a bundle  $b_i$  defined as a non-empty set of products such that  $b_i \subseteq \mathcal{K}$ , and  $\mathcal{B}$  is the set of bundles. Note that a product  $k \in \mathcal{K}$  can appear at most once in a bundle. Let  $p_j(n)$  be a non-increasing function of the price of product  $j$  where  $n$  is the number of items purchased. A coordinator groups potential consumers in  $\mathcal{J}$  who wish to take advantage of the wholesale price. Consumer  $j \in \mathcal{J}$  wants to buy at most one bundle from the set of bundles  $\mathcal{B}_j \subset \mathcal{B}$  that is interesting for her. Consumers have heterogeneous preferences over the product in the bundles, leading to reservation price  $r_{i,j}$  for bundle  $b_i \in \mathcal{B}_j$  for consumer  $j \in \mathcal{J}$ .

Then we define the coalition  $C_k$  of customers that purchase product  $k \in \mathcal{K}$ . The price of product  $k$  for this coalition  $p_k(|C_k|)$ . By extension, a bundle coalition  $C_{b_i}$  is the buyer group that buys bundle  $b_i$ . Then, we can define  $C_k = \cup_{b_i | k \in b_i} C_{b_i}$ . Therefore, a coalition configuration denoted by  $C$  is the set of bundles coalitions  $C_{b_i}$ ,  $b_i \in \mathcal{B}$ . The surplus of coalition configuration,  $v(C)$ , is defined as:

$$v(C) = \sum_{C_{b_i} \in C} \left( \sum_{k \in C_{b_i}} r_{i,k} - \sum_{j \in b_i} p_j(|C_j|) |C_j| \right)$$

The coordinator must decide which bundle each consumer purchases to maximize the surplus obtained by subtracting the total cost of bundles from the reservation prices. The coalition utility must be non-negative; otherwise, it does not make sense. In this case, it is essential to allow for the existence of utility transfer between coalition members. Indeed, some buyers might have individual negative utility, which helps reduce the prices in the aggregate, thus increasing the surplus of the coalition participants. After the coalitions are formed, the coordinator must determine the payoff for each coalition member according to some criterion (typically fairness or stability). For instance, [21] used the cost-share as a stability criterion. In this case, members with negative individual utility paid their reservation price - obtaining zero utility - and the rest of the participants receive the payoff with the remaining surplus; therefore, the surplus is distributed so that every coalition member has a positive gain.

Table 1 provide the set and parameter notation.

#### 2.1.1. Example

We present a small illustrative instance of Combinatorial Coalition Formation from [21]. Assume four buyers  $a, b, c$  and  $d$  are offered three items  $g_1, g_2$  and  $g_3$ , which are camera A, camera B and a flash memory, respectively. The cameras are considered substitutes (though not necessarily perfect substitutes), while any camera and the memory

Sets		Parameters	
$\mathcal{J}$	set of buyers	$r_{i,j}$	reservation price of client $j \in \mathcal{J}$
$\mathcal{K}$	set of products		for bundle $b_i \in \mathcal{B}_j$
$\mathcal{B}$	set of bundles	$p_k(n)$	unit price for product $k \in \mathcal{K}$
$\mathcal{B}_j$	set of attractive bundles of customer $j$		with demand $n$

Table 1: Notations of the sets and parameters

card are complementary goods. Table 2 shows the reservation prices of the buyers, while Table 3 provides the price-quantity curve for each item, as it specifies the unit price per item as a non-increasing discrete function of the quantity. Now, assume that the bundle coalitions are configured as follows:  $C_{2,3} = \{a, b\}$  for  $g_{2,3}$ , and  $C_2 = \{c, d\}$  for  $g_2$ . Notice that this is equivalent to the set of item coalitions:  $\tilde{C}_2 = \{a, b, c, d\}$  for  $g_2$ , and  $\tilde{C}_3 = \{a, b\}$  for  $g_3$ . Given this configuration, no buying group is formed for  $g_1$ , the resulting price for  $g_2$  is 350 (four buyers), and that for  $g_3$  is 40 (two buyers). Thus, the unit price of the bundle  $g_{2,3}$  is  $350 + 40 = 390$ . Then, the total cost of  $C_{2,3}$  is  $2 \cdot 390 = 780$ , and the cost of the  $C_2$  is  $2 \cdot 350 = 700$ . Therefore, the surplus of the bundle coalition  $C_{2,3}$  is  $385 + 405 - 780 = 10$ , and the surplus of  $C_2$  is  $355 + 360 - 700 = 15$ . Finally, the total surplus of the coalition configuration is  $10 + 15 = 25$ .

	$g_1$	$g_2$	$g_3$	$g_{1,3}$	$g_{2,3}$	$g_{1,2}$	$g_{1,2,3}$
a	150	350	20	380	385	500	545
b	380	365	25	400	405	600	650
c	0	355	0	0	360	355	360
d	360	360	0	365	370	510	525

	Volume (units)				
	1	2	3	4	$\geq 5$
$g_1$	340	340	320	320	320
$g_2$	365	365	365	350	340
$g_3$	40	40	38	38	35

### 2.1.2. Computational complexity of CFF

We first define the decision version of problem CCF as follows:

*Instance:* A set  $\mathcal{J}$  of buyers, a set  $\mathcal{K}$  of products, and a set  $\mathcal{B} = \{b_1, \dots, b_{|\mathcal{B}|}\}$  of bundles where  $b_i \subseteq \mathcal{K}$ . Each buyer  $j \in \mathcal{J}$  has a reservation price  $r_{i,j}$  for the bundle  $b_i$ . The unit price of product  $k$  is  $p_k(|S|)$  where  $|S|$  is the number of buyers that purchase product  $k$ . An integer number  $M$ .

*Question:* Does there exist a coalition structure  $C = \{C_{b_1}, \dots, C_{b_{|\mathcal{B}|}}\}$  with  $C_{b_i} \subset \mathcal{J}$  for all  $b_i \in \mathcal{B}$  and  $C_{b_i} \cap C_{b_{i'}} = \emptyset$ , for all  $i, i' \in \mathcal{I}$  and whose value

$$v(C) = \sum_{b_i \in \mathcal{B}} \left( \sum_{j \in C_{b_i}} r_{i,j} - \sum_{k \in b_i} p_k(|C_k|) |C_k| \right)$$

is it at least  $M$ ?

Notice a link between the *decision* and *maximization* problems exists. In fact, for one side, whether there exists a polynomial algorithm that solves the decision problem, hence it can find the maximum value for the maximization problem in polynomial time by applying this algorithm iteratively while decreasing the goal value. On the other hand, whether an algorithm finds the maximum value of the maximization problem in polynomial time, then the decision problem can be solved in polynomial time by comparing the value of the solution output by this algorithm with the goal value. Therefore both versions are of similar theoretical difficulty.

**Theorem 1.** *The decision problem CCF is NP-complete even if all bundles are singletons.*

*Proof.* First, it is easy to see that the decision version of Problem CCF belongs to NP. Indeed, given a coalition structure  $C$ , its value  $V(C)$  can be computed in polynomial time.

Then, to prove that decision problem CCF is NP-complete, we show that *Set Packing Problem* reduces polynomially to CCF. First, we define Problem Set Packing (SP) as follows :

- *Instance* : A collection  $S$  of finite sets  $S_i, i = 1, \dots, m$  and an integer  $M$ .
- *Question* : Does  $S$  contains  $M$  mutually disjoint sets?

Given an instance of SP, we construct an instance of CCF as follows. The set of buyers  $\mathcal{J} = \cup_{i=1}^m S_i$  and the set of bundles  $\mathcal{B}$  contains  $m$  bundles  $b_j$  that are all distinct singletons. We define the reservation price  $r_{i,j}$  of buyer  $j$  for bundle  $b_i$  and the price function  $p_i(k)$  as follows :

$$r_{i,j} = \begin{cases} \frac{|S_i|+2}{|S_i|} & \text{if } j \in S_i \\ 0 & \text{if } j \notin S_i \end{cases}; \quad p_i(k) = \begin{cases} 2 \cdot |I| & \text{if } k < |S_i| \\ \frac{|S_i|+1}{|S_i|} & \text{if } k \geq |S_i| \end{cases}$$

Remark that as  $\mathcal{J}$  is the set of customers,  $\mathcal{J} \neq \emptyset$  and  $|\mathcal{J}| \geq |S_i|$  then  $2 \cdot |I| > (|S_i| + 1)/|S_i|$ . Hence, we can define our problem as  $\forall i$  determine a coalition  $C_i$  such that each customer belongs to at most one coalition and the value of coalition structure is:

$$v(C) = \sum_{i=1}^m \left( \sum_{j \in C_i} r_{i,j} - p_i(|C_i|)|C_i| \right) \geq B$$

We now show that there exists a coalition structure  $C$  with value  $V(C) \geq B$  if and only if there exist  $M$  mutually disjoint sets in  $\{S_i\}$ . Let define the contribution  $v(C_i)$  of product  $i$  to the value  $V(C)$  of a coalition structure as

$$v(C_i) = \sum_{j \in C_i} r_{i,j} - p_i(|C_i|)|C_i|$$

We prove that  $v(C_i)$  is equal to 1 if  $C_i = S_i$  and is negative otherwise.

We consider three different possible cases as follows.

(i) if  $|C_i| \geq |S_i| + 1$  then

$$\begin{aligned} v(C_i) &= \sum_{j \in C_i \cap S_i} r_{i,j} - p_i(|C_i|)|C_i| = |C_i \cap S_i| \frac{|S_i|+2}{|S_i|} - \frac{|S_i|+1}{|S_i|} |C_i| \\ &\leq |S_i| \frac{|S_i|+2}{|S_i|} - \frac{(|S_i|+1)^2}{|S_i|} = \frac{-1}{|S_i|} < 0. \end{aligned}$$

where the inequality is obtained since  $|C_i \cap S_i| \leq |S_i|$  and  $|C_i| \geq |S_i| + 1$ .

(ii) if  $|C_i| \leq |S_i| - 1$  then

$$\begin{aligned} &= |S_i| + 2 - \left(1 + \frac{2}{|S_i|}\right) - 2|I| = \\ &= |S_i| + 1 - \frac{2}{|S_i|} - 2|I| < |S_i| + 1 - 2|I| \leq 1 - |I| < 0 \end{aligned}$$

where the first inequality is obtained since  $|C_i \cap S_i| \leq |S_i| - 1$ . The last inequality is obtained due to  $|C_i| \leq |I|$ .



(iii) if  $|C_i| = |S_i|$  and  $C_i \neq S_i$  then  $v(C_i) = |C_i \cap S_i| \frac{|S_i|+2}{|S_i|} - \frac{|S_i|+1}{|S_i|} |S_i|$  as  $C_i \neq S_i$  implies  $|C_i \cap S_i| \leq |S_i| - 1$ . Hence,

$$v(C_i) \leq (|S_i| - 1) \frac{(|S_i| + 2)}{|S_i|} - |S_i| - 1 = |S_i| + 2 - 1 - \frac{2}{|S_i|} - |S_i| - 1 = \frac{-2}{|S_i|} < 0$$

Finally, if  $C_i = S_i$  then

$$v(C_i) = |S_i| \frac{|S_i| + 2}{|S_i|} - \frac{|S_i| + 1}{|S_i|} |S_i| = |S_i| + 2 - |S_i| - 1 = 1.$$

Therefore, a Set Packing with  $B$  sets exists if and only if a group of coalition with total value  $B$  exists. The CCF optimization problem is, consequently, NP-hard.  $\square$

## 2.2. CCF formulations

The combinatorial coalition formation problem can be modeled as a mathematical program. We propose two different mathematical programming formulations depending on the characteristics of the product price as a function of the product quantity. The first can be used for any non-increasing price as a function of product quantity, while the second formulation assumes the product price is a decreasing step-wise function of the quantity.

### 2.2.1. A nonlinear optimization formulation

We define a binary variable  $y_{i,j}$  that equals 1 when the customer  $j$  purchases bundle  $b_i$  and 0 otherwise. We also consider a continuous variable  $z_k$  that indicates the number of customers that purchase product  $k$ , which corresponds to the size of the coalition  $C_k$ . The coalition formation problem can be stated as follows:

$$(NLP) \quad \max \sum_{j \in \mathcal{J}} \sum_{b_i \in \mathcal{B}_j} r_{i,j} y_{i,j} - \sum_{k \in \mathcal{K}} p_k(z_k) z_k \quad (1a)$$

$$s.t. \quad \sum_{b_i \in \mathcal{B}_j} y_{i,j} \leq 1 \quad (j \in \mathcal{J}), \quad (1b)$$

$$\sum_{j \in \mathcal{J}} \sum_{\substack{b_i \in \mathcal{B}_j \\ b_i \ni k}} y_{i,j} = z_k \quad (k \in \mathcal{K}), \quad (1c)$$

$$y_{i,j} \in \{0, 1\} \quad (b_i \in \mathcal{B}_j), (j \in \mathcal{J}), \quad (1d)$$

$$z_k \geq 0 \quad (k \in \mathcal{K}). \quad (1e)$$

The objective function maximizes the social utility (1a). Constraints (1b) indicate that every customer can buy at most one bundle, and constraints (1c) determine the number of customers who buy product  $k$ . The model above is a nonlinear mixed-integer optimization problem, where the nonlinearity is due to the total purchase cost in the objective function.

Given that customer purchasing decisions are integer, the total quantity  $z_k$  of product  $k$  sold is also an integer. Therefore, the values of product prices turn out to be a discrete group of prices. Let us denote by  $p_{k,l}$  the unit price of product  $k$  if  $l$  units of that product are sold. These price values can be determined in  $O(|\mathcal{J}||\mathcal{K}|)$  time. Last, when the price function of each product is constant, i.e.,  $p_{k,l} = p_k$ , for all possible  $l$ , then (NLP) is a mixed integer linear optimization problem. The optimal coalition structure is obtained by assigning each customer to the bundle that maximizes the difference between her reservation price for this bundle and the sum of the prices of the products constituting the bundle.

Based on these observations, we now present two mixed integer linear formulations of the coalition formation problem. The first formulation, *Generic Price Function (GPF)* formulation, can be used for any price function  $p_k(z_k)$ . The second formulation is for a non-increasing price step function.

### 2.2.2. Generic Price Function formulation

Given that the number of purchases of every product is integer, we define a binary variable  $x_{k,l}$  that is equal to 1 if  $l$  customers purchase product  $k$  and 0 otherwise, for  $k \in \mathcal{K}$  and  $l \in \mathcal{L} = \{1, 2, \dots, |\mathcal{I}|\}$ . We reformulate the problem (NLP) as a mixed-integer linear model, called *Generic Price Function* (GPF), by adding the following set of constraints for each product  $k \in \mathcal{K}$ :

$$\sum_{l \in \mathcal{L}} l \cdot x_{k,l} = z_k, \quad (2a)$$

$$\sum_{l \in \mathcal{L}} x_{k,l} \leq 1, \quad (2b)$$

$$x_{k,l} \geq 0. \quad (2c)$$

Then, the nonlinear terms  $p_k(z_k)z_k$  in the objective function of (NLP) can be replaced by the following linear expression  $p(z_k)z_k = \sum_{l \in \mathcal{L}} l p_{k,l} x_{k,l}$ .

### 2.2.3. Step Price Function formulation

In many cases, given its simplicity, the retailer uses wholesale prices that follow a non-increasing step function, in which the price remains constant within a purchase range. When it exceeds the upper limit of the range, the price is reduced and remains constant until the next limit. Although the (GPF) can model this type of function, the number of binary variables it uses may be larger than necessary. Therefore, given the particular structure of the piecewise constant decreasing price function, we introduce the *Step Price Function* (SPF) formulation based on [8], who proposed a *the multiple-choice model* and an *incremental model* for the piecewise function described below.

Given a product  $k \in \mathcal{K}$ , we set a collection of purchase size thresholds  $a_{k,1}, \dots, a_{k,L_k}$  and corresponding prices  $d_{k,1} > d_{k,2} > \dots > d_{k,L_k}$ . The set of indices of the different segments of the price function of product  $k$  is denoted by  $\mathcal{L}_k = \{1, \dots, L_k\}$ . By definition, we set  $ba_{k,0} = 0$  for each product  $k \in \mathcal{K}$ . Therefore, given that  $z_k$  buyers purchase product  $k$ , where  $z_k$  belongs to the interval  $[a_{k,l}, a_{k,l+1})$ , the price of product  $k$  that these buyers pay is defined as:  $p_k(z_k) = d_{k,l}$ . Figure 2.2.3 illustrates a step price function with this notation.

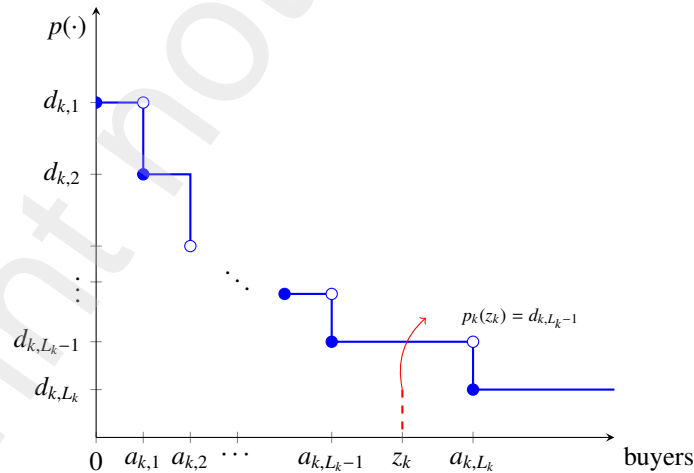


Figure 1: Step price function representation

The multi-choice-model [1] uses binary variable  $s_{k,l}$  to identify the interval in the price function that is active. Specifically, we have  $s_{k,l} = 1$  if  $z_k \in [a_{k,l-1}, a_{k,l})$  for  $k \in \mathcal{K}$  and  $l \in \mathcal{L}_k$ . The model also uses variable  $\tilde{z}_{k,l}$  to decompose the total purchases of product  $k$  into the intervals of the step function, and finally  $\varphi_k$  a nonnegative variable that represents the total cost paid for product  $k$ . With these variables, we can express the coalition formation problem given a step

price function as a mixed integer linear optimization problem. To do this, we replace  $p(z_k)z_k = \varphi_k$  in the objective function and add constraints (3a)-(3e) below. The resulting model represents the *multi-choice model* for step function price.

$$\sum_{l \in \mathcal{L}_k} \tilde{z}_{k,l} = z_k \quad (3a)$$

$$\sum_{l \in \mathcal{L}_k} s_{k,l} = 1 \quad k \in \mathcal{K}, \quad (3b)$$

$$a_{k,l-1} s_{k,l} \leq \tilde{z}_{k,l} \quad k \in \mathcal{K}, l \in \mathcal{L}_k, \quad (3c)$$

$$a_{k,l} s_{k,l} \geq \tilde{z}_{k,l} \quad k \in \mathcal{K}, l \in \mathcal{L}_k, \quad (3d)$$

$$d_{k,l} z_k - (1 - s_{k,l}) M \leq \varphi_k \quad k \in \mathcal{K}, l \in \mathcal{L}_k. \quad (3e)$$

The constraint (3a) establishes the relation between  $z_k$  and  $\tilde{z}_{k,l}$ ; constraint (3b) forces that just one interval  $l \in \mathcal{L}_k$  must be active. Constraints (3c) and (3d) limit the value  $\tilde{z}_{k,l}$  to the bounds of the interval  $l \in \mathcal{L}_k$ , while constraints (3e) determines the total cost  $\varphi_k$  of the product  $k \in \mathcal{K}$ . Note that the big  $M$  can be tightened by setting  $M=d_{k,L_k}$ . The model generated by embedding these constraints set to (NLP) is called (SPF -  $M$ ).

Note that for the *multi-choice model* only one variable  $s_{k,l}$  with  $l \in \mathcal{L}_k$  can be active for each product  $k$ , i.e.,  $\tilde{z}_{k,l}$  is equal zero for all intervals, except the variable associated with the interval active. Thus, we can substitute the constraint (3e) by the following constraint for  $k \in \mathcal{K}$ :

$$\sum_{l \in \mathcal{L}_k} d_{k,l} \tilde{z}_{k,l} = \varphi_k \quad k \in \mathcal{K} \quad (\text{cost-SPF})$$

From another point of view, the *incremental model* uses the variables  $\tilde{z}_{k,l}$  and  $s_{k,l}$  in a different way. In this case, the variable  $s_{k,l}$  is active in each segment until the interval is reached such that  $z_k \in \{a_{k,l-1}, a_{k,l}\}$  while the  $\tilde{z}_{k,l}$  completes the intervals until they reach  $a_{k,l}$ . Hence the constraint (3b) is not necessary. Therefore, we can replace the constraints set (3b) - (3e) for each product by the following set of constraints:

$$\sum_{l \in \mathcal{L}_k} \tilde{z}_{k,l} = z_k \quad (4a)$$

$$(a_{k,l} - a_{k,l-1}) \cdot s_{k,l+1} \leq \tilde{z}_{k,l} \quad l \in \mathcal{L}_k \quad (4b)$$

$$(a_{k,l} - a_{k,l-1}) \cdot s_{k,l} \geq \tilde{z}_{k,l} \quad l \in \mathcal{L}_k \quad (4c)$$

$$d_{k,l} z_k - (1 + s_{k,l} - s_{k,l+1}) \cdot M \leq \varphi_k \quad l \in \mathcal{L}_k \quad (4d)$$

Again, the big  $M$  can be tightened by using  $d_{k,L_k}$  for every  $l \in \mathcal{L}_k$  and a similar reformulation to (cost-SPF) of this constraint can be used. We note that both, the *multiple-choice model* and the *incremental model*, consider the same number of total variables, while the *multiple-choice model* adds  $\sum_{k \in \mathcal{K}} 4 \cdot L_k + 2$  constraints and the *incremental model* adds  $\sum_{k \in \mathcal{K}} 3 \cdot L_k + 1$  additional constraints to (NLP). Recall that  $L_k = |\mathcal{L}_k|$ . Our preliminary computational results show that both the *incremental model* and *multiple-choice model* have a similar performance. This is why, in the rest of this article, we consider the *multiple-choice model* because it is easier to develop a Benders decomposition method for this model.

### 3. Solution methods

The formulations (GPF) and (SPF) for the CCF problem presented above are binary optimization models, and a commercial solver can directly handle small-size instances. However, real-world applications often involve many buyers and product offers or require a solution in a short time frame, as in a web platform. Then, a special procedure

is required to find the optimal solution in a reduced time. In contrast to previous works that have proposed heuristic methods [20, 21, 18], we focus on an exact method. In particular, to efficiently solve large-scale instances, we investigate the decomposition strategy for the CCF. In this section, we present a Benders decomposition strategy that applies to (SPF-M). To our knowledge, Benders decomposition has not been previously applied to CCF problem.

The Benders decomposition splits a mixed integer programming into a master problem incorporating the binary variables and a subproblem with continuous variables that use the solution of the master problem as parameters. Through a cut-generation method, Benders cuts are generated iteratively for the master problem using the subproblem as a separation problem. To describe the method, we first rewrite the model (SPF-M). Consider the equation (cost-SPF) to replace the variable  $\varphi$  and the equation (3a) to substitute the variable  $z$  in equation (1c) by the variable  $\tilde{z}$  obtaining of following simplified formulation:

$$(RSPF1) \max \sum_{j \in \mathcal{J}} \sum_{b_i \in \mathcal{B}_j} r_{i,j} y_{i,j} - \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{L}_k} d_{kl} \tilde{z}_{k,l} \quad (5a)$$

$$s.t. \sum_{j \in \mathcal{J}} \sum_{\substack{b_i \in \mathcal{B}_j \\ b_i = k}} y_{i,j} = \sum_{l \in \mathcal{L}_k} \tilde{z}_{k,l} \quad k \in \mathcal{K}, \quad (5b)$$

$$\sum_{b_i \in \mathcal{B}_j} y_{i,j} \leq 1 \quad j \in \mathcal{J}, \quad (5c)$$

$$\sum_{l \in \mathcal{L}_k} s_{k,l} = 1 \quad k \in \mathcal{K}, \quad (5d)$$

$$a_{k,l-1} s_{k,l} \leq \tilde{z}_{k,l} \quad k \in \mathcal{K}, l \in \mathcal{L}_k, \quad (5e)$$

$$a_{k,l} s_{k,l} \geq \tilde{z}_{k,l} \quad k \in \mathcal{K}, l \in \mathcal{L}_k, \quad (5f)$$

$$s_{k,l} \in \{0, 1\} \quad k \in \mathcal{K}, l \in \mathcal{L}_k, \quad (5g)$$

$$y_{i,j} \in \{0, 1\} \quad b_i \in \mathcal{B}_j, j \in \mathcal{J}. \quad (5h)$$

$$\tilde{z}_{k,l} \geq 0 \quad k \in \mathcal{K}, l \in \mathcal{L}_k. \quad (5i)$$

Note that the Benders decomposition applies naturally in (RSPF1) because the variables can be divided into the integer variables  $(s, y)$  and continuous variables  $\tilde{z}$ . In the master problem with integer variables, we decide what bundles are purchased and what is the active interval, i.e., the value  $\tilde{z}_k$  of each product  $k \in \mathcal{K}$ , which can be split in a subproblem per product. Observe that the variable  $s$  activates the price interval corresponding to the number of buyers that purchase product  $k \in \mathcal{K}$ . Hence, given a solution  $(s, y)$ , if the number of buyers does not correspond to the interval given by  $s$ , the sub-problem is infeasible; otherwise, is feasible.

We define the master problem that contains the binary variables as follows:

$$\max \sum_{j \in \mathcal{J}} \sum_{b_i \in \mathcal{B}_j} r_{i,j} y_{i,j} - \sum_{k \in \mathcal{K}} \xi_k, \quad (6a)$$

$$s.t. \sum_{b_i \in \mathcal{B}_j} y_{i,j} \leq 1 \quad j \in \mathcal{J}, \quad (6b)$$

$$\sum_{l \in \mathcal{L}_k} s_{k,l} = 1 \quad k \in \mathcal{K}, \quad (6c)$$

$$\sum_{l \in \mathcal{L}_k} s_{k,l} a_{k,l-1} \leq \sum_{j \in \mathcal{J}} \sum_{\substack{b_i \in \mathcal{B}_j \\ b_i = k}} y_{i,j} \quad k \in \mathcal{K}, \quad (6d)$$

$$\sum_{j \in \mathcal{J}} \sum_{\substack{b_i \in \mathcal{B}_j \\ b_i \geq k}} y_{i,j} \leq \sum_{l \in \mathcal{L}_k} s_{k,l} a_{k,l} \quad k \in \mathcal{K}, \quad (6e)$$

$$\alpha_k^e \sum_{\substack{b_i \in \mathcal{B}_j \\ b_i \geq k}} y_{i,j} + \sum_{l \in \mathcal{L}_k} \beta_{k,l}^e a_{k,l} s_{k,l} + \sum_{l \in \mathcal{L}_k} \gamma_{k,l}^e a_{k,l-1} b e s_{k,l} \leq \xi_k \quad k \in \mathcal{K}, e \in \mathcal{E}_k, \quad (6f)$$

$$\alpha_k^r \sum_{\substack{b_i \in \mathcal{B}_j \\ b_i \geq k}} y_{i,j} + \sum_{l \in \mathcal{L}_k} \beta_{k,l}^r a_{k,l} s_{k,l} + \sum_{l \in \mathcal{L}_k} \gamma_{k,l}^r a_{k,l-1} s_{k,l} \geq 0 \quad k \in \mathcal{K}, r \in \mathcal{R}_k, \quad (6g)$$

$$y_{i,j} \in \{0, 1\} \quad b_i \in \mathcal{B}_j, j \in \mathcal{J}, \quad (6h)$$

$$s_{k,l} \in \{0, 1\} \quad k \in \mathcal{K}, l \in \mathcal{L}_k. \quad (6i)$$

Here, constraints (6f) and (6g) are the *optimality cuts* and the *feasibility cuts* resulting from the Benders decomposition. For each  $k \in \mathcal{K}$ , the sets  $\mathcal{E}_k$  and  $\mathcal{R}_k$  denote the extreme points and extreme rays, respectively, of the polyhedron that defines the feasible region of the second stage problem described below. Since the number of extreme points and extreme rays can be large, a cut generation algorithm using Benders cuts is used.

Given  $(\hat{s}, \hat{y}, \hat{\xi})$ , a feasible solution to Master Problem, then the subproblem problem is separable in  $|\mathcal{K}|$  sub-problems as follows:

$$G_k(y, s) = \min_{\tilde{z}_k} \sum_{l \in \mathcal{L}_k} d_{k,l} \tilde{z}_{k,l}, \quad (7a)$$

$$s.t. \sum_{j \in \mathcal{J}} \sum_{\substack{b_i \in \mathcal{B}_j \\ b_i \in k}} \hat{y}_{i,j} = \sum_{l \in \mathcal{L}_k} \tilde{z}_{k,l} \quad (7b)$$

$$a_{k,l-1} \hat{s}_{k,l} \leq \tilde{z}_{k,l} \quad l \in \mathcal{L}_k, \quad (7c)$$

$$a_{k,l} \hat{s}_{k,l} \geq \tilde{z}_{k,l} \quad l \in \mathcal{L}_k, \quad (7d)$$

$$\tilde{z}_{k,l} \geq 0 \quad l \in \mathcal{L}_k. \quad (7e)$$

We remark that if the purchase size of product  $k$  -described by  $Y_k = \sum_{j \in \mathcal{J}} \sum_{\substack{a_i \in \mathcal{B}_j \\ a_i \in k}} \hat{y}_{i,j}$  does not coincide with the interval of the price at which by variable  $s$  is activated, hence the problem above is infeasible. Otherwise, the quantity  $Y_k$  belongs to the active price interval. Thus, the variable  $\tilde{z}$  corresponding to that interval takes the value of  $Y_k$ . Therefore, the following constraints for  $k \in \mathcal{K}$  are added to the master problem so that it is always feasible.

$$\sum_{l \in \mathcal{L}_k} s_{k,l} b_{k,l-1} \leq \sum_{j \in \mathcal{J}} \sum_{\substack{a_i \in \mathcal{B}_j \\ a_i \in k}} y_{i,j} \leq \sum_{l \in \mathcal{L}_k} s_{k,l} b_{k,l} \quad (\text{feas-MP})$$

Once the above inequality is added to the master problem, there are no extreme rays in the definition of the polyhedron associated with the feasible region of the problem  $G_k$ , i.e., set  $\mathcal{R}_k = \emptyset \forall k \in \mathcal{K}$ . Moreover, the following proposition 1 shows that the solution of the subproblem can be determined analytically, thus reducing computation time.

**Proposition 1.** Let  $k = \{1, \dots, |\mathcal{K}|\}$ ,  $y \in \{0, 1\}^{|\mathcal{I}| \times |\mathcal{B}|}$ ,  $s \in \{0, 1\}^{|\mathcal{K}| \times |\mathcal{L}_k|}$ ,  $l^* = \arg \max_{l \in \mathcal{L}_k} \{s_{k,l}\}$  and  $Y_k = \sum_{\substack{b_i \in \mathcal{B}_j \\ b_i \in k}} y_{i,j}$ . The value of optimal solution problem (7) is given by

$$G_k(s, y) = Y_k d_{k,l^*}, \quad \text{if } Y_k \in [b_{k,l^*-1}, b_{k,l^*}] \quad (8)$$

The proof of Proposition 1 follows from the fact that the problem is *only* feasible if  $Y_k \in [a_{k,l^*-1}, a_{k,l^*}]$ .

The generation to cuts for the master problem is based on the dual problem of the subproblem, which can be stated as follows:

$$\max_{\alpha_k, \beta_k, \gamma_k} \alpha_k Y_k + \sum_{l \in \mathcal{L}_k} \beta_{k,l} b_{k,l} \hat{s}_{k,l} + \sum_{l \in \mathcal{L}_k} \gamma_{k,l} b_{k,l-1} \hat{s}_{k,l} \quad (9a)$$

$$s.t. \alpha_k + \beta_{k,l} + \gamma_{k,l} \leq d_{k,l} \quad (l \in \mathcal{L}_k), \quad (9b)$$

$$\beta_{k,l} \leq 0 \quad (l \in \mathcal{L}_k), \quad (9c)$$

$$\gamma_{k,l} \geq 0 \quad (l \in \mathcal{L}_k). \quad (9d)$$

where  $(\alpha, \gamma, \beta)$  are the variables dual of constraints (7b), (7c) and (7d) respectively.

Similarly to the primal model, Proposition 2 shows how to solve the dual problem analytically.

**Proposition 2.** Let  $k = \{1, \dots, |\mathcal{K}|\}$ ,  $\mathbf{y} \in \{0, 1\}^{|\mathcal{I}|\times|\mathcal{B}|}$ ,  $\mathbf{s} \in \{0, 1\}^{|\mathcal{K}|\times|\mathcal{L}_k|}$ ,  $l^* = \arg \max_{l \in \mathcal{L}_k} \{s_{k,l}\}$  and  $Y_k = \sum_{\substack{b_i \in \mathcal{B}_j \\ b_i \in k}} y_{i,j}$  the optimal solution problem (9) is given by if  $Y_k \in [b_{k,l-1}, b_{k,l}]$ , and the optimal solution problem (9) is bound then

$$\alpha_k = d_{k,l^*}, \quad (10a)$$

$$\beta_{k,l} = \begin{cases} d_{k,l} - d_{k,l^*} & \text{if } l > l^*, \\ 0 & \text{otherwise.} \end{cases} \quad \forall l \in \mathcal{L}_k, \quad (10b)$$

$$\gamma_{k,l} = 0 \quad \forall l \in \mathcal{L}_k. \quad (10c)$$

*Proof.* The proof is divided into two steps: first, we show that the proposed solution is feasible, and then we show that the objective value of the proposed solution is exactly  $d_{k,l^*} Y_k$ , which is the optimal objective of the primal problem.

- *Feasibility:* Observe that  $d_{k,1} > d_{k,2} > \dots > d_{k,L_k}$ . Thus, we need to verify constraint (11a). For  $l \leq l^*$ , we have that

$$\begin{aligned} \alpha_k + \beta_{k,l} + \gamma_{k,l} &= d_{k,l^*} + 0 + 0 \\ &\leq d_{k,l} \end{aligned}$$

and for  $l > l^*$ , we have that

$$\begin{aligned} \alpha_k + \beta_{k,l} + \gamma_{k,l} &= d_{k,l^*} + (d_{k,l} - d_{k,l^*}) + 0 \\ &= d_{k,l} \end{aligned}$$

This concludes our proof of the feasibility  $(\alpha_k, \beta_k, \gamma_k)$ .

- *Objective function:* Observe that  $\beta_{k,l^*} = 0$ . Thus, we have that

$$\begin{aligned} \alpha_k \sum_{\substack{b_i \in \mathcal{B}_j \\ k \in a_i}} y_{i,j} + \sum_{l \in \mathcal{L}_k} \beta_{k,l} (b_{k,l} s_{k,l}) + \sum_{l \in \mathcal{L}_k} \gamma_{k,l} (a_{k,l-1} s_{k,l}) &= d_{k,l^*} Y_k + 0 + 0 \\ &= d_{k,l^*} Y_k \end{aligned}$$

which is exactly the optimal value of the primal problem. Thus, it follows that  $(\alpha_k, \beta_k, \gamma_k)$  - given by Proposition 1 - is an optimal solution dual of the sub-problem 7.

□

Therefore, the *Benders cut* proposed by Proposition 1 is defined by:

$$d_{k,l^*} z_k + \sum_{l'=l^*+1}^{L_k} (d_{k,l'} - d_{k,l^*}) a_{k,l'} s_{k,l'} \leq \varphi_k,$$

which is exactly the valid inequality cost-SPF suggested above for  $l = l^* \in \mathcal{L}_k$  and  $k \in \mathcal{K}$ .

We also obtain the solution of problem (7) when  $\hat{\mathbf{s}}^k$  and  $Y_k$  comes of solving the linear relaxation of the master problem (6), efficiently using the following Algorithm 1. The variable  $\tilde{z}_{k,l}$  must belong to interval  $[a_{k,l-1} \hat{s}_{k,l}, a_{k,l} \hat{s}_{k,l}]$  to satisfy the constraints (7c), (7d). By including the constraint (feas-MP), the primal problem is always feasible. Since  $Y_k$  is the purchase size of product  $k$ , we can set as base the lower bound of the interval as  $a_{k,l-1} \hat{s}_{k,l} = \tilde{z}_{k,l}$ . Hence, we add the complementary part to reach the upper bound or  $Y_k$ . Since problem (7) is a minimization problem and the price  $d_{k,l}$

---

**Algorithm 1** Algorithm for Solving Problem (7)

---

1: **Require:**  $Y_k = \sum_{\substack{b_i \in \mathcal{B}_j \\ b_i \in k}} y_{i,j}$ , set  $\mathcal{L}_k$  ordered s.t.  $d_{k,L_k} < \dots < d_{k,1}$   
2:  $\tilde{z}_{k,l} = a_{k,l-1} \hat{s}_{k,l}$  for all  $l \in \mathcal{L}_k$  and  $Z_k = \sum_{l \in \mathcal{L}_k} \tilde{z}_{k,l}$   
3: **for**  $l \in \{L_k, \dots, 1\}$  **do**  
4:      $\tilde{z}_{k,l} = \tilde{z}_{k,l} + \min\{Y_k - Z_k, a_{k,l} \hat{s}_{k,l} - \tilde{z}_{k,l}\}$   
5:      $Z_k = Z_k + \min\{Y_k - Z_k, a_{k,l} \hat{s}_{k,l} - \tilde{z}_{k,l}\}$   
6:     **if**  $Z_k = Y_k$  **then**  
7:          $G_k = \sum_{l \in \mathcal{L}_k} \tilde{z}_{k,l} d_{k,l}$   
8:          $l_k^* \leftarrow l$   
9:         **break**

---

decreases, we inverse order to fill the intervals, i.e., we begin with the interval with the lowest price. The process is stopped when  $Z_k = Y_k$ .

Once we obtain the  $l_k^*$  values from Algorithm 1, we apply Algorithm 2 to get the values of the dual variables. Similarly to the MIP of problem (6), we establish nonzero dual variables different to  $l_k^*$ . The idea is to add generate cuts that incorporate both variables  $(s, y)$  and add them to the master problem. Therefore, we give the value  $\alpha_k = d_{k,l_k^*}$  for all  $k \in \mathcal{K}$ . Note that  $l = l_k^*$  can be a constraint (7c) active or inactive, so for simplicity, the associated dual variables are set equal to zero. Additionally, the constraints (7d) with  $l_k^* > l$  and  $s_{k,l} > 0$  are inactive, then we set the dual variable  $\beta_{k,l} = 0$  for  $l_k^* > l$ . On the other hand, the constraints (7c) are active, then we have  $\gamma_{k,l} > 0$   $l_k^* > l$ . Similarly, the constraints (7c) with  $l_k^* < l$  and  $s_{k,l} > 0$  are inactive, then the dual variable  $\beta_{k,l} < 0$   $l_k^* > l$  the constraints (7d) are inactive, then  $\gamma_{k,l} = 0$  for  $l_k^* < l$ .

---

**Algorithm 2** Algorithm for Solving Problems (9)

---

1: **Require:**  $s^k, l_k^*$  obtained since the Algorithm ,  $Y_k = \sum_{j \in \mathcal{J}} \sum_{\substack{b_i \in \mathcal{B}_j \\ b_i \ni k}} y_{i,j}$   
2:  $\alpha_k = d_{k,l_k^*}, \beta_k = \gamma_k = 0$  for all  $l \in \mathcal{L}_k$   
3: **for**  $l \in \{1, \dots, L_k\}$  **do**  
4:     **if**  $l > l_k^*$  **then**  
5:          $\beta_{k,l} = d_{k,l} - d_{k,l_k^*}$   
6:     **else if**  $l < l_k^*$  **then**  
7:          $\gamma_{k,l} = d_{k,l_k^*} - d_{k,l}$

---

**Theorem 2.** Let  $\tilde{z}$  be the solution produced by Algorithm 1, and let  $(\alpha, \beta, \gamma)$  be the solution produced by Algorithm 2. Then,

- $\tilde{z}$  is feasible for the primal problem (7);
- $(\alpha, \beta, \gamma)$  is feasible for the dual problem (9); and
- the two solutions are optimal for their respective problems.

*Proof.* We proceed in three stages. First, we show that  $\tilde{z}$  is primal feasible. Second, we show that  $(\alpha, \beta, \gamma)$  is dual feasible. Finally, we show that the solutions satisfy complementary slackness, establishing that they are optimal.

- **Primal feasibility**

Note that constraints (7c) and (7d) are never violated in the algorithm. Since these constraints are never violated,

the slackness associated with the inequality constraint is never negative. We only need to verify that constraint (7b) is satisfied, i.e., that the sum of the  $\tilde{z}_{k,l}$  variables equals to  $Y_k$ .

Observe that we start with  $Z_k = \sum_{l \in \mathcal{L}_k} s_{k,l} a_{k,l-1}$ , then if  $Z_k = Y_k$  the algorithm terminates. If  $S_k < Y_k$  we add the remaining part. In decreasing order in  $\mathcal{L}_k$  (so that  $d_{k,L_k} > d_{k,L_k-1} > \dots > d_{k,1}$ ), we increase  $\tilde{z}_{k,l}$  by  $\min\{Y_k - Z_k, a_{k,l} s_{k,l} - a_{k,l-1} s_{k,l}\}$  i.e., we increase  $\tilde{z}_{k,l}$  so that either  $Y_k$  is reached or the maximum quantity available in interval  $l$  is reached. The algorithm ends in the first case, while in the second case, we continue with the next interval. Note that when  $\hat{s}_{k,l} = 0$  this interval does not participate in the sum and can be omitted.

- **Dual feasibility**

By definition to Algorithm 2, we have that

$$\alpha_k = d_{k,l^*} \text{ and } \beta_{k,l^*} = \gamma_{k,l^*} = 0$$

The dual constraints are :

$$\alpha_k + \beta_{k,l} + \gamma_{k,l} \leq d_{k,l} \quad l \in \mathcal{L}_k, \quad (11a)$$

$$\beta_{k,l} \leq 0 \quad l \in \mathcal{L}_k, \quad (11b)$$

$$\gamma_{k,l} \geq 0 \quad l \in \mathcal{L}_k. \quad (11c)$$

Following the initial definition, when Algorithm 2 terminates, we obtain  $\gamma_{k,l} = 0$  and  $\beta_{k,l} = d_{k,l} - d_{k,l^*} \leq 0$  for all  $l \in \{1, \dots, l^*\}$ . Analogously, we get  $\beta_{k,l} = 0$  and  $\gamma_{k,l} = d_{k,l} - d_{k,l^*} \geq 0$  for all  $l \in \{l^*, \dots, L_k\}$ . This setting of the dual variables ensures that all constraints are active. Consequently, the solution is dual feasible.

- **Complementary slackness**

The complementary slackness conditions are:

$$\alpha_k \cdot \sum_{j \in \mathcal{J}} \sum_{\substack{b_i \in \mathcal{B}_j \\ k \in b_i}} y_{i,j} - \sum_{l \in \mathcal{L}_k} \tilde{z}_{k,l} = 0, \quad (12a)$$

$$\beta_{k,l} \cdot (b_{l-1} s_{k,l} - \tilde{z}_{k,l}) = 0, \quad (12b)$$

$$\gamma_{k,l} \cdot (b_l s_{k,l} - \tilde{z}_{k,l}) = 0, \quad (12c)$$

$$\hat{z}_{k,l} \cdot (\alpha_k + \beta_{k,l} + \gamma_{k,l} - d_{k,l}) = 0 \quad (12d)$$

Equation (12a) is satisfied due to (7b). Similarly, the solution  $(\alpha, \beta, \gamma)$  obtained from Algorithm 2 satisfies by construction constraints (11a) with equality, implying that equation (12d) is satisfied. In addition, this solution satisfies equations (12b) and (12c) since  $\gamma_{k,l} = 0$  for all  $l \in \{1, \dots, l^* - 1\}$  and  $b_{k,l} s_{k,l} = \tilde{z}_{k,l}$  otherwise, and  $\beta_{k,l} = 0$  if  $l \in \{l^*, \dots, L_k\}$  with  $b_{k,l-1} s_{k,l} = \tilde{z}_{k,l}$  otherwise. Then, since complementary slackness holds, the primal and dual solutions are optimal. □

Note that extending the method for a variant includes a capacity constraint on the products is possible. In fact, we can do this in two ways: indirectly and directly. First, we can adjust the price function, adding a sufficiently large value for quantities above the available capacity of the product. Another alternative is to add the capacity constraint directly, in which case we can add the constraint to the main problem in the following way without changing the subproblem:

$$\sum_{j \in \mathcal{J}} \sum_{\substack{b_i \in \mathcal{B}_j \\ b_i \in k}} y_{i,j} \leq \text{cap}_k \quad k \in \mathcal{K}, \quad (\text{cap-SPF})$$

where  $\text{cap}_k$  is the available capacity of product  $k \in \mathcal{K}$ .



## 4. Computational experiments

In this Section, we present the computational tests performed. All procedures and algorithms have been written using the language Python; we used GUROBI version 9.1. for the resolution of the MIPs. In Section 4.1 we describe the synthetic data that were used in our experiments, and Section 4.2 report the results of our experiments.

### 4.1. Instances: description

We now describe the generation of instances used for our computational experiments. Due to the fact that SPF supports only a step price function, we compare the performance of both GPF and SPF for this type of price function. Therefore, we describe below the reservation price, the function price, and finally, the bundle formation considered. The instances considered are based on the experiment set used in [21] for Combinatorial Coalition Formation.

*Reservation prices:* the reservation price depends on the bundle size rather than the products it contains. We establish the willingness to pay for the bundle  $b_i$  of purchaser  $j$  as  $r_{i,j} \in [\underline{r} \cdot k^\alpha, \bar{r} \cdot k^\alpha]$  where  $\underline{r}$  and  $\bar{r}$  are the lower and upper bounds of the reservation price for a single item,  $k$  is the number of items in the bundle, and  $\alpha > 0$  indicate if the products in a bundle are substitutes or complements. In this manner,  $\alpha = 1$  represents the case in which the customer is indifferent between buying a bundle or buying the items in the bundle separately.  $\alpha > 1$  represents the case in which the buyer prefers to purchase the items as a bundle instead of a subset of them, i.e., the products are complementary. In contrast, when  $\alpha < 1$ , the reservation price of purchasing the items as a bundle is less than buying a subset of them, i.e., the items are substitutes.

*Price-quantity function:* it describes the price drop between  $p_h$  and  $p_l$  as a function of an increasing quantity. We assume that the price drops from  $p_h$  to  $p_l$  regularly, i.e., each decreasing step corresponds to equal increases in the order quantity and results in equal price reductions. We denote the *Number of steps* by  $R_k \in [\underline{R}, \bar{R}]$  where  $\underline{R}$  and  $\bar{R}$  are the minimum and a maximum number of steps respectively. The Price Decrease Rate (PDR) is defined as the ratio of price reduction to the quantitative increase over each step. We define  $\underline{PDR}$  and  $\bar{PDR}$  as the minimum and maximum step size, respectively. The PDR determines the size of the quantity discount. The higher the PDR, the more buyers there are in the buying group. With more steps on the price curve, forming the buying group is easier because even a small increase in the quantity ordered can lower the price.

*Bundle formation:* we randomly select  $|\mathcal{B}|$  sets from the full list of possible permutations with a maximum length  $Q$  of the  $|\mathcal{K}|$  products with  $Q < |\mathcal{K}|$ . In preliminary experiments, we observed that instances with overlap between bundles are harder to solve. So, in order to exploit possible overlaps between bundles, we fix the bundle size to six and assume that there exist two essential that substitute each other, and that only one of the two appears in each bundle. For instance, suppose that  $\mathcal{K} = \{k_1, k_2, k_3, k_4, k_5\}$  with  $k_1$  and  $k_2$  as essential products. If the bundle size is fixed to three, then the possible bundles are  $\mathcal{B}$  are  $\{k_1, k_3, k_4\}$ ,  $\{k_1, k_3, k_5\}$ ,  $\{k_1, k_4, k_5\}$ ,  $\{k_2, k_3, k_4\}$ ,  $\{k_2, k_3, k_5\}$  and  $\{k_2, k_4, k_5\}$ .

For our experiments, we used the following values:  $\underline{r} = 80, \bar{r} = 130, |\mathcal{J}| = \{100, 150, 200, 300, 500, 1000\}$ ;  $|\mathcal{B}| = \{15, 31, 62, 93, 124, 155, 207\}$ ;  $p_l = 70, p_h = 100$ ;  $|\mathcal{K}| = 15$ ;  $Q = 5$ ;  $\alpha = \{0.5, 0.55, \dots, 1.25, 1.30\}$ ;  $\underline{R} = 4$  and  $\bar{R} = 10$ . We generated 100 instances randomly for each set of parameters value described above.

### 4.2. Comparison between the different formulations

Average statistics over all instances are reported in Table 4. The column headings are defined as follows:

- Instance: Name of instance;

- GAP: average gap  $100 \frac{UB-LB}{LB} \%$ , where UB(LB) is the best upper(lower) bound;
- RGAP: average integrality gap  $100 \frac{LP-LB}{LB} \%$ , with LP = linear relaxation value;
- Sol: number of instances that reached optimal solution during the maximum time of experiment;
- $t(s)$ : average CPU time in seconds (Total time).

As expected, the decrease in the  $GAP_{LP}$  average shows the efficiency of the strengthening set for each formulation. In particular, although reformulations (SPF-S1) and (SPF-S2) have the same linear relaxation, the strengthened set (SPF-S2) for (SPF) is tighter than (SPF-S1), and additionally, no higher computational cost is perceived. In fact, the (SPF-S2) decreases the computing time as compared to the SPF. Similarly, the average time required by the (GPF-S) solution is less than the time required by (GPF), in a significant number of instances 50-20-15 and 50-20-62 solved to optimality.

On the other hand, the results show that the (GPF) is able to solve only a few instances of small size with 15 bundles in one hour, which shows that the time required by (SPF) is better than the that of (GPF). Note that the number of constraints and variables generated by (SPF) is larger than the number of constraints of (GPF), which may impact the algorithm's processing time. Nevertheless, (GPF) also works for not necessarily non-decreasing functions while (SPF) requires a non-decreasing function.

As expected, the computational performance of formulation (GPF) is worse than (SPF1) and (SPF2). Although (SPF1) and (SPF2) reformulation has the same linear relaxation, the (SPF2) obtained a better GAP than (SPF1), and additionally, no higher computational cost is perceived. At the same time, (SPF2) presented the best value in most of the instances.

In addition, note that the number of constraints and variables generated in (GPF) is higher than in (SPF1) and (SPF2), which may impact the algorithms' processing time. However (GPF) can be used for any function price even if it is nonincreasing; hence it is useful when there are few buyers. Despite (SPF) being a model to represent step price functions, since purchase quantities are integer, (SPF) can represent any non-increasing function by considering unit length intervals.

Table 4: Comparison of GAP, RGAP and  $t(s)$  between the formulations GPF and SPF.  $|K| = 12$

Instance	SPF1				SPF2				GPF				SP1				SPF2				GPF				
	GAP	RGAP	$t(s)$	Sol	GAP	RGAP	$t(s)$	Sol	GAP	RGAP	$t(s)$	Sol	GAP	RGAP	$t(s)$	Sol	GAP	RGAP	$t(s)$	Sol	GAP	RGAP	$t(s)$	Sol	
100-62	0.337	13.524	1,949	82	0.016	13.516	1,412	98	7.345	14.435	3,600	-	300-62	0.052	0.998	553	91	0.020	0.995	291	98	0.565	1.023	1,962	53
100-93	0.261	9.627	1,966	78	0.019	9.621	1,151	99	9.224	11.157	3,600	-	300-93	9.346	12.191	3,563	4	8.790	12.051	3,556	4	11.527	12.169	3,600	-
100-124	0.254	8.768	2,294	79	0.066	8.756	1,690	95	8.598	10.109	3,600	-	300-124	8.345	10.928	3,588	2	10.171	12.256	3,536	7	11.066	11.422	3,600	-
100-155	0.214	7.842	2,476	85	0.020	7.835	1,867	99	7.757	9.005	3,600	-	300-155	10.002	11.782	3,604	-	11.247	12.918	3,580	4	11.036	11.318	3,600	-
100-207	0.693	6.769	3,150	63	0.021	6.740	2,331	99	6.665	7.701	3,600	-	300-207	10.874	11.950	3,585	2	9.523	11.828	3,419	21	10.377	10.568	3,601	-
150-62	0.579	8.977	2,444	56	0.116	8.949	1,539	91	6.211	9.363	3,600	-	500-62	0.000	0.002	1	100	0.000	0.002	1	100	0.000	0.002	3	100
150-93	1.130	9.586	2,907	53	0.119	9.538	2,059	95	10.779	11.929	3,600	-	500-93	13.037	15.074	3,582	2	13.178	14.606	3,585	1	12.471	12.838	3,601	-
150-124	2.829	8.917	3,262	32	1.076	8.732	2,817	64	9.022	10.162	3,600	-	500-124	15.037	15.686	3,603	-	13.570	14.437	3,602	-	11.252	11.514	3,602	-
150-155	3.447	7.816	3,349	20	1.516	7.704	3,002	56	8.177	9.038	3,600	-	500-155	14.436	15.136	3,603	-	14.389	15.261	3,585	2	10.964	11.151	3,655	-
150-207	4.177	7.382	3,374	14	3.743	7.419	3,064	31	8.607	9.184	3,600	-	500-207	14.512	15.162	3,595	1	14.085	14.977	3,571	3	10.209	10.343	3,691	-
200-62	0.210	5.466	1,887	67	0.064	5.463	920	93	3.361	5.713	3,486	4	1000-62	0.000	0.000	1	100	0.000	0.000	1	100	0.000	0.000	3	100
200-93	5.230	10.336	3,387	14	2.815	9.934	3,130	39	10.370	11.415	3,600	-	1000-93	13.346	14.907	3,581	2	12.495	14.258	3,582	2	13.336	13.633	3,602	-
200-124	5.234	9.228	3,499	10	5.347	9.732	3,232	27	9.588	10.277	3,600	-	1000-124	16.404	16.706	3,608	-	16.206	16.506	3,603	-	12.969	13.190	3,602	-
200-155	5.480	8.722	3,516	9	5.507	8.872	3,444	14	9.182	9.652	3,600	-	1000-155	16.287	16.528	3,607	-	17.078	17.321	3,603	-	13.449	13.626	3,600	-
200-207	5.485	7.866	3,562	5	5.191	8.282	3,310	23	9.094	9.391	3,600	-	1000-207	16.296	16.461	3,604	-	19.507	19.668	3,602	-	11.300	11.423	3,600	-

### 4.3. Evaluation of Benders decomposition

We studied three different ways to add the Benders cuts into the *Branch and Bound* (B&B) process: traditional version of *Branch and Cut* (B&C), where cuts are added in all nodes; reduced version of *Branch and Cut* (B&C2) where cuts are added only in the incumbent nodes (nodes with integer solution), and *Cut and Branch* (C&B) where the cuts are applied only in the root node, i.e., Benders decomposition is applied to the linear relaxation of the problem. Cuts are added as lazy constraints embedded in the B&B proceeding using the callback environment provided by

GUROBI. GUROBI default cuts were turned off to test the efficiency of our own cuts. The results of the computational experiments are displayed in Table 5. It is important to note that the methods we study incorporate the stabilization scheme. To construct the feasible solution of the stabilization scheme, we consider, given a fractional solution, that each customer selects the bundle with a fractional value closest to one. We note that the *C&B* obtained the best performance in most cases reducing the GAP and t(s), and solving more instances than its competitors optimally.

Table 5: Comparison of GAP and t(s) between of Benders Decomposition approach for SPF.  $|\mathcal{K}| = 12$

Instance	RSPF-CB			RSPF-BD2			RSPF-BD			Instance	RSPF-CB			RSPF-BD2			RSPF-BD		
	GAP	t(s)	Sol	GAP	t(s)	Sol	GAP	t(s)	Sol		GAP	t(s)	Sol	GAP	t(s)	Sol	GAP	t(s)	Sol
100-62	0.005	177	100	0.010	455	99	0.008	446	99	300-62	0.002	17	100	0.000	34	100	0.000	33	100
100-93	0.006	280	100	0.000	565	100	0.000	550	100	300-93	0.077	1,210	96	0.451	2,079	78	0.431	2,021	80
100-124	0.007	439	100	0.028	946	98	0.022	900	98	300-124	0.136	1,674	94	0.910	2,685	59	0.828	2,665	60
100-155	0.006	500	100	0.032	1,046	98	0.022	959	99	300-155	0.274	1,867	86	0.960	2,801	57	0.937	2,754	57
100-207	0.006	669	100	0.065	1,349	96	0.063	1,329	96	300-207	0.671	2,335	70	1.608	2,991	36	1.614	2,991	35
150-62	0.006	248	100	0.053	558	96	0.052	551	96	500-62	0.001	6	100	0.000	5	100	0.000	4	100
150-93	0.006	545	100	0.075	1,069	96	0.077	1,022	96	500-93	0.553	2,092	80	1.424	3,013	46	1.341	2,972	49
150-124	0.006	715	100	0.112	1,449	92	0.103	1,410	94	500-124	0.994	2,616	61	1.910	3,177	33	1.763	3,139	37
150-155	0.006	770	100	0.081	1,486	95	0.089	1,449	95	500-155	1.532	3,050	46	2.626	3,442	14	2.514	3,423	15
150-207	0.021	1,140	99	0.332	1,955	81	0.306	1,903	83	500-207	2.100	3,184	29	2.920	3,499	11	2.935	3,488	10
200-62	0.004	54	100	0.000	118	100	0.000	115	100	1000-62	0.000	17	100	0.000	7	100	0.000	7	100
200-93	0.007	865	100	0.158	1,621	90	0.164	1,573	91	1000-93	2.852	3,298	23	4.127	3,547	7	4.079	3,540	7
200-124	0.056	1,088	97	0.364	1,917	83	0.351	1,899	84	1000-124	3.138	3,432	16	4.177	3,565	4	4.017	3,569	3
200-155	0.081	1,332	95	0.543	2,281	70	0.511	2,239	73	1000-155	3.684	3,478	9	4.213	3,556	4	4.134	3,550	4
200-207	0.206	1,611	91	0.704	2,435	66	0.700	2,437	66	1000-207	3.381	3,492	9	4.339	3,555	5	4.281	3,560	5

The analysis shows no clear difference in results between B&C and B&C2. We can explain these results by counting the number of possible cuts generated in each method. Recall also that the cuts generated by benders are equivalent to strengthened constraint (feas-MP), then, the possible cuts are  $|\mathcal{K}| \times L_k$ . Once added the cuts in the root node, the possible number of cuts is reduced.

Figure 2 shows the mean gap and the percentage of optimally solved instances by instance size ( $|\mathcal{I}|, |\mathcal{B}|$ ) obtained for B&C and C&B. Note that instances of up to 1000 clients have been solved. We can observe that the higher the number of clients and bundles, the more complex the instances are, with the increase in bundles complicating the problem more than the increase in buyers. Still, the B&C and C&B methods optimally solve instances of up to 1000 customers and 62 bundles. Moreover, for most instance sizes, C&B can solve some percentage (except with 1000 customers and more than 155 bundles). Furthermore, C&B obtains the best gap in all instances, with a mean gap of less than 6% for instances within 1000 purchasers.

Next, we study the effect of incorporating a stabilization scheme and providing a warm start solution on the performance of the *C&B* and *B&C* methods. We propose a simple warm start, in which each customer chooses the bundle with the best reservation price. Subsequently, we prove a local change by removing customers that can improve the solution. In the first part, let us see how including stabilization affects these methods. The results shown in Figure 3 show that the use of stabilization in both methods does not reduce the number of explored nodes. However, Figure 4 shows that more solutions are solved optimally in the preset time (one hour) when stabilization is included. At the same time, Figure 5 shows that when a stabilization scheme was added, the methods solved a higher percentage of instances close to optimal, and for the remaining instances, the gap was reduced. Therefore, although including stabilization does not apparently decrease the number of explored nodes, it goes deeper into more efficient nodes and helps find better solutions in a shorter time.

It can also be observed that for both *C&B* and *B&C* methods, the methods worsen their overall performance if an initial solution is provided without stabilization. On the contrary, when a warm start is added to the stabilization scheme, it improves the performance of *B&C*. In contrast, for *C&B*, the opposite is true, i.e., it is not useful to combine stabilization with a warm start.

Finally, it can be concluded that the results show a clear advantage in the yield obtained by solving *C&B* over *B&C*,

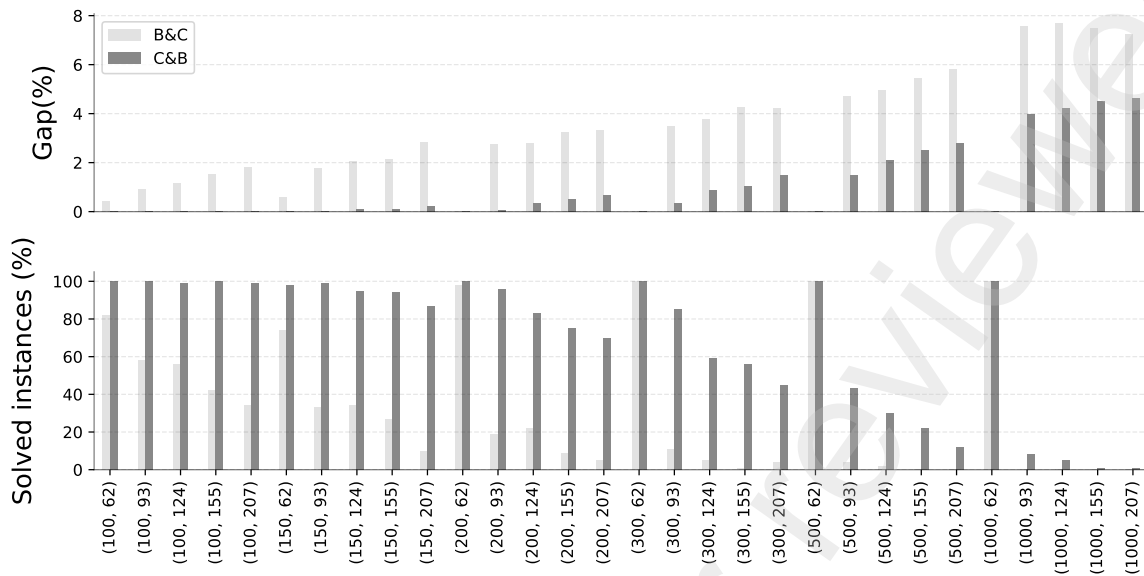


Figure 2: Barplot upper represents the mean gap of each method by size, and lower barplot represents the percentage of solved instances by size.

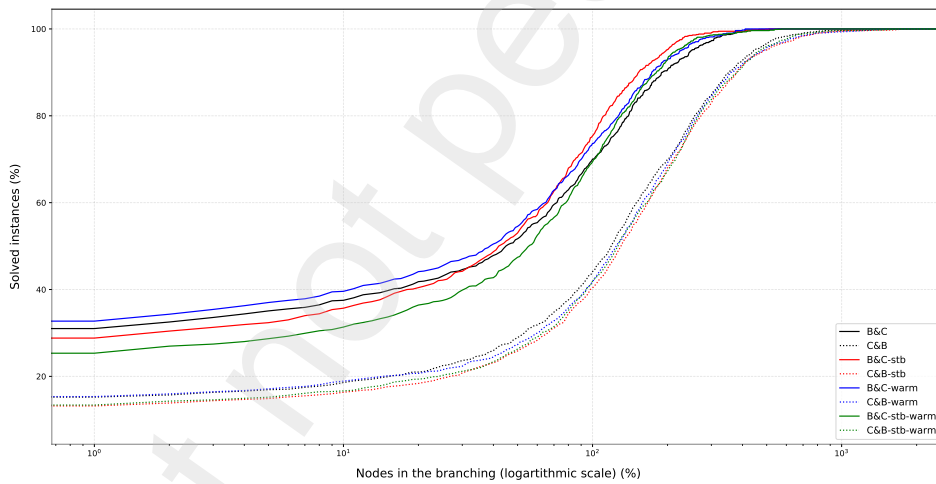


Figure 3: The percentage of solved instances with a number nodes less or equal to that corresponding abscissas is represented for each methods.

although, as Figure 3 shows, *C&B* must explore a larger number of nodes. On the other hand, Figure 4 shows that the percentage of instances solved by *C&B* before one hour is twice that of *B&B* (over 60% for the case of *C&B* and less than 30% for *B&C*). In the same sense, Figure 5 displays that 80% of the instances solved with *C&B* obtain a gap of less than 30% within one hour, while using *B&C*, only a 40% of instances reach the same percentage of the gap.

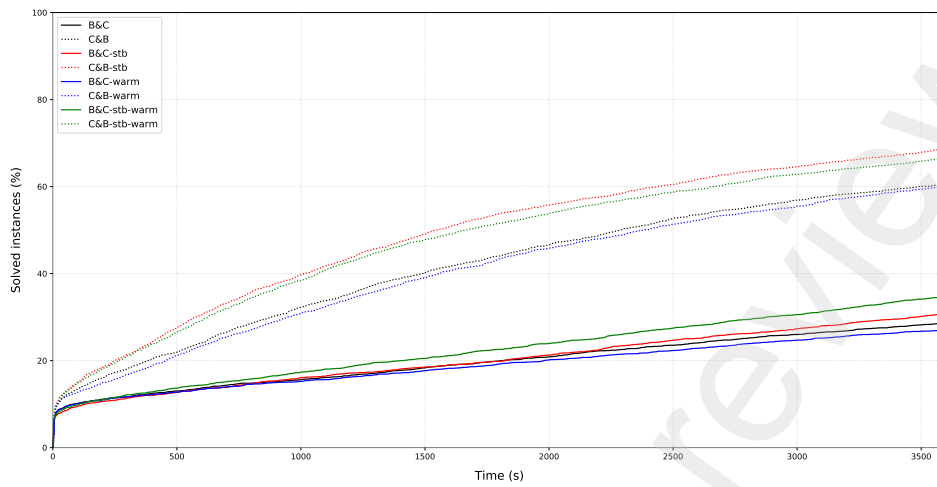


Figure 4: The percentage of solved instances in an time less or equal to that corresponding abscissas is represented for each methods.

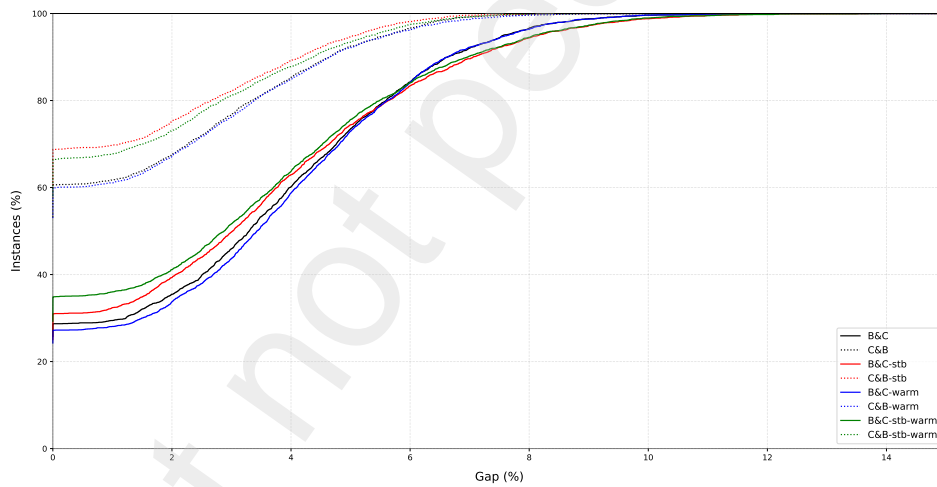


Figure 5: The percentage of instances with a gap less or equal to that corresponding abscissas is represented for each methods.

## 5. Conclusions

The constant expansion of the internet and the dramatic increase in its users worldwide have boosted the growth of digital marketplaces, which invites new forms of interaction between purchasers and merchants of all types. For example, sellers can offer wholesale prices to lower unit prices, but not necessarily because the buyers wish to or can purchase many items. However, purchasers with similar needs or desires can be grouped to take advantage of wholesale prices. Furthermore, this can be extended to the case in which purchasers require different sets of products, each bought at a wholesale price and then grouped into bundles according to different purchasers' segments' needs. In these circumstances, a coordinator is required who groups both products and purchasers in such a way as every

purchaser's segment gets the required bundle at a price that is better than if they would buy the products separately at a retail price. To do that, the coordinator needs to know every segment's reservation price for each possible bundle and, after the groups and bundles are formed, must assign the prices to every segment of buyers. The whole process maximizes the total reward. This problem is known in the literature as the coalition formation problem.

We solve a *Combinatorial Coalition Formation* problem, in which the products are offered in bundles, and each customer is willing to buy at most one bundle. Every purchaser has a reservation price for each bundle, which –at the same time– depends on whether the products are complementary or substitute. This decision problem may be reduced to a Set Packing Problem, which is an *NP-complete* problem; consequently, the optimization problem is *NP-hard*. The main reason for the problem's difficulty is that customers purchase bundles. Furthermore, the coalition value depends on the number of buyers purchasing any product, so both values must be found simultaneously.

From the sellers' point of view, wholesale pricing helps reduce inventory costs, saves marketing costs, and preserves or increases market power. Additionally, by offering certain bundles, the sellers can decrease the inventory levels of items that are difficult to sell by including them together with higher turnover products.

This paper proposes mixed-integer optimization models for the Coalition Formation Problem we just described. We show a general price function (GPF) formulation and a step price function (SPF) formulation. The first formulation uses a binary variable for every purchase size, discretizing the cost function and obtaining a value for the linear coalition. Since the step price function is widely used for wholesale prices, we introduce a multi-choice model for SPF. Given its structure, it is possible to apply a Benders decomposition approach.

The computational experience shows that, as expected, the (SPF) performs better than the (GPF) for the step formulation. Among other reasons, there are fewer variables in (SPF) than in (GPF). An interesting theoretical outcome suggests that the cut produced into the incumbent nodes by the Benders Decomposition approach becomes equal to a proposed valid inequality cost-SPF. Additionally, the experiments revealed a higher performance obtained using the Cut and Branch than the Branch and Cut procedure. This phenomenon arises from the fact that the cuts generated at the root nodes are not tight enough to reduce the, producing growth in the size of the problem solved in each node.

This work covered the buyers' side. Thus, it would be interesting to investigate this problem from the seller's perspective when faced group of buyers that can form coalitions. This problem can be modeled as a Stackelberg game in which the firm is the leader, and the customers are the followers; therefore, the firm proposes a price function depending on the purchase size; subsequently, with this information, the buyer groups purchase a particular bundle. Furthermore, the products bundled together in a bundle could also be part of the decision as another model extension.

Besides, it is possible to incorporate decisions depending on time, such as the buyers' arrival and the deadline for coalition formation, where the fairness and stability solutions are different. The outcomes will change whether we consider the choice online, which depends on how to arrive at the buyers, or offline, where all the information is revealed beforehand. Comparing both approaches is an attractive investigation for the future.

## Acknowledgements

The authors gratefully acknowledge the support by Grants FONDECYT 1190064, CONICYT PIA AFB180003 and INRIA Associated Team BIPLOS. Also, to CONICYT for a Doctorate fellowship for Sebastián Dávila Folio 21161328/2016. Martine Labbé has been partially supported by the Fonds de la Recherche Scientifique -FNRS under Grant PDR T0098.18. Powered@NLHPC: This research was partially supported by the supercomputing infrastructure of the NLHPC (ECM-02) of the Universidad de Chile. M.L.C. and H.X.

## References

- [1] A. Balakrishnan and S. C. Graves. A composite algorithm for a concave-cost network flow problem. *Networks*, 19(2):175–202, 1989.
- [2] P. Belotti, F. Malucelli, and L. Brunetta. Multicommodity network design with discrete node costs. *Networks: An International Journal*, 49(1):90–99, 2007.
- [3] F. Bistaffa, A. Farinelli, G. Chalkiadakis, and S. D. Ramchurn. A cooperative game-theoretic approach to the social ridesharing problem. *Artificial Intelligence*, 246:86–117, 2017.
- [4] F. Bistaffa, A. Farinelli, and S. Ramchurn. Sharing rides with friends: A coalition formation algorithm for ridesharing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.
- [5] E. Y. Bitar, E. Baeyens, P. P. Khargonekar, K. Poolla, and P. Varaiya. Optimal sharing of quantity risk for a coalition of wind power producers facing nodal prices. In *2012 American Control Conference (ACC)*, pages 4438–4445. IEEE, 2012.
- [6] L. Boongasame, P. Temdee, and F. Daneshgar. Forming buyer coalition scheme with connection of a coalition leader. *Journal of theoretical and applied electronic commerce research*, 7(1):111–122, 2012.
- [7] T. R. Christensen and M. Labbé. A branch-cut-and-price algorithm for the piecewise linear transportation problem. *European journal of operational research*, 245(3):645–655, 2015.
- [8] K. L. Croxton, B. Gendron, and T. L. Magnanti. A comparison of mixed-integer programming models for nonconvex piecewise linear cost minimization problems. *Management Science*, 49(9):1268–1273, 2003.
- [9] I. R. de Farias Jr, M. Zhao, and H. Zhao. A special ordered set approach for optimizing a discontinuous separable piecewise linear function. *Operations Research Letters*, 36(2):234–238, 2008.
- [10] Y. Fang, L. Sun, and Y. Gao. Bundle-pricing decision model for multiple products. *Procedia computer science*, 112:2147–2154, 2017.
- [11] A. Frangioni and B. Gendron. Piecewise linear cost network design. In *Network Design with Applications to Transportation and Logistics*, pages 167–185. Springer, 2021.
- [12] D. Gao, W. Jiao, and J. Zhang. Capacitated facility location problem with freight cost discount. In *2010 7th International Conference on Service Systems and Service Management*, pages 1–5. IEEE, 2010.
- [13] G. Greco and A. Guzzo. Constrained coalition formation on valuation structures: Formal framework, applications, and islands of tractability. *Artificial Intelligence*, 249:19–46, 2017.
- [14] L. He and T. R. Ioerger. Combining bundle search with buyer coalition formation in electronic markets: A distributed approach through explicit negotiation. In *Proceedings of the 6th international conference on Electronic commerce*, pages 95–104, 2004.
- [15] K. Holmberg. Solving the staircase cost facility location problem with decomposition and piecewise linearization. *European Journal of Operational Research*, 75(1):41–61, 1994.
- [16] K. Holmberg and J. Ling. A lagrangean heuristic for the facility location problem with staircase costs. *European journal of operational research*, 97(1):63–74, 1997.
- [17] M. Hyodo, T. Matsuo, and T. Ito. An optimal coalition formation among buyer agents based on a genetic algorithm. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 759–767. Springer, 2003.
- [18] A. Iwasaki, S. Ueda, N. Hashimoto, and M. Yokoo. Finding core for coalition structure utilizing dual solution. *Artificial Intelligence*, 222:49–66, 2015.
- [19] A. B. Keha. *A polyhedral study of nonconvex piecewise linear optimization*. Georgia Institute of Technology, 2003.
- [20] C. Li and K. Sycara. Algorithm for combinatorial coalition formation and payoff division in an electronic marketplace. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, pages 120–127, 2002.
- [21] C. Li, K. Sycara, and A. Scheller-Wolf. Combinatorial coalition formation for multi-item group-buying with heterogeneous customers. *Decision Support Systems*, 49(1):1–13, 2010.
- [22] H. Li and J. Jiang. An optimal combinational coalition formation based on genetic algorithm. In *Proceedings of the 1st International Conference on E-Business Intelligence (ICEBI2010)*, pages 159–165. Atlantis Press, 2010.
- [23] H. Li and F. Zhu. Information transparency, multihoming, and platform competition: A natural experiment in the daily deals market. *Management Science*, 67(7):4384–4407, 2021.
- [24] X. Liang, L. Ma, L. Xie, and H. Yan. The informational aspect of the group-buying mechanism. *European Journal of Operational Research*, 234(1):331–340, 2014.
- [25] S. Narabin and V. Boonjing. A location-price-based buyer coalition. In *2012 IEEE 12th International Conference on Computer and Information Technology*, pages 416–420. IEEE, 2012.
- [26] T. Rahwan, T. P. Michalak, M. Wooldridge, and N. R. Jennings. Coalition structure generation: A survey. *Artificial Intelligence*, 229:139–174, 2015.
- [27] F. S. Salman, R. Ravi, and J. N. Hooker. Solving the capacitated local access network design problem. *INFORMS Journal on Computing*, 20(2):243–254, 2008.
- [28] J. Shu, Z. Li, and W. Zhong. A production–transportation problem with piecewise linear cost structures. *IMA Journal of Management Mathematics*, 23(1):67–79, 2012.
- [29] A. Sukstrienwong. Forming buyer coalition with bundles of items by ant colony optimization. In *Proceedings of the 2011 international conference on Computers and computing*, pages 111–118, 2011.
- [30] A. Sukstrienwong. A price-based mechanism for online buyer coalition by genetic algorithms. *International Journal of Innovative Computing, Information and Control (IJICIC)*, 14(05):1653–1679, 2018.
- [31] L. Tran-Thanh, T.-D. Nguyen, T. Rahwan, A. Rogers, and N. R. Jennings. An efficient vector-based representation for coalitional games. In *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.
- [32] J. P. Vielma, S. Ahmed, and G. Nemhauser. Mixed-integer models for nonseparable piecewise-linear optimization: Unifying framework and extensions. *Operations research*, 58(2):303–315, 2010.
- [33] J. P. Vielma, A. B. Keha, and G. L. Nemhauser. Nonconvex, lower semicontinuous piecewise linear optimization. *Discrete Optimization*, 5(2):467–488, 2008.