Omega 91 (2020) 102021

Contents lists available at ScienceDirect

Omega

journal homepage: www.elsevier.com/locate/omega

Collaborative job scheduling in the wine bottling process☆

Franco Basso[a,*], Mario Guajardo[b], Mauricio Varas[c]

[a] Escuela de Ingeniería Industrial, Universidad Diego Portales, Santiago, Chile
[b] Department of Business and Management Science, NHH Norwegian School of Economics, Bergen, Norway
[c] Facultad de Ingeniería, Universidad del Desarrollo, Santiago, Chile

ARTICLE INFO

Article history:
Received 15 December 2017
Accepted 25 December 2018
Available online 27 December 2018

Keywords:
Horizontal collaboration
Scheduling
Cooperative game theory
Wine industry

ABSTRACT

This paper proposes a horizontal collaborative approach for the wine bottling scheduling problem. The opportunities for collaboration in this problem are due to the fact that many local wine producers are usually located around the same region and that bottling is a standard process. Collaboration among wineries is modeled as a cooperative game, whose characteristic function is derived from a mixed integer linear programming model. Real world instances of the problem are, however, unlikely to be solved to optimality due to its complex combinatorial structure and large dimension. This motivates the introduction of an approximated version of the original game, where the characteristic function is computed through a heuristic procedure. Unlike the exact game, the approximated game may violate the subadditivity property. Therefore, it turns relevant not only to find a stable cost allocation but also to find a coalition structure for selecting the best partition of the set of firms. We propose a maximum entropy methodology which can address these two problems simultaneously. Numerical experiments illustrate how this approach applies, and reveal that collaboration can have important positive effects in wine bottling scheduling decreasing delay by 33.4 to 56.9% when improvement heuristic solutions are used. In contrast to the exact game in which the grand coalition is always the best outcome, in the approximated game companies may be better forming smaller coalitions. We also devise a simple procedure to repair the characteristic function of the approximated game so that it recovers the subadditivity property.

1. Introduction

In horizontal collaboration, two or more entities that operate on the same level of the supply chain (e.g., they are competitors) join efforts to perform tasks together [52]. The collaboration usually leads to an overall better outcome compared to the outcomes the entities would achieve if they would act separately [18]. The improved outcome can be, for example, an increase in profits or a reduction in costs. Recent implementations and developments, such as the joint transportation planning between Tupperware and Procter and Gamble [46] and the exchange of timber among Swedish forestry companies [23], have proved the effectiveness of collaboration in practice. Accordingly, the interest for such collaborative approaches has received increasing attention in the management science literature.

Horizontal collaboration may play a particularly interesting role in industries with low margins, where any cost reduction could be crucial to assure profitability [22]. The wine industry is an example of such type of industries. Under high competition in the international markets, wine producers are often on the weaker side of the negotiation with large customers, such as government monopolies or large retail chains. According to the Wine Institute of California[1], 28,230 millions liters of wine were produced in 2014 by 57 different countries around the world. None of these countries reach more than 17% of the total production, and 16 of them produced more than 1%. To stay competitive and profitable in the markets, wineries must manage their production processes as efficiently as possible.

The best regions for grape growth are usually well delimited which implies that wineries are often quite close to each other. For instance, in Chile's Casablanca Valley there are 13 wineries with a maximum pairwise distance of 18 km. Transportation between the bottling lines of these wineries can be done in just a few minutes. While some tasks in the production process of wine are specific to every particular producer, some other tasks, such as packaging, are rather standard. Nowadays, some Chilean wineries use a non-structured outsourcing policy through third-party logistics

[1] www.wineinstitute.org.

☆ This manuscript was prossesed by Associate Editor Harks.

* Corresponding author.
E-mail addresses: franco.basso@udp.cl (F. Basso), mario.guajardo@nhh.no (M. Guajardo), mavaras@udd.cl (M. Varas).

https://doi.org/10.1016/j.omega.2018.12.010
0305-0483/

providers (3PL) to bottle the wine at the plants of other wineries when their capacity is not sufficient. The movement of the wine is done by trucks directly loaded with the wine. This procedure does not represent a major operational challenge. In this scenario, horizontal collaboration emerges as a feasible and appealing opportunity for local producers to reduce production costs and thus increase margins and gain competitive advantages. Moreover, initiatives such as the creation of *Wines of Chile*, an association that gathers many local producers to promote their wines in the international market and to foster research and development, denote the willingness of different companies to join efforts for improvements in the industry.

In this article, we study collaboration in the wine bottling scheduling problem. In this problem, bottling jobs must be assigned to different lines taking into account setup times, processing times and deadlines, in such a way that the total delay costs are minimized. When two or more firms collaborate, they can share their bottling lines, and an optimal schedule for all firms' jobs is attempted to be found. In both non-collaborative and collaborative cases, the scheduling bottling problem can be formulated as a mixed integer programming (MIP) model. We show that the optimal solution to the collaborative case is at least as good as the sum of the costs of the non-collaborative solutions. When such an optimal solution is known, a further problem is how to allocate the costs among firms such that none of them has incentives to deviate from the so-called grand coalition. We address this problem by traditional cost allocation methods, derived primarily from cooperative game theory [67]. In practice, however, the optimal solutions might not be easy to find, because of the combinatorial structure of the scheduling problem and the dimension of real-world instances. Implementable solutions are, therefore, commonly found by heuristic approaches. While these may conduce to good solutions in relatively short times, it turns less clear whether collaboration still provides the best solution or not. We study this problem using heuristics and numerical experiments. Our results reveal that collaboration among all companies is often outperformed by a partition of them into smaller sets, which is remarkable as most related literature has assumed that the grand coalition forms.

The contribution of this paper is twofold. First, we introduce cooperation to the bottling scheduling problem in the wine industry, and we show through numerical experiments that considerable savings can be achieved. Second, we tackle the quite unexplored area of approximated cooperative games (that is when the characteristic function is computed through a heuristic). As it is shown in this paper, the approximated game does not necessarily satisfy the subadditivity property. So it turns relevant first to find an alternative to recover this property and second to address the question of which coalitions provide the overall better outcome. For the first problem, we propose a simple procedure that repairs the approximated characteristic function in such a way that the resulting approximated game satisfies subadditivity. For the second one, we propose a novel cost allocation and coalition formation model for the approximated game which seeks equity within each coalition formed while simultaneously allocating costs to the firms. We prove that our method provides a unique solution which also minimizes the overall cost. While we focus on the wine bottling problem, these contributions are relevant in a broader scope. In fact, the wine bottling problem can be seen as a job scheduling problem that arises in many other industries. Likewise, the approximation of the characteristic function in a cooperative game is relevant not only in this collaborative job scheduling problem but also in many other challenging combinatorial problems where the optimal solution is rarely available, such as in collaborative vehicle routing and collaborative inventory routing. In addition, finding a coalition structure is an important problem when the grand coalition is inviable, for example, because of the managerial burden of coordinating the cooperation when the number of partners is too large.

The rest of the paper is organized as follows. In Section 2, we present some background and related literature. In Section 3, we formulate the bottling scheduling problem as a MIP model and also formulate its collaborative version as a cooperative game in both exact and approximated forms. In Section 4, we outline traditional cost allocation methods and propose a new method based on a maximization entropy principle which can also deal with the coalition formation problem. In Section 5, we use a numerical example to illustrate how previous models and methods apply and perform a numerical experiment to test the new method. Our concluding remarks are presented in Section 6.

2. Background

Wine production has roughly four stages: grape production, wine manufacturing, bottling/packaging, and distribution (Fig. 1). The grape production consists mainly of agricultural operations [9,38], such as plantation and harvest. The wine manufacturing includes all the operations needed to transform the grape into wine, such as fermentation and storage. This stage is usually under the strict control of the oenologist who is in charge to ensure the quality of the wine. In the packaging stage, processes such as bottle filling, corking, capsuling, and labeling are developed. The distribution stage is mainly transport related, but also the management of sales channels is included. A number of articles have focused on wine production and its supply chain, such as Adamo [1], Garcia et al. [25], Mac Cawley [44], Ting et al. [59], and Varsei and Polyakovskiy [64]. In particular, for the wine grape harvesting we refer the reader to Ferrer et al. [20], for the wine manufacturing stage to Cakici et al. [11], for the packaging stage to Berruto et al. [10], Basso and Varas [8] and Varas et al. [62], and for the distribution stage to Cholette [13]. In this paper, we focus on collaborative opportunities in the third stage of the process. Bottling and packaging products have been identified as tasks where collaboration among different firms can be useful not only from cost savings and service level perspectives but also from an environmental perspective [27].

During the bottling/packaging stage, wine is kept in intermediate tanks waiting to be bottled, labeled and finally packaged in jobs (Fig. 2). The bottling machines need lengthy non-added value setup times which are job and line dependent. Labeling usually takes place right after the bottling, but sometimes could be decoupled to tackle demand uncertainty using a postponement strategy [14]. Each job should be delivered to clients by its corresponding due date. Then, the main problem of this stage is to determine a job sequence for each production line such that the delays are minimized.

To the best of our knowledge, there are no previous works on collaborative approaches in the wine industry neither in this nor other stages of the production process. We seek to expand horizontal collaboration to this industry, by means of principles well established in the classic cooperative game theory literature [67] and emergent trends in collaborative logistics [4].

Although our work is primarily inspired by the wine industry, the methodology proposed in this paper is general enough to be expanded to other industries with similar production processes. In this respect, we note that as large wineries usually have several production lines and large amounts to produce, many products need to be scheduled on multiple non-equal parallel production lines. All the jobs take the same route in each line, and the job must be entirely bottled and labeled before continuing with the next one. Following the taxonomy of Graham et al. [28], the wine bottling problem studied in this paper can be classified as a $R/ST_{sd}/T_j$ scheduling problem, where R refers to the unrelated par-

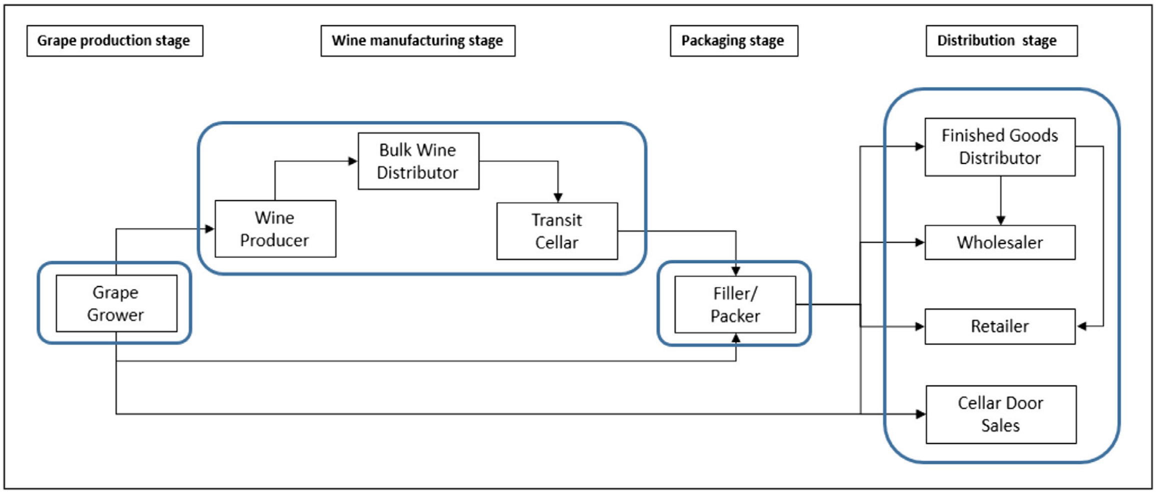

Fig. 1. The wine supply chain.

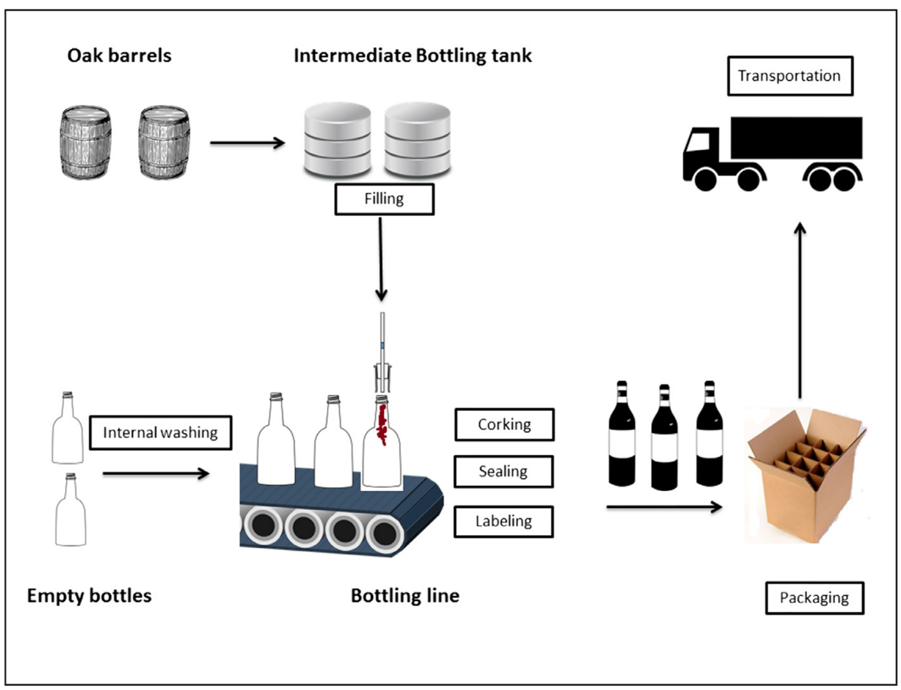

Fig. 2. Packaging activities.

allel machine shop problem, ST_{sd} to the sequence-dependent setup times, and T_j to the total tardiness/completion time criteria. Cooperation in the job scheduling problem assumes that the firms are able to share their parallel production lines. Thus, even if every firm would have a single machine, the cooperative scheme proposed in this paper involves solving a problem with multiple unrelated parallel machines [35,50,61]. This problem has many applications. These include, for example, shift-scheduling [15], operations at crossdock centres [40], assembly of electronic products [39], integrated-circuit packaging manufacturing systems [65], load balancing in project assignment [41], automobile gear manufacturing processes [26], air blast freezing in the food industry [12], and a negotiation scheme for scheduling problems in semiconductor manufacturing [48]. For a review on job scheduling and parallel machine problems we refer the reader to Allahverdi [2] and Mokotoff [45] respectively.

3. Collaborative bottling scheduling

In this section, we study the bottling problem in both the standard (single-company) and the collaborative cases. We start by describing the packaging problem faced by a winery, and we formulate the problem as a MIP. The model is then used to compute the characteristic function that defines the collaborative case as a cooperative game. We then define the approximated game in which the characteristic function is computed heuristically.

3.1. Description and formulation of the bottling problem

Most of the export-focused wineries sell their products *Free on Board* (FOB). This implies that the buyer is in charge of the transportation from the port of shipment to its destination. Thus, one of the primary concerns for the wineries is to dispatch the wine to meet the *stack* date, which is the time the ship sails. If the stack date is not met, customers will not have their shipment on time affecting the service level of the firm. In the wine industry, the job scheduling problem arises when the bottling and labeling packaging activities are requested by clients following a make-to-order policy. The jobs need to be allocated to the production lines which are highly automated. Between two consecutive jobs, lengthy non value-added setups must be performed. The resulting problem is highly combinatorial and, consequently, difficult to solve to optimality in reasonable computing times. Moreover, as cancellations and rush orders may occur, it is usually necessary to address the problem several times. For this purpose, Basso and Varas [8] formulated a MIP model and a heuristic solution approach following a greedy strategy. Their model includes multiple industry-specific constraints such as a non-wine acidification constraint (time windows resource constraints). In this paper, we assume that each winery could have multiple bottling lines and the processing time for each job is line-dependent. We assume setup times are sequence-dependent due to both, the time needed to clean the machine when passing from one type of wine to another, and the adjustment time of the machines required to change dry materials (bottles, corks, and labels) which are specific for each job. We assume that there are no time windows resource constraints and one job per customer only. These assumptions simplify the presentation of the problem and help us keep the focus of the paper on the opportunities for collaboration. At the same time, it also positions the problem within a broader context of job scheduling which, as seen in the background section, has many applications.

Sets

N	:	set of jobs (indexed by n).
L	:	set of bottling lines (indexed by l).

Parameters

$rc_{n,l} \in \mathbb{R}_+$	:	Processing time for job $n \in N$ in line $l \in L$.
$setup_{n,n',l}$	:	Setup time needed to pass from job $n \in N$ to job $n' \in N$ in line $l \in L$.
$t_n \in \mathbb{R}_+$	:	Deadline for job $n \in N$.
$M \in \mathbb{R}_+$	:	Constant large enough.

Variables

$g_n \in \mathbb{R}_+$	:	Initial processing time for job $n \in N$.
$o_n \in \mathbb{R}_+$	:	Final processing time for job $n \in N$.
$u_n \in \mathbb{R}_+$	:	Positive part of $o_n - t_n$.
$x_{n,l} \in \{0,1\}$	:	It takes value 1 if job $n \in N$ is processed in line $l \in L$.
$y_{n,n'} \in \{0,1\}$	:	It takes value 1 if job $n \in N$ is processed after job $n' \in N$ in the same line.
$yj_{n,n'} \in \{0,1\}$	:	It takes value 1 if job $n \in N$ is processed just after job $n' \in N$ in the same line.

Objective function

The objective is to minimize the total delay costs of the winery. For the sake of simplicity, we define this cost as equal to 1 per unit of time. Then, the objective function (1) is given by the following expression:

$$\min \sum_{n \in N} u_n \tag{1}$$

Constraints

$$u_n \geq o_n - t_n \quad \forall n \in N \tag{2}$$

$$\sum_{l \in L} x_{n,l} = 1 \quad \forall n \in N \tag{3}$$

$$o_n = g_n + \sum_{l \in L} rc_{n,l} \cdot x_{n,l} \quad \forall n \in N \tag{4}$$

$$o_{n'} + setup_{n',n,l} \leq g_n + M(1 - yj_{n,n'}) \quad \forall n, n' \in N, n \neq n', l \in L \tag{5}$$

$$y_{n,n'} \leq 1 + \frac{g_n - g_{n'}}{M+1} \quad \forall n, n' \in N, n \neq n' \tag{6}$$

$$y_{n,n'} \leq 1 - x_{n,l} + x_{n',l} \quad \forall n, n' \in N, n \neq n', l \in L \tag{7}$$

$$y_{n,n'} \geq x_{n,l} + x_{n',l} - 2 + \frac{g_n - g_{n'}}{M+1} \quad \forall n, n' \in N, n \neq n', l \in L \tag{8}$$

$$1 \geq x_{n,l} + x_{n',l} - y_{n,n'} - y_{n',n} \quad \forall n, n' \in N, n > n', \ l \in L \tag{9}$$

$$yj_{n',n} \leq y_{n',n} \quad \forall n, n' \in N, n \neq n' \tag{10}$$

$$yj_{n,n'} \leq 2 - y_{n,n''} - y_{n'',n'} \quad \forall n, n', n'' \in N, \ n \neq n' \neq n'' \neq n \tag{11}$$

$$y_{n,n'} \leq \sum_{n'' \in N} yj_{n,n''} \quad \forall n, n' \in N, n \neq n' \tag{12}$$

$$yj_{n,n} = 0 \quad \forall n \in N \tag{13}$$

Since objective function (1) minimizes the sum of variables u_n and constraints (2) impose a lower bound for these variables, both together define u_n as the non-negative part of $o_n - t_n$. Constraints (3) impose that each job must be assigned to exactly one line. Constraints (4) define o_n as the final processing time for job n. Constraints (5) state that job n can not start before the end of the previous job n' plus the respective setup time. Constraints (6)–(9) impose $y_{n,n'}$ to take the value 1 if and only if job n is processed after job n' in the same line. Constraints (10)–(13) impose $yj_{n,n'}$ to take value 1 if and only if job n is processed just after job n' in the same line. This model captures the essential features of the bottling scheduling problem and serves as a basis to define the collaborative bottling scheduling problem as a cooperative game with transferable utility. The definition of such a game requires the best outcomes that the different sets of companies can achieve by working together, which can be computed as the optimal objective values to different instances of the model. In what follows, we provide a formal definition of the collaborative problem preceded by the definitions of some concepts.

3.2. Cooperative game theory concepts

Early literature in cooperative game theory has dealt with the problem of how different agents may improve their overall results if instead of working separately they collaborate [67]. Such cooperative game theory principles have gained increasing attention in the literature on production and logistics because of a broad range of applications, such as in transportation [23] and inventory pooling [31]. The improvements of the collaborative solution over the non-collaborative solution in this context may involve cost savings, larger profits, higher service levels, or less environmental effects. Even if these improvements can be realized, [3] emphasize that the implementation of the collaborative solution requires to solve two key questions: how should potential savings be divided among a group of collaborating companies and how should collaborating groups be formed. Moreover, computing the outcomes of all the possible groups often may require solving a complex optimization problem a number of times that grows exponentially with the number of companies [19]. Our attention then focuses on addressing these challenges in the collaborative bottling scheduling problem. We proceed first by introducing some concepts. Let F be the set of all wineries. A non-empty subset of F is called a *coalition*. We define $\mathcal{K}$ the set of all possible coalitions. The coalitions represent a group of winery firms that cooperate by sharing their production lines. The set $F \in \mathcal{K}$ is called the *grand coalition*. The so-called *characteristic function* $C : \mathcal{K} \to \mathbb{R}_+$ computes the cost of each coalition. By convention, $C(\emptyset) = 0$. The cost $C(\{f\})$ correspond to the stand-alone cost of the winery firm $f \in F$, that is, it corresponds to the optimal bottling schedule cost for f. The pair (F, C) is called a *transferable cost game*. An important property of C is the *subadditivity* property, which states that $C(S \cup T) \leq C(S) + C(T),\ \forall S, T \in \mathcal{K}$. If this property is satisfied, a given coalition is at least as good as any of its partitions and, in consequence, the grand coalition minimizes the total costs.

3.3. The exact bottling scheduling game

To compute the characteristic function, we define the collaborative version of the MIP model presented in Section 3.1. For each winery $f \in F$, we define N_f as the set of jobs and L_f as the set of lines of winery f. We define a coalition $S \subset F$ as a subset of wineries which agreed to cooperate in the bottling and labeling process. We define the set N_S and L_S as the jobs and the production lines of the coalition S, respectively, as follows:

$$N_S = \bigcup_{f \in S} N_f, \ \ L_S = \bigcup_{f \in S} L_f,$$

The cost $C(S)$ of each coalition S is computed by solving the MIP model formulated in Section 3 taking $N = N_S$ and $L = L_S$. This optimization problem will be denoted $(\mathcal{O}_S)$. Once the model has been solved for all coalitions, the exact transferable cost game is defined by the pair (F, C).

Proposition 3.1. *The characteristic function C for the exact bottling cost transferable game satisfies the subadditivity property.*

The mathematical proof can be found in Appendix A. The statement is rather intuitive because the optimal solution of the problem without collaboration is a feasible solution for the problem with collaboration.

3.4. The approximated bottling scheduling game

Job scheduling problems are recognized in the literature as very hard to solve (see, e.g. [6] and [35]). As it was shown by Basso and Varas [8], the bottling scheduling problem is only solvable to optimality by the MIP approach when the number of jobs is quite small (≤ 15 jobs). Real world instances could easily exceed 100 jobs on a weekly basis. In these cases, an optimal schedule is practically impossible to compute. Even finding a feasible solution may turn an interminable task. A more viable way is to construct a solution based on a heuristic approach. Basso and Varas [8] present a greedy heuristic algorithm for a more complex version of the bottling scheduling problem presented in this paper. The logic of that greedy heuristic is to schedule jobs as soon as possible into the production line with the least processing time. For each discrete time and each job, the algorithm verifies if the partial solution is feasible. This is done through a verifier algorithm. If the partial solution is feasible, the heuristic continues with the next job. If not, the heuristic tries to schedule the job in the next period. Unfortunately, this procedure does not necessarily conduce to a feasible solution. To overcome this drawback, in this paper we propose an improvement to the greedy heuristic using Algorithm 1 below.

Algorithm 1 Bottling scheduling heuristic.

```
1: Define the parameter α > 1 as the increasing weight of the
   deadline in each iteration
2: while True do
3:     Apply the greedy algorithm presented in Basso and Varas
       (2017)
4:     if A solution is found then
5:         Break while
6:     end if
7:     t ← αt
8: end while
9: Compute the delays of the solution using the real deadlines
```

Algorithm 1 uses iteratively the greedy of Basso and Varas [8] and, if a solution is not found, the deadline vector t is extended by a weight α. Since there is no resource constraint, this algorithm must terminate with a feasible solution to the MIP model in a finite number of steps. We define $\widetilde{C}(S)$ as the cost of the coalition S computed through this heuristic procedure. Accordingly, we define the approximated transferable cost game by the pair $(F, \widetilde{C})$. Note that Proposition 3.1 is not necessarily true in the approximated game, as will be shown in our numerical results. Therefore, we also devise a procedure that repairs the approximated characteristic function in such a way that it recovers the subadditivity property. For this, we note that for any coalition S and a partition $\mathcal{P}$ of S, the solution composed by solving the job scheduling problem for each set in $\mathcal{P}$ is feasible in the job scheduling problem of S. We can, therefore, define an alternative characteristic function value $\widehat{C}(S)$ for the approximated game by taking the lowest value between the cost $\widetilde{C}(S)$ found by the heuristic and the lowest value of the cost among all partitions of S. We formalize this procedure in Algorithm 2 below.

Algorithm 2 Procedure to recover subadditivity.

```
1: for i ∈ {1, ..., |F|} do
2:     for S ∈ 𝒦 : |S| = i do
3:         Ĉ(S) ← C̃(S)
4:         for 𝒫 ∈ Π_S do
5:             Ĉ(S) ← min{Ĉ(S), Σ_{S̄∈𝒫} C̃(S̄)}
6:         end for
7:     end for
8: end for
```

In Algorithm 2, Π_S denotes the set of all partitions of S. The algorithm computes a characteristic function value $\widehat{C}(S)$ for each coalition S sequentially from lowest to highest cardinality. The value assigned to S is the best among the cost computed by the heuristic for coalition S and all the costs of the partitions $\mathcal{P}$ in Π_S. In this way, we assure that $\widehat{C}(S)$ is at least as good as the cost of any of its partitions, thus it satisfies subadditivity.

4. Cost allocation and coalition formation

4.1. Classical cost allocation methods

As emphasized in previous literature, an important question when firms collaborate is how to allocate the benefits of the collaboration among them [67]. Some of the challenges are to find allocations that provide benefits for all firms, create no incentives for firms to deviate from the collaboration, and guarantee that the benefits are shared in a *fair* way. Many allocation methods based on different fairness criteria have been developed in the literature. An early but good review can be found in [58] and a more recent one particularly focused on collaborative transportation can be found in [33]. The allocation methods usually assume as given the optimal value of the characteristic function for each possible coalition, and then compute an allocation assuming the grand coalition is formed. In what follows, we first introduce some of these methods and, in a second approach, we develop a new method that relaxes those assumptions.

Let γ_f be the cost allocated to the winery f. A fair allocation is commonly required to at least fulfill the *efficiency* and *rationality* conditions below.

$$\sum_{f\in F}\gamma_f = C(F) \tag{14}$$

$$\sum_{f\in S}\gamma_f \le C(S) \quad \forall S\in\mathcal{K} \tag{15}$$

The efficiency condition (14) states that all the cost is split among the members of the grand coalition, and the rationality conditions (15) provide that for all subsets of players there are no incentives to deviate from the grand coalition. The set of allocations that satisfy efficiency and rationality is called the *core* of the game. Among the many cost allocation methods proposed in the literature, the most used are the proportional methods, the Shapley value [56], and the nucleolus [54]. Other methods have been motivated in specific applications of collaborative logistics, such as the Equal profit method [23]. We present these four methods below.

Proportional methods. The proportional methods are the simplest. They assume that the cost of the coalition is split proportionally among its members following some pre-defined rule. Thus, to each winery it is allocated a cost $\gamma_f = \omega_f \cdot C(F)$, where $\omega_f\in[0,1]\,\forall f\in F$ and $\sum_{f\in F}\omega_f = 1$. The ω_f weight can be defined according to different criteria. A common one is to use the stand-alone costs, i.e., $\omega_f = \frac{C(\{f\})}{\sum_{f'\in F}C(\{f'\})}$.

Shapley values. This method allocates to each player f a cost according to the following formula:

$$\gamma_f = \sum_{S\subseteq F:f\in S}\left[\frac{(|F|-|S|)!(|S|-1)!}{|F|!}\right]\cdot[C(S)-C(S\setminus\{f\})] \quad \forall f\in F \tag{16}$$

Nucleolus. Define the excess of coalition S at γ in game (F, C) as $\varepsilon(\gamma, C, S) = C(S) - \sum_{f\in S}\gamma_f$. The excess of a coalition S at an allocation γ can be interpreted as a measure of satisfaction of the coalition with this allocation. The larger the excess of S, the more satisfied coalition S is, in the sense that it achieves larger savings. For a game (F, C), define the excess vector at γ as $e(\gamma, C) = (\varepsilon(\gamma, C, S_1), \dots, \varepsilon(\gamma, C, S_p))$, where the sets S_i represent the coalitions in $\mathcal{K}\setminus\{F\}$ and $p = 2^{|F|}-1$. We define $\theta(e(\gamma, C))$ as the vector that results from arranging the components of the excess vector in nondecreasing order. A vector $y\in\mathbb{R}^p$ is lexicographically greater than $\bar{y}$ (written $y\succeq\bar{y}$) if either $y=\bar{y}$ or there exists $h\in\{1,\dots,p\}$ such that $y_h > \bar{y}_h$ and $y_i = \bar{y}_i\ \forall i<h$. The nucleolus $\mathcal{F}$ of the cost sharing game (F, C) can be defined as $\mathcal{F} = \{\gamma\in\mathcal{X} : \theta(e(\gamma, C)) \succeq \theta(e(y, C))\ \forall y\in\mathcal{X}\}$, where $\mathcal{X}$ is the set of γ satisfying efficiency and the so called individual rationality property, defined as $\gamma_f \le C(\{f\})\,\forall f\in F$. The elements of $\mathcal{X}$ are called imputations. Thus, the nucleolus is the set of imputations that lexicographically maximizes the excess vector. If instead of $\mathcal{X}$, the definition above uses $\mathcal{X}^0$ (defined as the set of payoff vectors satisfying efficiency condition) we call it the pre-nucleolus. For games with a non-empty core, the pre-nucleolus coincides with the nucleolus. Appealing well-known features of the nucleolus is that it is unique and that it always belongs to the core whenever this is non-empty. Its computation is a bit more complex than applying a formula as for the Shapley method. Several algorithms have been proposed in the literature (see, e.g., [24], [29]).

Equal profit method (EPM). Suppose the cost allocated to player f is γ_f. Then, its relative savings compared to the stand-alone cost is $\frac{C(\{f\})-\gamma_f}{C(\{f\})}$. Thus, the difference between the relative savings of players f and f' is $\frac{\gamma_{f'}}{C(\{f'\})} - \frac{\gamma_f}{C(\{f\})}$. The EPM proposes to find an allocation in the core such that the maximum of the differences between relative savings is minimized. Such allocation can be found by solving the following linear programming model.

$$\min\ \Delta \tag{17}$$

$$\text{s.t.}\qquad \Delta \ge \frac{\gamma_{f'}}{C(\{f'\})} - \frac{\gamma_f}{C(\{f\})} \quad \forall\ f', f\in\ F \tag{18}$$

$$\sum_{f\in S}\gamma_f \le C(S) \quad \forall\ S\subset F \tag{19}$$

$$\sum_{f\in F}\gamma_f = C(F) \tag{20}$$

$$\gamma_f \ge 0 \quad \forall\ f\in F \tag{21}$$

Objective function (17) minimizes Δ and constraints (18) impose Δ as an upper bound on all pairwise differences between savings of the players. Thus, (17) and (18) together provide Δ equals the maximum of these differences. Constraints (19) and (20) impose rationality and efficiency, respectively. Then, if a solution to the model is obtained, it is an allocation in the core. Note this solution is not necessarily unique [17], a condition fulfilled by our method (see Section 4). If the core is empty, this model is infeasible.

4.2. The entropy method

Besides testing the methods above, our paper seeks to expand cooperative game theory applications to cases in which the characteristic function is computed heuristically which might, therefore, alter its properties. This is important in the bottling scheduling problem since due to its complexity in practice the implemented solution comes from a heuristic rather than an exact optimization approach. Contrary to the optimal approach, the characteristic function $\widetilde{C}$ does not necessarily satisfy the subadditivity property. Literature also suggests that cooperation in logistics is restricted to contain just a few partners [43], thus even when subadditivity holds it turns relevant to not only address the cost allocation but also the coalition formation problem. Our attention then turns to determine which coalitions should be formed from a min cost perspective. This problem relates to the cooperative game theory concept of *coalition structure* [5], that is, a collection of disjoint sets of players whose union is the overall set of players. This problem can be seen as a set partitioning problem which looks for the minimum cost partition and a cost allocation simultaneously [32] or in two separate subproblems [30]. In what follows, we formulate a model which finds the best partition and a cost allocation simultaneously, according to a maximum entropy criterion.

Sets

F	:	set of wineries (indexed by f).
$\mathcal{K}$	:	set of all possible coalitions (indexed by k).

Parameters

$\widetilde{C}_k \in \mathbb{R}_+$	:	Approximated cost of coalition k.
$\alpha_{fk} \in \{0, 1\}$	:	It takes value 1 if winery f belongs to coalition k.

Variables

$z_k \in \{0, 1\}$	:	It takes value 1 if coalition k is formed, and zero otherwise.
$\gamma_{fk} \in \mathbb{R}_+$	:	Cost allocated to winery f in coalition k.
$\rho_{fk} \in [0, 1]$	:	Proportion of cost allocated to winery f in coalition k.

The following mixed integer non-linear optimization problem seeks for a structure such that all coalitions are stable. The objective is to maximize for each formed coalition the Shannon's measure of entropy [55], which can be viewed as a proxy of equity.

$$\max -\sum_{k\in\mathcal{K}}\sum_{f\in F}\rho_{fk}\cdot\ln(\rho_{fk}) \tag{22}$$

$$\sum_{f\in F}\left(\alpha_{fk}\cdot\sum_{\bar{k}\in\mathcal{K}}\gamma_{f\bar{k}}\right)\leq\widetilde{C}_k\quad\forall k\in\mathcal{K} \tag{23}$$

$$\sum_{f\in F}\alpha_{fk}\cdot\gamma_{fk}=\widetilde{C}_k\cdot z_k\quad\forall k\in\mathcal{K} \tag{24}$$

$$\sum_{k\in\mathcal{K}}\alpha_{fk}\cdot z_k=1\quad\forall f\in F \tag{25}$$

$$\gamma_{fk}=\rho_{fk}\cdot\widetilde{C}_k\quad\forall f\in F,k\in\mathcal{K} \tag{26}$$

$$\sum_{f\in F}\rho_{fk}\leq z_k\quad\forall k\in\mathcal{K} \tag{27}$$

$$\sum_{f\in F}(1-\alpha_{fk})\cdot\rho_{fk}\leq 1-z_k\quad\forall k\in\mathcal{K} \tag{28}$$

$$\gamma_{fk}\geq 0, z_k\in\{0,1\}, \rho_{fk}\in[0,1]\qquad\forall f\in F,k\in\mathcal{K} \tag{29}$$

Constraints (23) state the strong rationality condition. Constraints (24) state the efficiency condition for all formed coalitions. Constraints (25) state that each winery must be assigned to exactly one coalition. Constraints (26) establish the relationship between cost allocation and the proportion of cost allocated. Constraints (27) state that cost splitting can be developed only within a formed coalition. Constraints (28) are logical relationships which, together with (24) and (26), provide that the cost allocation is performed only among the members of a formed coalition. Constraints (29) establish the nature of variables γ and z, and bounds on ρ.

Without constraints, the objective function (22) is optimized when all the wineries received the same percentage of the cost of the coalition [[55], pp. 11]. If this is not possible due to stability constraints, our model seeks to make the members of each coalition as equals as possible while satisfying rationality. This follows an egalitarian principle which has caught particular interest in recent years, especially in domains of cooperation where the characteristic function is affected [53]. Our equity method does not take into account the stand-alone cost of each winery of the coalition as, for example, the EPM does. This is done on purpose because for real instances the stand-alone cost is highly dependent on the heuristic approach to compute the characteristic function. This could imply some wineries to have an approximated cost extremely high compared to the optimal cost, affecting the outcome of the cost allocation method.

We end this section with the following two propositions about the proposed model.

Proposition 4.1. *The coalition structure given by the entropy method optimization problem minimizes the sum of overall costs.*

Proof. Let us suppose that a partition $\mathcal{Z}\subseteq\mathcal{K}$ of F is the optimal coalition structure given by the entropy method optimization problem. That is $z_k=1\ \forall k\in\mathcal{Z}$ and $z_k=0\ \forall k\in\mathcal{K}\setminus\mathcal{Z}$ and $\sum_{k\in\mathcal{Z}}\alpha_{fk}=1\ \forall f\in F$. By contradiction suppose that this partition does not minimize the overall cost. That implies there exist a partition $\overline{\mathcal{Z}}$ such that $\sum_{k\in\mathcal{Z}}\widetilde{C}_k>\sum_{k\in\overline{\mathcal{Z}}}\widetilde{C}_k$. The left-hand side in (24) can be replaced by $\sum_{k\in\mathcal{Z}}\widetilde{C}_k=\sum_{k\in\mathcal{Z}}\sum_{f\in F}\alpha_{fk}\gamma_{fk}=\sum_{k\in\mathcal{Z}}\sum_{f\in F}\gamma_{fk}$ (*). The last equality is valid because of equation (28) we have $\gamma_{fk}=0\ \forall f\in F,k\in K:\alpha_{fk}=0$. From (23), it holds that $\sum_{k\in\overline{\mathcal{Z}}}\widetilde{C}_k\geq\sum_{k\in\overline{\mathcal{Z}}}[\sum_{f\in F}\left(\alpha_{fk}\cdot\sum_{\bar{k}\in\mathcal{K}}\gamma_{f\bar{k}}\right)]=\sum_{k\in\overline{\mathcal{Z}}}\sum_{f\in F}\left(\alpha_{fk}\cdot\sum_{\bar{k}\in\mathcal{Z}}\gamma_{f\bar{k}}\right)=\sum_{f\in F}\sum_{\bar{k}\in\mathcal{Z}}[\gamma_{f\bar{k}}\sum_{k\in\overline{\mathcal{Z}}}\alpha_{fk}]$. Since the coalitions of $\overline{\mathcal{Z}}$ are a partition of F then for a fixed f we have that $\sum_{k\in\overline{\mathcal{Z}}}\alpha_{fk}=1$ so $\sum_{k\in\overline{\mathcal{Z}}}\widetilde{C}_k\geq\sum_{f\in F}\sum_{\bar{k}\in\mathcal{Z}}\gamma_{f\bar{k}}$ which it is a contradiction with (*).

The result in Proposition 4.1 is important to assure that the coalition structure achieves all the potential savings of the collaboration. Otherwise, there could be opportunities for a subset of players to look for other structures which would render them with more savings. □

Proposition 4.2. *Given a feasible coalition structure z_* the cost allocated to each winery of the formed coalitions under the entropy method γ_* is unique.*

Proof. Let $\mathcal{Z}\subseteq\mathcal{K}$ be a feasible partition of F. Considering both $\rho_{fk'}=0\ \forall f\in F,k'\in\mathcal{K}\setminus\mathcal{Z}$ and $\rho_{fk''}=0\ \forall f\in F,k''\in\mathcal{K}:\alpha_{fk''}=0$, the reduced feasible (cost proportion) allocation set $D(\mathcal{Z})$ is given by the intersection of the following two compact and convex sets:

$$D_1=\{\rho_{fk}\in\mathbb{R}^{|F|\times|\mathcal{Z}|}\mid 0\leq\rho_{fk}\leq 1,\sum_{f\in F}\rho_{fk}=1\ \forall k\in\mathcal{Z}\}$$

$$D_2=\{\rho_{fk}\in\mathbb{R}^{|F|\times|\mathcal{Z}|}\mid 0\leq\rho_{fk}\leq 1,\sum_{f\in F}\sum_{\hat{k}\in\mathcal{Z}}\alpha_{fk}\rho_{f\hat{k}}\widetilde{C}_{\hat{k}}\leq\widetilde{C}_k\ \forall k\in\mathcal{K}\}$$

It can be shown that $H(\rho)=-\sum_{f\in F}\sum_{k\in\overline{\mathcal{K}}}\rho_{f\bar{k}}\cdot\ln(\rho_{f\bar{k}})$ is strictly concave on D_1. Then $H(\rho)$ is strictly concave on $D(\mathcal{Z})=\cap_{j\in\{1,2\}}D_j\subseteq D_1$, which is also both compact and convex. Recall that on a convex set a strictly concave function has at most one global maximum. Now suppose that $D(\mathcal{Z})$ is not empty. Since $D(\mathcal{Z})$ is compact and $H(\rho)$ is continuous, $H(\rho)$ achieves a maximum on $D(\mathcal{Z})$. Therefore, the set of maximizers is a singleton $\{\rho_*\}$, and by (26), the allocation of cost γ_* must be unique.

The result of Proposition 4.2 is remarkable because it avoids the ambiguity of having two or more different solutions and thus prevents an arbitrary choice [17]. □

4.3. Addressing priority issues

In Subsection 4.2 we propose a novel cost allocation and coalition formation model. Our procedure seeks to allocate the costs as

Table 1
Characteristic function.

S	{1}	{2}	{3}	{4}	{1,2}	{1, 3}	{1, 4}	{2, 3}	{2, 4}	{3, 4}	{1, 2, 3}	{1, 2, 4}	{1, 3, 4}	{2, 3, 4}	{1, 2, 3, 4}
$C(S)$	3	7	11	7	10	13	10	15	12	18	17	15	19	21	24

Table 2
Cost for each firm obtained by different allocation methods.

Firm	Proportional	Shapley	Nucleolus	EPM
1	2.6	2.7	3	3
2	6	5.3	5	5
3	9.4	9.5	9	9
4	6	6.5	7	7

equal as possible among the wineries satisfying stability and rationality constraints. That is, all the wineries are treated equally. Nevertheless, depending on the negotiation some priorities could be given to some wineries depending on criteria such as size, the volume produced, operational costs and so on. Let us define a parameter $q_f > 0$ representing the priority of the winery $f \in F$. Lower values of q_f implies a higher priority of the winery $f \in F$, thus, a lower cost allocation if possible. Following the minimum cross-entropy idea [57] the model of Subsection 4.2 could be extended by changing the objective function by:

$$\max -\sum_{k\in\mathcal{K}}\sum_{f\in F}\rho_{fk}\cdot\ln\left(\frac{\rho_{fk}}{q_{fk}}\right)$$

where $q_{fk} = \dfrac{q_f}{\sum_{f'\in F}\alpha_{f'k}q_{f'}}$ corresponds to the priority weight of winery f in coalition k. It can be easily shown that if there are no constraints, the proportion of cost allocated to winery f in coalition k is q_{fk}. We conclude this section pointing out that both Propositions 4.1 and 4.2 also hold in this case with priorities. First, note that the objective function is not used in the proof of Proposition 4.1. Thus, the proposition also holds in this case. Second, the objective function with priorities remains strictly concave on $D(\mathcal{Z})$ and, therefore, Proposition 4.2 also holds in this case.

5. Numerical results

In this section, we perform some numerical experiments applying all models and methods from the previous sections. First, in a detailed example, we compute the optimal solution for the exact game and the corresponding allocations by the proportional method, Shapley value, Nucleolus, and EPM. The example is useful to illustrate how these concepts apply to the collaborative bottling scheduling problem. We also apply the heuristic and repairing procedure to this example, which is helpful to illustrate how the subadditivity property can be missed and recovered when instead of the exact approach the characteristic function values are just an approximation of the optimal values. We then perform experiments in additional randomly generated instances.

We have coded all models and methods in AMPL. For the linear formulations, we use the solver CPLEX 12.8.0.0. For the non-linear entropy model, we use the solver Artelis Knitro 10.3.0. The runs have been performed on a PC with an Intel(R) Core(TM) i7-7700HQ CPU@ 2.80 GHz processor and 16 GB RAM, each of them finishing in less than 20 minutes.

5.1. Illustrative example

We illustrate the application of all models and methods in a numerical example with four players. The data for one player comes from one of the three largest Chilean wineries in terms of export volume.

We took information of one day in which long jobs are carried out at this winery, and then we randomly generated data following the same structure for the other three players in the example. Each player has three jobs and one production line with the same characteristics (homogeneous production lines). Thus, setup and processing times are the same for all the wineries. The deadlines for each job of the three other players were randomly generated by a discrete uniform random variable in an interval [1,8]. The data is presented in Appendix B. Fig. 3 shows the resulting optimal schedule for the stand-alone cases and the grand coalition. The number in parentheses indicates the delay of each job. The total time saved is 4 units which correspond to a relative saving of 16.67%. The characteristic function for each coalition is shown in Table 1, whereas Table 2 shows the cost allocated to each winery by the four cost allocation methods defined for the exact game.

The characteristic function $\tilde{C}$ computed by the greedy heuristic for the same illustrative example is shown in Table 3. Note that $\tilde{C}(\{1,2,3,4\}) > \tilde{C}(\{1,4\}) + \tilde{C}(\{2,3\})$. Thus, if the grand coalition would be formed, players would have incentives to break it, forming {1, 4} and {2, 3}. Therefore, the grand coalition is not stable. This is a counter-example that proves the subadditivity property does not necessarily hold in the approximated version of the game. The entropy model for the approximated characteristic function $\tilde{C}$ provides as result precisely the coalitions {1, 4} and {2, 3}, together with cost allocation $\gamma_1 = 6, \gamma_4 = 8$ and $\gamma_2 = 9, \gamma_3 = 13$.

We also applied the entropy method (Table 4) for the repaired characteristic function $\hat{C}$ defined by Algorithm 2 (the characteristic function values modified by this algorithm in comparison to Table 3 appear underlined). In this case, the model provides as result the same coalitions than before, namely, {1, 4} and {2, 3} with the same allocated costs. This is an interesting result because by construction $\hat{C}$ satisfied the subadditivity property, so one could think that the grand coalition should form. As expected by Proposition 4.1, the grand coalition and the partition {1, 4} and {2, 3} have the same overall cost, namely, 36. Nevertheless, the partition {1, 4} and {2, 3} provides more equity according to the entropy objective function. This last solution has an objective function of 1.360 compared to 1.348 of the grand coalition.

5.2. Further experimental results

We have randomly generated $R = 100$ instances similar to the illustrative case but with non-homogenous production lines. Each instance has 4 wineries with 3 jobs and 1 bottling line each. We set $M = 240$ and the value of parameters t_n, $rc_{n,l}$, $setup_{n,n',l}$ are generated by a discrete uniform random variable in an interval [a, b], where a=1 for all of them, and b is equal to 8, 4, and 3, respectively. Since 15 different coalitions can be formed from the set of 4 wineries, the 100 instances involve solving 1500 scheduling problems.

The main findings of our experimental results are the following. Compared to the heuristical stand-alone situation, the delay costs decrease by 33.4% on average (95% percent confidence interval [30.9%, 35.9%]). This average value was computed by the for-

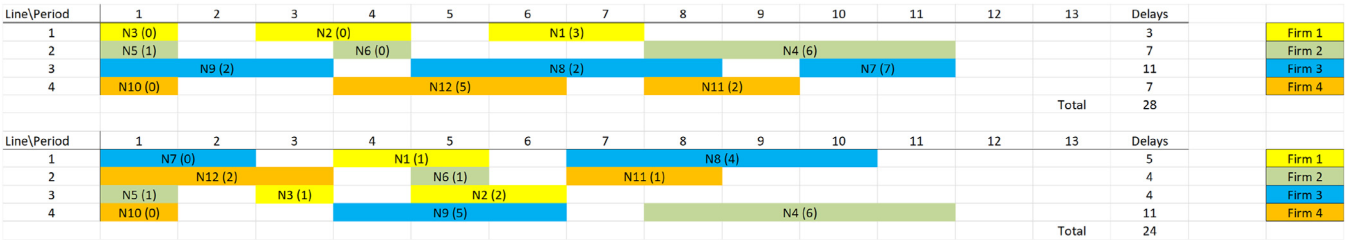

Fig. 3. Illustrative results for the stand-alone versus grand coalition solution.

Table 3
Characteristic function using Algorithm 1 (with $\alpha = 1.7$).

S	{1}	{2}	{3}	{4}	{1,2}	{1, 3}	{1, 4}	{2, 3}	{2, 4}	{3, 4}	{1, 2, 3}	{1, 2, 4}	{1, 3, 4}	{2, 3, 4}	{1, 2, 3, 4}
$\widetilde{C}(S)$	6	15	17	11	15	29	14	22	21	33	28	42	50	42	92

Table 4
Characteristic function using Algorithm 1 (with $\alpha = 1.7$) modified as to satisfy subadditivity by Algorithm 2.

S	{1}	{2}	{3}	{4}	{1,2}	{1, 3}	{1, 4}	{2, 3}	{2, 4}	{3, 4}	{1, 2, 3}	{1, 2, 4}	{1, 3, 4}	{2, 3, 4}	{1, 2, 3, 4}
$\widehat{C}(S)$	6	15	17	11	15	<u>23</u>	14	22	21	<u>28</u>	28	<u>26</u>	<u>31</u>	<u>33</u>	<u>36</u>

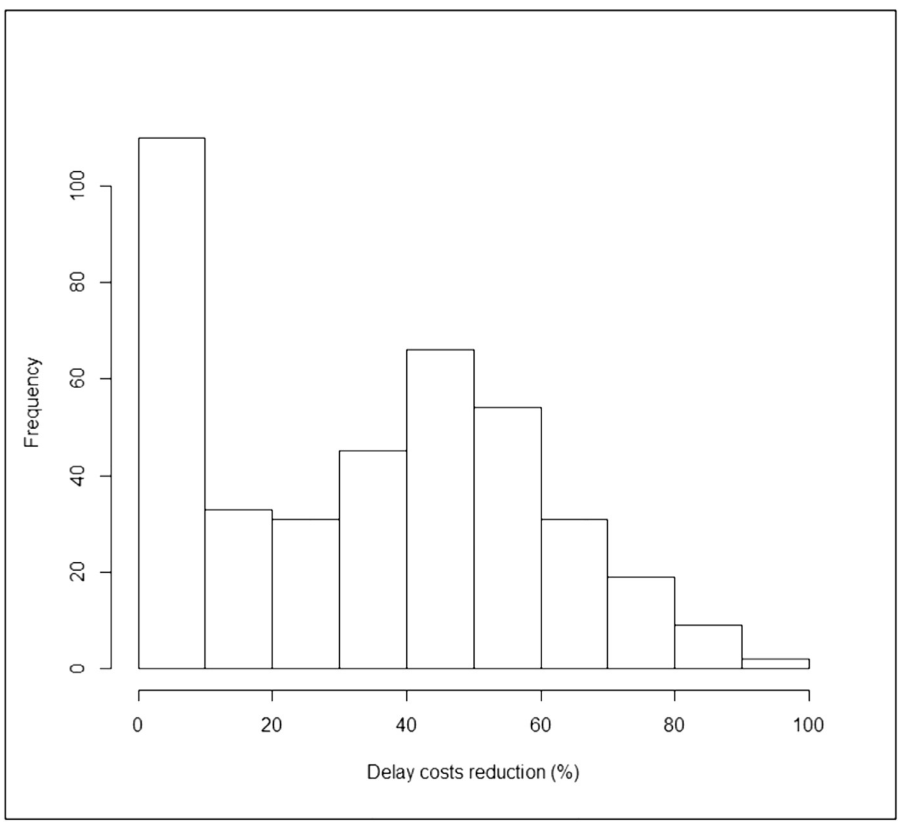

Fig. 4. Cost reduction histogram.

mula:

$$\frac{1}{R \cdot |F|} \sum_{r=1}^{R} \sum_{f \in F} \frac{\widetilde{C}_r(\{f\}) - \gamma_{f,r}}{\widetilde{C}_r(\{f\})} \tag{30}$$

where $\gamma_{f,\,r}$ is the cost allocated by the entropy method to the firm f in the instance r and $\widetilde{C}_r(\{f\})$ is the approximated characteristic function for instance r computed by the heuristic. The dispersion of the cost reduction is shown in the Fig. 4.

Table 5 shows the number of times each coalition is formed and the percentage that this number represents over the overall sum of these numbers. Table 6 shows the number of coalitions formed per instance, expressed as a percentage over the total number of instances. This reveals that the grand coalition forms in 39% of the instances, while the full non-collaborative situation never happens. The most preferred coalition structures include two coalitions.

Table 7 displays some important outcomes for the entropy method depending on the coalition size. The *Number of Wineries* row corresponds to the number of wineries belonging to a coalition of a given size. For example, 86 out of the 400 wineries belong to a coalition of size 2. The *Mean Proportion of the Cost* row corre-

Table 5
Coalition formation frequency.

S	{1}	{2}	{3}	{4}	{1,2}	{1, 3}	{1, 4}	{2, 3}	{2, 4}	{3, 4}	{1, 2, 3}	{1, 2, 4}	{1, 3, 4}	{2, 3, 4}	{1, 2, 3, 4}	Total
Frequency	11	16	12	11	6	7	7	7	6	10	9	11	10	10	39	172
Percentage (%)	6.4	9.3	7.0	6.4	3.5	4.1	4.1	4.1	3.5	5.8	5.2	6.4	5.8	5.8	22.6	100

Table 6
Number of coalitions per instance.

Number of coalitions per instance	1	2	3	4
Frequency (%)	39.0	54.0	7.0	0.0

Table 7
Cost information per winery in function of the coalition size.

Coalition size	1	2	3	4
Number of wineries	50	86	108	156
Mean proportion of the cost (%)	100.00	50.00	33.32	25.00
Mean optimal cost	7.86	9.16	9.52	10.21
Mean heuristical cost	9.12	12.00	12.70	13.14
Mean allocated cost	9.12	7.51	7.39	7.23
Heuristical cost reduction (%)	0	34.3	38.8	39.9

Table 9
Number of coalitions per instance for the entropy method applied to $\hat{C}$.

Number of coalitions per instance	1	2	3	4
Frequency (%)	85.0	15.0	0.0	0.0

Table 10
Cost information per winery in function of the coalition size or the entropy method applied to $\hat{C}$.

Coalition size	1	2	3	4
Number of wineries	1	56	3	340
Mean proportion of the cost (%)	100.00	50.00	33.33	25.00
Mean optimal cost	21.00	8.80	7.33	9.61
Mean heuristical cost	22.00	12.40	10.00	12.41
Mean allocated cost	22.00	8.20	4.30	7.60
Heuristical cost reduction (%)	0.00	33.87	57.00	38.76

sponds to the average percentage of the total cost of the coalition allocated to each winery. The *Mean Optimal Cost* row corresponds to the average optimal stand-alone cost computed for each winery by the MIP model described in Section 3.1. The *Mean Heuristical Cost* row corresponds to the average cost allocated to each winery by the heuristic described in Section 3.4. The *Mean Allocated Cost* row corresponds to the average cost allocated to each winery by the entropy method. Finally, the *Heuristical Cost Reduction* row corresponds to the percentage of cost reduction of the entropy method compared with the heuristical stand-alone situation, which represents the improvement achieved by the wineries due to the collaboration assuming that they schedule their jobs using the heuristic presented in this paper.

Conclusions can be drawn from Table 7. The heuristical cost reduction per winery increases as the coalition size increases. This is an expected result since if larger coalitions are feasible for the entropy method, the potential benefits of collaboration increase. When collaboration occurs (coalition size greater than 1), the average cost allocated to each winery in the collaborative environment is less than the optimal stand-alone cost. Moreover, in this situation the average cost reduction compared to the optimal stand-alone cost is 10.44%. This is an appealing result since in real situations the optimum is not even computable. As it is shown in row *Mean Proportion of the Cost*, on average the cost of each coalition is equally split among its members which is one of the main aims of the entropy method.

We also applied the entropy method for the repaired characteristic function $\hat{C}$. The results are shown in Tables 8, 9 and 10. The most interesting finding is that even though $\hat{C}$ satisfies the subadditivity property, the grand coalition does not form in all cases. In fact, in the 15% of the cases, the grand coalition is outperformed by a structure composed of two coalitions, in which, it is possible to split the costs more equitably.

5.3. Homogeneous production lines

The benefits of collaboration in the wine bottling job scheduling problem depend on the homogeneity of the data, for example, in the differences among the production lines. More specifically, if the values of the parameters $setup_{n,n',l}$ and $rc_{n,\,l}$ are highly line dependent then the impact of collaboration increases because it is possible to reduce delay costs by accommodating jobs in other more appropriate lines.

In this subsection, we analyze the limitations of the proposed collaborative scheme by analyzing the worst case scenario, that is, when all the wineries have identical production lines. To address this situation, we conducted $R = 10$ new instances similar to the illustrative case. As the previous subsection, each instance has 4 wineries with 3 jobs and 1 identical bottling line each. We set $M = 240$ and the value of parameters t_n, $rc_{n,\,l}$, $setup_{n,n',l}$ are generated by a discrete uniform random variable in an interval $[a,\, b]$, where a=1 for all of them, and b is equal to 8, 4, and 3, respectively. As opposed to Subsection 5.2, we impose that the processing times and setup times are not line dependent, that is, $rc_{n,l} = rc_{n,l'}$ and $setup_{n,n',l} = setup_{n,n',l'}$ $\forall l, l' \in L,\ \ \forall n, n' \in N$. Using equation (30), we compute the mean percentage cost reduction of the entropy method compared with the heuristical stand-alone situation for this new data set instance. In this situation, we obtain that delay costs decrease by 9.6%.

We remark, however, that an identical production line environment is unlikely to occur in practice. Job processing times are line dependent because production lines are usually specialized in some packaging format (e.g., 0.75 L or 1.5 L bottles; agglomerated or synthetic cork). As [44] asserts, the number of bottling lines can be over 10, and they can differ in their capacity and the types of products that can be bottled. Setup times are highly job, and line dependent since dry materials (bottles, corks, and labels) need to

Table 8
Coalition formation frequency for the entropy method applied to $\hat{C}$.

S	{1}	{2}	{3}	{4}	{1,2}	{1, 3}	{1, 4}	{2, 3}	{2, 4}	{3, 4}	{1, 2, 3}	{1, 2, 4}	{1, 3, 4}	{2, 3, 4}	{1, 2, 3, 4}	Total
Frequency	0	0	0	1	6	5	3	3	5	6	1	0	0	0	85	115
Percentage (%)	0.0	0.0	0 0.0	0.9	5.2	4.3	2.6	2.6	4.3	5.2	0.9	0.0	0.0	0.0	73.9	100

Table 11
Computational times for larger instances (in secs).

Firms \ Jobs	5	10	20	40	80	160
4	2.86	36.13	351.14	5,462.38	72,358.80	> 86, 400.00
5	10.49	154.29	1,984.07	23,186.17	> 86, 400.00	> 86, 400.00
6	56.15	258.39	6,242.56	> 86, 400.00	> 86, 400.00	> 86, 400.00
7	324.19	1,895.84	19,839.58	> 86, 400.00	> 86, 400.00	> 86, 400.00

be changed and prepared depending on the sequence. For instance, if two consecutive jobs have the same cork, the setup time will be less than if they have to change it. The same happens with the bottles and the labels. Those changes can be made faster or slower depending on the machine.

Finally, although our focus has been to analyze cooperation taking as given the characteristic function values computed by the heuristic rather than exploring its performance, we report on the computational times to find these values in instances with different numbers of players and jobs in Table 11.

When setting a time limit of one day (86,400 seconds), the heuristic can find solutions in instances with up to 80 jobs, when the number of players is four as in all our previous experiments. Increasing the number of players up to seven harms the possibilities to find solutions quickly for instances above 20 jobs.

5.4. Heuristic improvement

Depending on the context and instances, the performance of heuristic approaches could lead to better or worse solutions. In this section, we analyze what happens to our procedure when we improve the heuristic output through a metaheuristic. To do so, we implement a simplified version of the local branching (LB) scheme of Fischetti and Lodi [21], which has proved to be useful in a variety of applications (see, e.g. Rodríguez-Martín and Salazar-González [51]). This is in the spirit of well-known local search metaheuristics. For a description of local search strategies, see Pirlot [47]. Other general-purpose MIP heuristics using related approaches can be found in Lokketangen and Glover [42] and Hansen et al. [34]. Our local search utilizes a general MIP solver to explore the neighborhoods obtained through the introduction of invalid linear inequalities called local branching cuts. Specifically, we used the following cuts to reduce the solution space:

$$\sum_{l\in LS,\ n\in NS:\hat{x}_{n,l}=0} x_{n,l} + \sum_{l\in LS,\ n\in NS:\hat{x}_{n,l}=1} (1-x_{n,l}) + \sum_{n,m\in NS:\hat{y}_{n,m}=0} y_{n,m} + \sum_{n,m\in NS:\hat{y}_{n,m}=1} (1-y_{n,m}) \le K_s \tag{31}$$

where both $\hat{x}_{n,l}$ and $\hat{y}_{n,m}$ define a feasible solution, and K_s denotes the neighborhood size. The left hand side of (31) represents the Hamming distance between (x, y) and $(\hat{x}, \hat{y})$, which is denoted by $\delta_{(\hat{x},\hat{y})}(x, y)$. The simplified version of the Fischetti and Lodi [21] approach that we implement is described in Algorithm 3 below.

The general idea of Algorithm 3 is to explore the neighborhood of the best solution found so far using a state-of-the-art solver (CPLEX) within a time limit (at most T_{node}). If a better solution is found, it becomes the new incumbent solution and the neighborhood size and exploring time are reversed to their default values ($K_{\min}$ and $T_{\min}$, respectively). If the time limit is reached with no improved solution, a diversification step is applied to enlarge the current neighborhood size (in K_{step}) and to explore it for a larger time (at most T_{step}). The procedure is repeated until the total time limit ($T_{\max}$) or the maximum number of iterations allowed ($N_{\max}$) is exceeded.

We use the 1500 problems from the 100 instances described in Subsection 5.2 to analyze how the results behave for both the repaired and not repaired characteristic function when the greedy

Table 12
Coalition formation frequency for the entropy method.

Approach	S	{1}	{2}	{3}	{4}	{1,2}	{1, 3}	{1, 4}	{2, 3}	{2, 4}	{3, 4}	{1, 2, 3}	{1, 2, 4}	{1, 3, 4}	{2, 3, 4}	{1, 2, 3, 4}	Total
$(\tilde{C}, greedy + LB_1)$	Frequency	10	8	10	8	7	5	4	3	5	6	9	7	6	9	46	143
	Percentage (%)	7.0	5.6	7.0	5.6	4.8	3.5	2.8	2.1	3.5	4.2	6.3	4.8	4.2	6.3	32.2	100
$(\hat{C}, greedy + LB_1)$	Frequency	0	0	0	1	4	4	3	3	4	4	1	0	0	0	59	83
	Percentage (%)	0.0	0.0	0.0	1.2	4.8	4.8	3.6	3.6	4.8	4.8	1.2	0.0	0.0	0.0	71.1	100
$(\tilde{C}, greedy + LB_2)$	Frequency	4	1	3	1	7	0	1	1	0	7	1	3	1	4	53	87
	Percentage (%)	4.6	1.1	3.4	1.1	8.0	0.0	1.1	1.1	0.0	8.0	1.1	3.4	1.1	4.6	60.9	100
$(\hat{C}, greedy + LB_2)$	Frequency	0	0	0	0	3	0	2	2	0	3	0	0	0	0	56	66
	Percentage (%)	0.0	0.0	0.0	0.0	4.5	0	3.0	3.0	0	4.5	0	0	0	0	84.8	100

Algorithm 3 Local branching scheme.

Let $\hat{x}_{inc}, \hat{y}_{inc}, f_{inc} \leftarrow$ Algorithm 1 (Section 3.4)
Define $K_{\min}, K_{step}, T_{\min}, T_{step}, T_{\max}$ and $N_{\max}$ (user)
Let $T_{start} \leftarrow Time(), \ T_{node} \leftarrow T_{\min}, \ K_s \leftarrow K_{\min}, \ N_{iter} \leftarrow 0$ and $N_{stop} \leftarrow false$
Add $\delta_{(\hat{x}_{inc}, \hat{y}_{inc})}(x, y) \leq K_s$ to the collaborative bottling scheduling problem (Section 3.1). Let the constrained problem be denoted as $P(\hat{x}_{inc}, \hat{y}_{inc}, K_s)$.
while $Time() - T_{start} \leq T_{\max}$ **and** $N_{stop} = false$ **do**
  Let $\bar{x}_{cur}, \bar{y}_{cur}, f_{cur} \leftarrow$ Solve($P(\hat{x}_{inc}, \hat{y}_{inc}, K_s)$, CPLEX, T_{node})
  if $f_{curr} < f_{inc}$ **then**
    Let $\hat{x}_{inc} \leftarrow \bar{x}_{cur}, \hat{y}_{inc} \leftarrow \bar{y}_{cur}, f_{inc} \leftarrow f_{cur}$
    Let $K_s \leftarrow K_{\min}, T_{node} \leftarrow T_{\min}$
  else
    $K_s \leftarrow K_s + K_{step}, T_{node} \leftarrow T_{node} + T_{step}$
  end if
  Let $N_{iter} \leftarrow N_{iter} + 1$
  if $N_{iter} = N_{\max}$ **then**
    $N_{stop} \leftarrow true$
  end if
end while

Table 13
Number of coalitions as percentage of the total feasible instances.

Approach	1	2	3	4
$(\widetilde{C}, greedy + LB_1)$	50.5	45.1	4.4	0.0
$(\widehat{C}(S), greedy + LB_1)$	83.1	16.9	0.0	0.0
$(\widetilde{C}, greedy + LB_2)$	75.7	24.3	0.0	0.0
$(\widehat{C}(S), greedy + LB_2)$	91.8	8.2	0.0	0.0

heuristic is coupled with this local branching scheme. We analyze two settings which differ in the number of maximum neighborhood explorations: $K_{\min} = 2, \ K_{step} = 2, \ T_{\min} = 4$ (sec), $T_{step} = 2$ (sec), $T_{\max} = 15$ (sec), and $N_{\max} \in \{2, 10\}$. We refer as LB_1 when $N_{\max} = 2$, and LB_2 when $N_{\max} = 10$. We have also solved to optimality all these problems, by running the MIP model without time limit. It is noteworthy that the solution time to reach optimality can be prohibitively large, often in the order of several hours and some cases up to a couple of days. In contrast, the improved heuristic can find good quality solutions in much shorter times. In 62.0% of the instances, the optimality gap is less than 1%. Accounting for all the instances, the optimality gap is on average 7.4%. The computational times to arrive at this solutions is just a few seconds, as the time limits above have been set to at most 15 seconds.

As for the coalitions formed and the costs, a summary of the results is presented in Tables 12, 13 and 14. In the approach description, we denote first whether the entropy method is applied to the game with characteristic function value $\widetilde{C}$ or $\widehat{C}$, and second the maximum number of iterations used by the heuristic improvement (thus, for example, $(\widetilde{C}, greedy + LB_1)$ denotes the heuristic improvement on function $\widetilde{C}$ with maximum number of iterations equal to LB_1). With these four different settings for the 1500 problems, the improved heuristic has been run 6000 times. Because of the impossibility to satisfy the strong rationality constraints (23), some of the 100 instances turn infeasible. In the calculation of percentages, we take as a basis only the instances with feasible solutions.

Contrasting these results with those obtained in Subsection 5.2, the results in Tables 12 and 13 indicate that the improved heuristic conduces to a larger frequency in the formation of the grand coalition. This trend is stressed when using a better approximation of the characteristic function ($\widehat{C}$ instead of $\widetilde{C}$) and a larger maximum number of iterations (LB_2 instead of LB_1). These observations are reasonable, as when the solutions get closer to optimal, we would expect the grand coalition to achieve more savings than other structures. Note, however, that there are still many instances (ranging from 49.5 to 8.2%) where the coalition structure that forms differs from the grand coalition. This occurs because there are partitions that allow achieving as much savings as the grand coalition and, besides, they allow to find allocations with greater equity than the grand coalition.

A comparison of the values in rows *Mean Optimal Cost* and *Mean Heuristical Cost* of Table 14, shows that the heuristic improvements conduce to very close to optimal costs. More important, the savings due to collaboration (displayed in row *Heuristical Cost Reduction*) range from 37.50 to 60.84%, which are in general more significant than those obtained in Subsection 5.2. Overall, when opportunities for collaboration exist, the entropy method shows effectiveness in finding coalition structures and stable allocations that leave all players with significant savings with respect to their stand-alone costs and with no incentives to deviate from the collaboration.

6. Concluding remarks

We have studied collaborative job scheduling in the wine bottling problem, by combining methods of mixed integer programming and cooperative game theory. We have shown that the grand coalition is, theoretically, better than (or equal to) the non-collaborative solution and any other coalition structure. However, as shown in our numerical results, structures in which firms collaborate within smaller sets may behave better than the grand coalition when the solutions to the bottling scheduling problem are found through heuristics. Literature and real cases have shown that big coalitions are more likely to fail [16,22,43]. Our results may help to explain this discrepancy between theory and reality.

Based on a maximum entropy principle, a new method proposed in this paper allows to deal with both coalition formation and cost allocation problems simultaneously, in contrast to the most traditional methods that only cope with the latter. To our knowledge, this is the first article studying collaboration in wine production. Overall, we find that collaboration may be useful for wineries to reduce their costs and increase their service levels. This is not only supported by the theoretical foundation developed in Sections 3 and 4, but also for our numerical experiments in Section 5, where the decrease in delays averages from 33.4 to 56.9% when improvement heuristic solutions are used. Our work identifies the wine industry as especially suitable for such a collaborative approach to be implementable in practice, because wineries' facilities are often concentrated within nearby regions and some of the wine production processes are standard. Moreover, our proposed method takes into account the difficulties of generating a bottling schedule and includes several factors that belong to wine decision making such as setups highly sequence dependent. Given the high competition in international markets, finding ways to improve the margins and service levels can be crucial for the success of wine producers. In this line, our work provides the following managerial insights:

- Collaborating with different companies can help to reduce the costs in wine bottling. Thus, it turns relevant for managers to attempt getting agreements for cooperation with other wine producers.
- As the overall result by a group of companies in collaboration may be better than the sum of its separate parts, it is relevant for managers to include in their agreements a fair way to split the benefits of the cooperation. Our article has shown how different methods can deal with this problem, according to different notions of fairness.

Table 14
Cost information per winery in function of the coalition size or the entropy method.

	Coalition size	1	2	3	4
$(\widetilde{C}, greedy + LB_1)$	Number of wineries	36	60	84	184
	Mean proportion of the cost (%)	100.00	50.00	33.12	25.21
	Mean optimal cost	6.55	9.23	9.48	10.21
	Mean heuristical cost	6.66	9.53	9.88	10.43
	Mean allocated cost	6.66	5.50	5.60	5.40
	Heuristical cost reduction (%)	0.00	42.28	43.31	48.22
$(\widehat{C}(S), greedy + LB_1)$	Number of wineries	1	44	3	236
	Mean proportion of the cost (%)	100.00	50.00	33.33	25.02
	Mean optimal cost	21.00	9.00	7.33	9.72
	Mean heuristical cost	22.00	9.34	7.66	9.99
	Mean allocated cost	22.00	5.30	3.00	5.50
	Heuristical cost reduction (%)	0.00	43.25	60.84	49.94
$(\widetilde{C}, greedy + LB_2)$	Number of wineries	9	32	91	212
	Mean proportion of the cost (%)	100.00	49.87	33.32	24.98
	Mean optimal cost	4.22	8.41	11.26	10.21
	Mean heuristical cost	4.22	8.41	11.26	10.21
	Mean allocated cost	4.22	4.50	4.60	4.20
	Heuristical cost reduction (%)	0.00	46.49	59.15	58.86
$(\widehat{C}(S), greedy + LB_2)$	Number of wineries	0	20	0	224
	Mean proportion of the cost (%)	-	50.00	-	25.00
	Mean optimal cost	-	8.00	-	10.14
	Mean heuristical cost	-	8.00	-	10.14
	Mean allocated cost	-	5.00	-	4.19
	Heuristical cost reduction (%)	-	37.5	-	58.68

- Since managerial decisions in practice are often driven by heuristic rather than optimal approaches, some fundamental properties of collaboration can be broken. We have shown how the subadditivity property of a cooperative game can be affected when the characteristic function is an approximation instead of an exact solution to the underlying optimization problem. As we have done in this article, devising a repairing procedure to recover such properties or designing a method tailored to the approximated situation, may turn useful for managers to address this situation in practice.
- While collective initiatives attempt to gather efforts from many companies in the same industry (such as *Wines of Chile*), both practice and previous literature have indicated that cooperation usually involves just a few partners [7]. Thus, from an overall industry perspective, it becomes interesting to not only find ways to split the benefits of collaboration but also to identify groups of companies within an industry to form coalitions. We have addressed this as a coalition structure problem, using stability principles from cooperative game theory, that might become important for managers when selecting partners and keeping the incentives for all of them to stay in the coalition.

While we focused on the bottling scheduling problem, studying collaboration in other problems related to wine production remains an interesting challenge. One of these, for example, is how different wineries could join forces to negotiate better rates with the shipping carriers that transport the wine from their regions to the international markets. Recent approaches for this problem in other contexts have been provided in [66] and [60]. Although we neglected the transportation costs, due to the proximity of wineries within their region, in other cases one could introduce these costs into the analysis following approaches on collaborative transportation recently reviewed in [33]. Another problem where wineries have potential for collaboration is in their inventory management efforts, for example, by sharing capacity in a warehouse [49]. Wineries from the same region may also collaborate by advertising their wines together in other markets, for which the literature also offers a vast number of approaches [36,37,68]. From a practical point of view, in all of these problems it turns interesting to dig further in applied settings such as the case-based analytical modeling in the wine industry provided by [63] and [64].

Besides wineries, our work may expand to other industries and applications. In this respect, the maximum entropy method that we proposed for the approximated game in this article could be tested in other problems of coalition formation and cost allocation, such as collaborative vehicle routing and inventory sharing. Dealing with the approximate game turns particularly important when the number of firms grows because the number of coalitions grows exponentially. In problems where computing the optimal solution for a single instance is already complicated, solving such exponential number of instances is practically impossible. Thus, studying approximated versions of the cooperative game and designing methods for it may become essential to implement the collaboration. In our numerical results, when using the greedy heuristic, the grand coalition was the best solution in only 39% of the cases, even though the exact game was proved to be subadditive. This percentage grows as the heuristic is improved, reaching from 50.5 to 91.8% in our experiments with the local branching heuristic. Since most literature in collaborative logistics so far has focused on the allocation of costs assuming the grand coalition is formed, our results show the importance of further investigation in coalition structures where this assumption is broken.

Acknowledgments

We would like to sincerely thank the two reviewers and the editor for their valuable suggestions that allowed us to considerably improve a preliminary version of this work. We also gratefully acknowledge financial support from the Complex Engineering Systems Institute, ISCI (grant CONICYT FB0816).

Appendix A

Proof of Proposition 3.1. Let consider two coalitions $S, T \in \mathcal{K}$. By definition

$$N_{S \cup T} = \bigcup_{f \in S \cup T} N_f, \; L_{S \cup T} = \bigcup_{f \in S \cup T} L_f$$

This implies

$$N_S, N_T \subseteq N_{S\cup T}$$

$$L_S, L_T \subseteq L_{S\cup T}$$

Given an optimal vector (g^S, o^S, u^S, x^S, y^S, yj^S) for ($\mathcal{O}_S$) and an optimal vector (g^T, o^T, u^T, x^T, y^T, yj^T) for ($\mathcal{O}_T$) since $N_S \cap N_T = \emptyset, L_S \cap L_T = \emptyset$ we can obtain a feasible solution ($g^{S\cup T}$, $o^{S\cup T}$, $u^{S\cup T}$, $x^{S\cup T}$, $y^{S\cup T}$, $yj^{S\cup T}$) for ($\mathcal{O}_{S\cup T}$) by the following formula:

$$\begin{aligned}
g_n^{S\cup T} &= g_n^S \cdot \mathbb{1}_{N_S}(n) + g_n^T \cdot \mathbb{1}_{N_T}(n) \quad \forall n \in N_S \cup N_T \\
o_n^{S\cup T} &= o_n^S \cdot \mathbb{1}_{N_S}(n) + o_n^T \cdot \mathbb{1}_{N_T}(n) \quad \forall n \in N_S \cup N_T \\
u_n^{S\cup T} &= u_n^S \cdot \mathbb{1}_{N_S}(n) + u_n^T \cdot \mathbb{1}_{N_T}(n) \quad \forall n \in N_S \cup N_T \\
x_{n,l}^{S\cup T} &= x_{n,l}^S \cdot \mathbb{1}_{N_S}(n) \cdot \mathbb{1}_{L_S}(l) + x_{n,l}^T \cdot \mathbb{1}_{N_T}(n) \cdot \mathbb{1}_{L_T}(l) \\
&\quad \forall n \in N_S \cup N_T, l \in L_S \cup L_T \\
y_{n,n'}^{S\cup T} &= y_{n,n'}^S \cdot \mathbb{1}_{N_S}(n) \cdot \mathbb{1}_{N_S}(n') + y_{n,n'}^T \cdot \mathbb{1}_{N_T}(n) \cdot \mathbb{1}_{N_T}(n') \\
&\quad \forall n, n' \in N_S \cup N_T \\
yj_{n,n'}^{S\cup T} &= yj_{n,n'}^S \cdot \mathbb{1}_{N_S}(n) \cdot \mathbb{1}_{N_S}(n') + yj_{n,n'}^T \cdot \mathbb{1}_{N_T}(n) \cdot \mathbb{1}_{N_T}(n') \\
&\quad \forall n, n' \in N_S \cup N_T
\end{aligned}$$

Since ($g^{S\cup T}$, $o^{S\cup T}$, $u^{S\cup T}$, $x^{S\cup T}$, $y^{S\cup T}$, $yj^{S\cup T}$) it is a feasible solution for ($\mathcal{O}_{S\cup T}$) then

$$\begin{aligned}
C(S\cup T) &\le \sum_{n\in N_{S\cup T}} u_n^{S\cup T} \\
&= \sum_{n\in N_{S\cup T}} (u_n^S \cdot \mathbb{1}_{N_S}(n) + u_n^T \cdot \mathbb{1}_{N_T}(n)) \\
&= \sum_{n\in N_{S\cup T}} u_n^S \cdot \mathbb{1}_{N_S}(n) + \sum_{n\in N_{S\cup T}} u_n^T \cdot \mathbb{1}_{N_T}(n) \\
&= \sum_{n\in N_S} u_n^S + \sum_{n\in N_T} u_n^T \\
&= C(S) + C(T)
\end{aligned}$$

□

Appendix B. Illustrative example data

The following Tables B.1–B.3 show the data used in our illustrative example in Section 5.1. In this case, M was set at 240.

Table B.1
Sets.

Set	Cardinality
F	4
N_f	3
L_f	1

Table B.2
Job parameters.

(n)	(f)	t_n	$rc_{n,1}$	$rc_{n,2}$	$rc_{n,3}$	$rc_{n,4}$
1	1	5	2	2	2	2
2	1	5	2	2	2	2
3	1	3	1	1	1	1
4	2	6	4	4	4	4
5	2	1	1	1	1	1
6	2	5	1	1	1	1
7	3	5	2	2	2	2
8	3	7	4	4	4	4
9	3	2	3	3	3	3
10	4	2	1	1	1	1
11	4	8	2	2	2	2
12	4	2	3	3	3	3

Table B.3
Setups for line $i \in \{1, 2, 3, 4\}$.

$setup_{n,n',i}$	1	2	3	4	5	6	7	8	9	10	11	12
1	1	1	1	3	3	3	3	1	2	2	3	3
2	1	3	1	3	3	3	2	2	3	3	1	2
3	3	1	2	2	1	2	3	3	2	3	1	3
4	3	1	2	1	3	1	3	3	3	1	3	1
5	2	1	1	3	1	2	3	3	1	2	3	2
6	3	2	3	3	3	3	1	3	1	1	1	3
7	1	1	3	3	3	3	1	3	2	1	2	2
8	3	1	1	3	2	2	1	1	3	3	1	1
9	1	2	2	1	3	1	2	1	2	1	1	2
10	3	2	3	2	1	3	3	2	2	3	3	2
11	2	3	1	3	2	1	2	1	2	1	3	3
12	3	2	2	2	1	1	3	2	1	2	1	3

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.omega.2018.12.010.

References

[1] Adamo C. A global perspective of the wine supply chain: the case of Argentinean wineries and the US market Ph.D. thesis. Massachusetts Institute of Technology; 2004.
[2] Allahverdi A. The third comprehensive survey on scheduling problems with setup times/costs. Eur J Oper Res 2015;246(2):345–78.
[3] Audy J-F, D'Amours S, Rönnqvist M. An empirical study on coalition formation and cost/savings allocation. Int J Prod Econ 2012;136(1):13–27.
[4] Audy J-F, Lehoux N, D'Amours S, Rönnqvist M. A framework for an efficient implementation of logistics collaborations. Int Trans Operat Res 2012;19(5):633–57.
[5] Aumann RJ, Dreze JH. Cooperative games with coalition structures. Int J Game Theory 1974;3:217–37.
[6] Bai D, Tang M, Zhang Z-H, Santibanez-Gonzalez ED. Flow shop learning effect scheduling problem with release dates. Omega (Westport) 2018;78:21–38.
[7] Basso F, D'Amours S, Rönnqvist M, Weintraub A. A survey on obstacles and difficulties of practical implementation of horizontal collaboration in logistics. Int Trans Operat Res 2019;26(3):775–93.
[8] Basso F, Varas M. A mip formulation and a heuristic solution approach for the bottling scheduling problem in the wine industry. Comp Indust Eng 2017;105:136–45.
[9] Behzadi G, O'Sullivan MJ, Olsen TL, Zhang A. Agribusiness supply chain risk management: a review of quantitative decision models. Omega (Westport) 2018;79:21–42.
[10] Berruto R, Tortia C, Gay P. Wine bottling scheduling optimization. Trans-Am Soc AgricEng 2006;49(1):291.
[11] Cakici E, Jia J, Yu P, Mason SJ, Richard Cassady C, Pohl L, Lachowsky AJ. Cellar tank piping network analysis at e. & j. gallo winery. J Wine Res 2006;17(3):145–60.
[12] Chatavithee P, Piewthongngam K, Pathumnakul S. Scheduling a single machine with concurrent jobs for the frozen food industry. Comp Indust Eng 2015;90:158–66.
[13] Cholette S. A novel problem for a vintage technique: using mixed-integer programming to match wineries and distributors. Interfaces (Providence) 2007;37(3):231–9.
[14] Cholette S. Mitigating demand uncertainty across a winery's sales channels through postponement. Int J Prod Res 2009;47(13):3587–609.
[15] Ciancio C, Laganà D, Musmanno R, Santoro F. An integrated algorithm for shift scheduling problems for local public transport companies. Omega (Westport) 2018;75:139–53.
[16] Cruijssen F, Cools M, Dullaert W. Horizontal cooperation in logistics: opportunities and impediments. Transport Res Part E LogistTransport Rev 2007;43(2):129–42.
[17] Dahlberg J, Göthe-Lundgren M, Engevall S. A note on the nonuniqueness of the equal profit method. Appl Math Comput 2017;308:84–9.
[18] Defryn C, Sörensen K, Dullaert W. Integrating partner objectives in horizontal logistics optimisation models. Omega (Westport) 2019;82:1–12.
[19] Drechsel J, Kimms A. Computing core allocations in cooperative games with an application to cooperative procurement. Int J Prod Econ 2010;128(1):310–21.
[20] Ferrer J-C, Mac Cawley A, Maturana S, Toloza S, Vera J. An optimization approach for scheduling wine grape harvest operations. Int J Prod Econ 2008;112(2):985–99.
[21] Fischetti M, Lodi A. Local branching. Math Program 2003;98(1–3):23–47.
[22] Flisberg P, Frisk M, Rönnqvist M, Guajardo M. Potential savings and cost allocations for forest fuel transportation in sweden: a country-wide study. Energy 2015;85:353–65.
[23] Frisk M, Göthe-Lundgren M, Jörnsten K, Rönnqvist M. Cost allocation in collaborative forest transportation. Eur J Oper Res 2010;205(2):448–58.

[24] Fromen B. Reducing the number of linear programs needed for solving the nucleolus problem of *n*-person game theory. Eur J Oper Res 1997;98(3):626–36.
[25] Garcia FA, Marchetta MG, Camargo M, Morel L, Forradellas RQ. A framework for measuring logistics performance in the wine industry. Int J Prod Econ 2012;135(1):284–98.
[26] Gokhale R, Mathirajan M. Scheduling identical parallel machines with machine eligibility restrictions to minimize total weighted flowtime in automobile gear manufacturing. Int J Adv ManufactTechnol 2012;60(9–12):1099–110.
[27] Gonzalez-Torre PL, Adenso-Di B, Artiba H, et al. Environmental and reverse logistics policies in european bottling and packaging firms. Int J Prod Econ 2004;88(1):95–104.
[28] Graham RL, Lawler EL, Lenstra JK, Kan AR. Optimization and approximation in deterministic sequencing and scheduling: a survey. Annal Discrete Math 1979;5:287–326.
[29] Guajardo M, Jörnsten K. Common mistakes in computing the nucleolus. Eur J Oper Res 2015;241(3):931–5.
[30] Guajardo M, Jörnsten K, Rönnqvist M. Constructive and blocking power in collaborative transportation. OR Spect 2016;38(1):25–50.
[31] Guajardo M, Rönnqvist M. Cost allocation in inventory pools of spare parts with service-differentiated demand classes. Int J Prod Res 2015;53(1):220–37.
[32] Guajardo M, Rönnqvist M. Operations research models for coalition structure in collaborative logistics. Eur J Oper Res 2015;240(1):147–59.
[33] Guajardo M, Rönnqvist M. A review on cost allocation methods in collaborative transportation. Int Trans Operat Res 2016;23(3):371–92.
[34] Hansen P, Mladenović N, Urošević D. Variable neighborhood search and local branching. Comp Operat Res 2006;33(10):3034–45.
[35] Huo Y, Zhao H. Two machine scheduling subject to arbitrary machine availability constraint. Omega (Westport) 2018;76:128–36.
[36] Jørgensen S, Zaccour G. A survey of game-theoretic models of cooperative advertising. Eur J Oper Res 2014;237(1):1–14.
[37] Karray S. Cooperative promotions in the distribution channel. Omega (Westport) 2015;51:49–58.
[38] Kusumastuti RD, van Donk DP, Teunter R. Crop-related harvesting and processing planning: a review. Int J Prod Econ 2016;174:76–92.
[39] Li K, Sivakumar AI, Ganesan VK. Complexities and algorithms for synchronized scheduling of parallel machine assembly and air transportation in consumer electronics supply chain. Eur J Oper Res 2008;187(2):442–55.
[40] Li Y, Lim A, Rodrigues B. Crossdocking: jit scheduling with time windows. Journal of the Operational Research Society 2004:1342–51.
[41] Liang Z, Li Y, Lim A, Guo S. Load balancing in project assignment. Comp Operat Res 2010;37(12):2248–56.
[42] Lokketangen A, Glover F. Solving zero-one mixed integer programming problems using tabu search. Eur J Oper Res 1998;106(2–3):624–58.
[43] Lozano S, Moreno P, Adenso-Díaz B, Algaba E. Cooperative game theory approach to allocating benefits of horizontal cooperation. Eur J Oper Res 2013;229(2):444–52.
[44] Mac Cawley AF. The international wine supply chain: challenges from bottling to the glass Ph.D. thesis. Georgia Institute of Technology; 2014.
[45] Mokotoff E. Parallel machine scheduling problems: a survey. Asia-Pacific J Operat Res 2001;18(2):193.
[46] Muylaert K, Stofferis L. Driving sustainability through horizontal supply chain collaboration. In: CO3 Final Conference. CO3; 2014. Available in: http://www.co3--project.eu.
[47] Pirlot M. General local search methods. Eur J Oper Res 1996;92(3):493–511.
[48] Ramacher R, Mönch L. An automated negotiation approach to solve single machine scheduling problems with interfering job sets. Comp Indust Eng 2016;99:318–29.
[49] Rivera L, Sheffi Y, Knoppen D. Logistics clusters: the impact of further agglomeration, training and firm size on collaboration and value added services. Int J Prod Econ 2016;179:285–94.
[50] Rocha PL, Ravetti MG, Mateus GR, Pardalos PM. Exact algorithms for a scheduling problem with unrelated parallel machines and sequence and machine-dependent setup times. Comp Operat Res 2008;35(4):1250–64.
[51] Rodríguez-Martín I, Salazar-González JJ. A local branching heuristic for the capacitated fixed-charge network design problem. Comp Operat Res 2010;37(3):575–81.
[52] Roma P, Perrone G. Cooperation among competitors: a comparison of cost-sharing mechanisms. Int J Prod Econ 2016;180:172–82.
[53] Routledge R. Information, egalitarianism and the value. Operat Res Lett 2016;44(6):775–8.
[54] Schmeidler D. The nucleolus of a characteristic function game. SIAM J Appl Math 1969;17(6):1163–70.
[55] Shannon CE. A mathematical theory of communication. Bell Syst Tech J 1948;27:379–423.
[56] Shapley L. A value for n-person games. Annal Math Stud 1953;28:207–317.
[57] Shore J, Johnson R. Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy. IEEE Trans Inf Theory 1980;26(1):26–37.
[58] Tijs SH, Driessen TS. Game theory and cost allocation problems. Manage Sci 1986;32(8):1015–28.
[59] Ting S, Tse Y, Ho G, Chung S, Pang G. Mining logistics data to assure the quality in a sustainable food supply chain: a case in the red wine industry. Int J Prod Econ 2014;152:200–9.
[60] Tinoco SVP, Creemers S, Boute RN. Collaborative shipping under different cost-sharing agreements. Eur J Oper Res 2017;263(3):827–37.
[61] Tran TT, Araujo A, Beck JC. Decomposition methods for the parallel machine scheduling problem with setups. INFORMS J Comput 2016;28(1):83–95.
[62] Varas M, Maturana S, Cholette S, Cawley AM, Basso F. Assessing the benefits of labelling postponement in an export-focused winery. Int J Prod Res 2018;56(12):4132–51. doi:10.1080/00207543.2018.1431415.
[63] Varsei M, Christ K, Burritt R. Distributing wine globally: financial and environmental trade-offs. Int J Phys DistribLogist Manag 2017;47(5):410–28.
[64] Varsei M, Polyakovskiy S. Sustainable supply chain network design: a case of the wine industry in australia. Omega (Westport) 2017;66:236–47.
[65] Yang T. An evolutionary simulation–optimization approach in solving parallel-machine scheduling problems–a case study. Comp Indust Eng 2009;56(3):1126–36.
[66] Yilmaz O, Savasaneril S. Collaboration among small shippers in a transportation market. Eur J Oper Res 2012;218(2):408–15.
[67] Young HP. Cost allocation: methods, principles, applications. North Holland Publishing Co.; 1985.
[68] Zhou Y-W, Li J, Zhong Y. Cooperative advertising and ordering policies in a two-echelon supply chain with risk-averse agents. Omega (Westport) 2018;75:97–117.