



# Optimal coalition structures for probabilistically monotone partition function games

Shaheen Fatima<sup>1</sup> · Michael Wooldridge<sup>2</sup>

Accepted: 15 March 2022 / Published online: 19 April 2022  
© The Author(s) 2022

## Abstract

For cooperative games with externalities, the problem of optimally partitioning a set of players into disjoint exhaustive coalitions is called *coalition structure generation*, and is a fundamental computational problem in multi-agent systems. Coalition structure generation is, in general, computationally hard and a large body of work has therefore investigated the development of efficient solutions for this problem. However, the existing methods are mostly limited to deterministic environments. In this paper, we focus attention on uncertain environments. Specifically, we define *probabilistically monotone partition function games*, a subclass of the well-known *partition function games* in which we introduce uncertainty. We provide a constructive proof that an exact optimum can be found using a greedy approach, present an algorithm for finding an optimum, and analyze its time complexity.

**Keywords** Coalition formation · Cooperative games · Partition function games · Optimization · Uncertainty

## 1 Introduction

A key open problem in multi-agent systems research is how to organise agents into disjoint teams so as to maximise some overall welfare measure. This *coalition structure generation* (CSG) problem is in general computationally complex: NP-hard even under quite modest assumptions. For this reason, there have been many studies directed at finding easy instances of the problem (see [2, 3, 17, 35, 37] for some examples and [32] for a detailed survey).

Coalition/cooperative game theory provides a conventional framework for modelling CSG problems [9, 34, 40]. A coalition game is defined by a pair comprised of a set of agents and a function that maps coalitions to values. A widely studied subclass of coalition games are *characteristic function games* (CFGs). For a CFG, the value of a coalition

---

✉ Shaheen Fatima  
S.S.Fatima@lboro.ac.uk

Michael Wooldridge  
Michael.Wooldridge@cs.ox.ac.uk

<sup>1</sup> University of Loughborough, Loughborough, UK

<sup>2</sup> University of Oxford, Oxford, UK

depends only on its members. A relatively less well studied subclass of coalition games are *partition function games* (PFGs). For a PFG, the value of a coalition depends on its members as well as on the make-up of the non-members. CFGs are a proper subclass of PFGs and CFGs are the most well studied coalition games. Except for a few recent ones, most of the existing methods for solving the CSG problem are for CFGs [32].

Existing solutions for the CSG problem for PFGs were devised either by placing constraints on externalities or else on the function that maps coalition structures to values (see Sect. 6 for details). A common feature of existing work is that it is focussed on games whose properties are known with certainty (we will call such games *deterministic*). However, stochasticity is inherent to many multi-agent settings. Given this, the goal of our present work is to investigate how to solve the CSG problem for *stochastic environments* in which some aspects of the problem are not known with certainty. To this end, we build on our prior work [17] in which we considered the CSG problem for PFGs with priority ordered players and a restricted class of value functions, viz., those that satisfy a certain monotonicity property, and devised a polynomial time solution. In this previous work, the notion of monotonicity was deterministic in the sense that, with probability one, the function that maps coalition structures to values satisfies monotonicity. In this paper, we relax the deterministic monotonicity assumption by allowing a certain degree of non-monotonicity. Specifically, we replace the deterministic monotonicity restriction by *probabilistic monotonicity*. Probabilistic monotonicity means that the value function obeys monotonicity with a certain probability  $0 < p \leq 1$  (for the deterministic case  $p = 1$ ). For probabilistically monotone PFGs with priority ordered players, we devise an algorithm for optimally solving the CSG problem and characterize its time complexity.

The need for optimal coalition structures arises in many real-world applications. For example, in the formation of supply chains [18, 26]. In such a setting, several different manufacturers of components form coalitions to achieve what they cannot do individually. Externalities arise, for example, from the requirement that all components ultimately conform to the same standards. The cost of standardisation procedures incurred by any coalition depends on the number and structure of other coalitions. Another application is in job scheduling [19]. In a job scheduling problem, there is a set of jobs and a set of machines. Each job must be allocated to a machine such that the overall cost of processing jobs is minimized. For further applications of cooperative games, [12] is a comprehensive set of references. In all such applications, the general problem of optimal coalition structure determination involves a combinatorial search. In many cases, the search space is not always unstructured – often there is some form of inherent regularity in at least a part of the space. For example, consider an airline crew scheduling problem, which requires organising staff into coalitions based on individual characteristics, and optimally scheduling the coalitions. The players, i.e., the crew, are ordered in that any non-optimal placement of an individual in the early part of a schedule can propagate inefficiencies down the chain and reduce the value of the entire partition. It is possible that the earlier in the schedule a non-optimality is introduced, the greater the reduction in the value of the partition as a whole, relative to the optimum. In other words, the search space is structured in that there is a relation between how close a partition is to the optimum, and the value of that partition. The closer a structure is to the optimum, the more *likely* it is to have a higher value. A lack of certainty in the values can arise because the number of all possible coalition structures is too huge to enable an accurate measurement of their values. It is therefore important to consider such uncertainties.

To the best of our knowledge, we are the first to consider the CSG problem in an uncertain environment. The key contributions of this paper are:

**Table 1** A summary of notation

$N = \{1, \dots, n\}$	The set of players
$C_{1 \leq i \leq n}$	A coalition of players; $C_i \subseteq N$
$\mathbb{P}_i$	The top $i$ th priority player
$\Delta$	$\Delta \subseteq N, \Delta = \{\mathbb{P}_1, \dots, \mathbb{P}_\delta\}$
$\delta$	$\delta =  \Delta $
$\pi$	A coalition structure over the set $N$
$\beta_\pi^i$	The index of the coalition in $\pi$ to which player $i$ belongs
$\Pi^N$	The set of all coalition structures over $N$
$\Pi_\mathbb{O}$	The set of all optimal coalition structures
$\mathbb{B}_1$	$\frac{(\text{Bell}(n-1) - \text{Bell}(n-2))^2}{3}$
$\mathbb{B}_2(k)$	$\frac{\text{Bell}^2(n-k-2)}{3}$
$\mathbb{B}$	$\frac{\text{Bell}^2(n-\delta-2)}{3}$

1. Developed a new model of probabilistic monotonicity in partition function games, that extends previously studied deterministic monotonicity.
2. Analysed the model and constructively proved that an exact optimum can be found.
3. Devised a greedy algorithm for solving the CSG problem for probabilistic monotone partition function games.
4. Analysed the time complexity of the devised algorithm.

The remainder of the article is organised as follows. Section 2 provides background. Section 3 is a description of the model under investigation and Sect. 4 of the proposed method. An algorithm for solving the CSG problem is in Sect. 5. A review of related literature is in Sects. 6 and 7 concludes. Appendix A collects together all our theorems and proofs. Appendix B is a description of an algorithm for generating the partitions of a set, and an analysis of its time complexity.

## 2 Background

There is a finite non-empty set of *players*  $N = \{1, \dots, n\}$  (see Table 1 for a summary of key notation). The term *coalition* refers to a non-empty subset of  $N$ . The symbol  $C$  possibly with sub/superscripts denotes a coalition and  $\mathcal{C}$  denotes the set of all coalitions of  $N$ .

$$\mathcal{C} = \{C \mid C \subseteq N, C \neq \emptyset\}$$

A *coalition structure* is an exhaustive partition of a set of players into mutually disjoint coalitions. Formally:

**Definition 1** For any coalition  $C$ , let  $\Pi^C$  denote the set of all coalition structures over  $C$ . Then  $\{C_1, C_2, \dots, C_k\} \in \Pi^C$  iff

$$\bigcup_{i=1}^k C_i = C, \quad \forall i \ C_i \neq \emptyset, \quad \text{and} \quad \forall i \neq j \ C_i \cap C_j = \emptyset.$$

The symbol  $\pi$  possibly with sub/superscripts will denote a coalition structure. An *embedded coalition* is a coalition together with a specification of how the non-members are organised into coalitions. It is formally defined as follows:

**Definition 2** Let  $\mathcal{E}$  denote the set of all embedded coalitions. Then

$$\mathcal{E} = \{(C, \pi) \mid C \in \pi \in \Pi^N\}$$

**Definition 3** A characteristic function game (CFG) is a pair  $(N, v_1)$  where  $v_1 : 2^N \rightarrow \mathbb{R}$  and  $2^N$  denotes the set of all subsets of  $N$ . A partition function game is a pair  $(N, v_2)$  where  $v_2 : \mathcal{E} \rightarrow \mathbb{R}$ .

Thus CFGs are a subclass of PFGs.

**Definition 4** The value of a coalition structure over  $N$  is given by an objective function  $v : \Pi^N \rightarrow \mathbb{R}$ .

In the literature on CSG, the function that maps coalition structures to values, i.e., the *objective function*  $v$ , is a social welfare function. It is commonly assumed to be the sum of coalition values. In the proposed model (described in Sect. 3), however, the value of a structure does not have to be the sum of the values of its coalitions but could be any function.

The CSG problem then is to find an *optimal structure*, i.e., a structure  $\mathbb{O}$  such that  $v(\mathbb{O})$  is the highest between all coalition structures. For  $n$  player games, the number of all possible structures is  $Bell(n)$  [5, 6] where

$$Bell(n) = \sum_{i=0}^{n-1} \binom{n-1}{i} \times Bell(i)$$

with  $Bell(0) = Bell(1) = 1$ . Since  $Bell(n) \sim \Theta(n^n)$  [6], a brute force method for PFGs (and CFGs) is not a feasible.

Given the computational complexity of CSG, it is natural to investigate easy instances of the problem. In prior work [17], we showed how the CSG problem for PFGs can be solved optimally in polynomial time, provided that the set  $N$  is ordered and the objective function is deterministically *monotonic*. Our goal now is to generalize this method to stochastic environments where monotonicity is satisfied with a certain probability. Monotonicity is defined in terms of a player ordering.

### 2.1 Player ordering

In the definition of both CFGs and PFGs, the term *coalition* refers to a set of players, i.e., there is no notion of ordering. Nevertheless, the Shapley value [36], a well-known axiomatic solution for CFGs, and its adaptation [27] to PFGs are motivated by a bargaining procedure in which the players are assigned a random ordering and coalition formation is viewed as a sequential process that happens as per the ordering. In generalized CFGs [28], attention is paid to ordering within the definition of a game. A generalized CFG is defined in terms of a set of players and a characteristic function that maps orderings on coalitions to numbers.

The notion of ordering is prevalent not only in the definition of cooperative games and solutions to them, but also in their applications to computationally hard optimization problems such as matching, network optimisation, and scheduling [12, 22]. In the context of these applications, the notion of ordering is used to determine computationally easy problem instances. For example, consider scheduling. In a scheduling problem [39], there is a set of jobs and a set of machines. Each job must be allocated to a machine and, for each machine, an ordering over its allocated jobs must be determined such that the overall cost of processing jobs as per the order is minimized. Computationally easy instances of this problem are sought by imposing restrictions such as the cost function being monotonic, and the set of jobs being an ordered set. In particular, much of the scheduling literature [19, 20] has focussed on a restricted class of objective functions called *priority-generating functions*. An objective function is said to be priority-generating if a related function called *priority function* exists which imposes an ordering over the set of jobs by assigning to jobs certain values called *priorities*. Crucially, priorities are assigned to jobs such that, based on priorities, the optimal schedule can be found in polynomial time. Thus, if a priority function can be found for a given objective function, and it takes polynomial time to compute job priorities using the function, then the scheduling problem can be solved in polynomial time [39].

Motivated by the above observations, in prior work [17] we took a priority-based approach and showed how the CSG problem for PFGs can be solved optimally in polynomial time provided the set  $N$  is ordered and the function  $v$  is monotonic (see Sect. 2.2). The CSG problem is related to scheduling in that players are analogous to jobs and coalitions to machines. A crucial difference though between existing priority-based scheduling methods [19, 20, 39] and [17] is that the former require job priorities as an input, while latter requires only the existence of player priorities to be known without requiring the actual priorities as problem input.

## 2.2 Monotonicity

In [17], the players are assumed to be priority ordered, and *monotonicity* of the objective function  $v$  is defined in terms of a distance metric  $d$ . For any two structures  $\pi^1$  and  $\pi^2$ ,  $d(\pi^1, \pi^2)$  denotes the distance between  $\pi^1$  and  $\pi^2$ .

**Definition 5** (*Deterministic monotonicity*) For any two structures  $\pi^1$  and  $\pi^2$ , and a unique optimum  $\mathbb{O}$ :

$$d(\mathbb{O}, \pi^1) < d(\mathbb{O}, \pi^2) \Rightarrow v(\pi^1) > v(\pi^2) \quad (1)$$

with probability one. □

Then a PFG is deterministically monotonic if, for some player ordering,  $v$  is monotonic. It was shown that deterministic monotonicity is a property that can be satisfied by PFGs with positive only, negative only, and mixed externalities. Further, deterministic monotonicity was shown to be satisfiable for a wide range of objective functions, i.e., the value of a partition is not restricted to be the sum of the values of its coalitions but can be any function of these values. Illustrations of deterministically monotonic PFGs, and a polynomial time method for solving the CSG problem for deterministically monotonic PFGs are in [17]. Our aim now is to generalise this method to probabilistic monotonicity.

### 3 Coalition structure generation in stochastic environments

The players to be partitioned are given by the set  $N$ . The priority ordering referred to in Sect. 2.1 exists over certain players in  $N$ . The set of these ordered players is denoted  $\Delta \subseteq N$  with  $\delta = |\Delta|$ . Each player in  $N - \Delta$  is called a non-priority player. Let  $\mathbb{P}_i \in \Delta$  denote the top  $i$ th priority player, i.e., the ordering is  $\mathbb{P}_1 > \dots > \mathbb{P}_\delta$ . The quality of a coalition structure depends on how the priority players are positioned in the structure. The higher up a player is in the ordering, the more important it is to have the player in its optimal position.

Any coalition that contains at least one priority player is called a *priority coalition*. An ordering over  $\Delta$  induces an ordering over the priority coalitions: they are ordered as per the priorities of their highest priority members. Suppose  $hpm(X)$  denotes the highest priority member of coalition  $X$ . Suppose that the priority players are spread over  $m$  coalitions. Then these  $m$  coalitions form a sequence  $(C_1, C_2, \dots, C_m)$  such that,  $\mathbb{P}_1 \in C_1$ , and if  $\mathbb{P}_x > \mathbb{P}_y$  and  $\mathbb{P}_x = hpm(C_i)$  and  $\mathbb{P}_y = hpm(C_j)$ , then  $i < j$ . For any coalition structure over  $N$ , there is no ordering over coalitions comprised solely of non-priority players.

We consider stochastic environments modelled as *probabilistically monotone* PFGs for which it is known that a priority ordering exists, but the ordering itself is unknown, i.e., the identities of the priority players  $\mathbb{P}_1, \dots, \mathbb{P}_\delta$  are unknown. Solving the CSG problem requires determining these identities and an optimal way of partitioning the players in  $N$ . To achieve this, we represent coalition structures as described in Sect. 3.1. In Sect. 3.2, we introduce a distance metric, in terms of which we define probabilistic monotonicity in Sect. 3.3. The proposed method for solving the CSG problem is described in Sect. 4.

#### 3.1 Representation

Let  $\mathbb{P}$  denote the sequence  $(\mathbb{P}_1, \dots, \mathbb{P}_\delta)$ . For any  $k \leq \delta$ ,  $\mathbb{P}_{|k}$  will denote the  $k$  element prefix of  $\mathbb{P}$ , i.e.,  $\mathbb{P}_{|k} = (\mathbb{P}_1, \dots, \mathbb{P}_k)$ .  $\mathbb{P}_{|k}^E$  is defined as

$$\mathbb{P}_{|k}^E = \{X \mid X \in Perm(\Delta) \text{ and } X_{|k} = \mathbb{P}_{|k}\}$$

where  $Perm(\Delta)$  denotes the set of all permutations of  $\Delta$ . A coalition structure over  $\Delta$  is represented as a sequence  $\pi = (x_1, x_2, \dots, x_\delta)$  such that  $x_i \in \{1, \dots, \delta\}$  is the index of the coalition to which player  $\mathbb{P}_i$  belongs. Then the set of all structures over  $\Delta$  is:

$$\Pi^{\Delta} = \{(x_1, \dots, x_\delta) \mid x_1 = 1, x_{i>1} \in \{1, \dots, \max(x_1, \dots, x_{i-1}) + 1\}\} \tag{2}$$

$\pi_{|k}$  will denote the  $k$  element prefix of  $\pi = (x_1, x_2, \dots, x_\delta)$ .  $\pi_{|k}^E$  will denote the set of all those sequences in  $\Pi^{\Delta}$  whose  $k$  element prefix is  $\pi_{|k}$ .

$$\pi_{|k}^E = \{X \mid X \in \Pi^{\Delta} \text{ and } X_{|k} = \pi_{|k}\}$$

**Example 1** Consider a PFG with  $n = 3$  players who are members of an airline crew. There are  $Bell(3) = 5$  possible coalition structures. Let  $\delta = 3$ . Column 1 of Table 2 shows how coalition structures will be represented. For three players, there are six possible orderings of players as illustrated in Columns 2 to 7. How the representation of a particular structure is interpreted depends on the player ordering  $\mathbb{P}$ . For example, for  $\mathbb{P} = (1, 2, 3)$  (see Column 2), we have  $\mathbb{P}_1 = 1$ ,  $\mathbb{P}_2 = 2$ , and  $\mathbb{P}_3 = 3$ . The representation  $(1, 1, 1)$  (see Row 1, Column 2) means the structure comprised of the single grand coalition  $C_1 = \{1, 2, 3\}$ . The

**Table 2** An illustration of the coalition structure representation and semantics for each one of the six possible player orderings for  $n = 3$  player games

	(1)	(2)	(3)	(4)	(5)	(6)	(7)
Representation		$\mathbb{P} = (1, 2, 3)$	$\mathbb{P} = (1, 3, 2)$	$\mathbb{P} = (2, 1, 3)$	$\mathbb{P} = (2, 3, 1)$	$\mathbb{P} = (3, 1, 2)$	$\mathbb{P} = (3, 2, 1)$
(1)	(1, 1, 1)	({1, 2, 3})	({1, 2, 3})	({1, 2, 3})	({1, 2, 3})	({1, 2, 3})	({1, 2, 3})
(2)	(1, 1, 2)	({1, 2}, {3})	({1, 3}, {2})	({1, 2}, {3})	({2, 3}, {1})	({1, 3}, {2})	({2, 3}, {1})
(3)	(1, 2, 1)	({1, 3}, {2})	({1, 2}, {3})	({2, 3}, {1})	({1, 2}, {3})	({2, 3}, {1})	({1, 3}, {2})
(4)	(1, 2, 2)	({1}, {2, 3})	({1}, {2, 3})	({2}, {1, 3})	({2}, {1, 3})	({3}, {1, 2})	({3}, {1, 2})
(5)	(1, 2, 3)	({1}, {2}, {3})	({1}, {2}, {3})	({1}, {2}, {3})	({1}, {2}, {3})	({1}, {2}, {3})	({1}, {2}, {3})

representation (1, 1, 2) (see Row 2, Column 2) is the structure  $(\{1, 2\}, \{3\})$  in which the two top priority players are together in the coalition  $C_1$  and the player  $\mathbb{P}_3$  is in the singleton coalition  $C_2$ . However, the same representation (1, 1, 2) for the ordering  $\mathbb{P} = (1, 3, 2)$  (see Row 2, Column 3) is the structure  $(\{1, 3\}, \{2\})$ . The remaining entries in the table may be interpreted similarly. □

Let  $\Pi_O$  denote the set of all optimal structures over  $N$ . For any  $\pi \in \Pi^N$ ,  $\beta_\pi^i$  will denote the index of the coalition to which player  $i$  belongs in the structure  $\pi$ . We assume that the optima are unique up to the positions of the top  $3 < \delta \leq n - 3$  priority players, i.e., for any two distinct optimal structures  $\pi^1 \in \Pi_O$  and  $\pi^2 \in \Pi_O$ ,  $\beta_{\pi^1}^i = \beta_{\pi^2}^i$  for each  $1 \leq i \leq \delta$ .

### 3.2 Distance measure

To measure the distance between any two structures  $\pi^1$  and  $\pi^2$  over  $N$ , we define a metric  $d$  in terms of the positions of the priority players, i.e., in terms of the restriction of  $\pi^1$  and  $\pi^2$  to  $\Delta$ . A restriction<sup>1</sup> can be represented as described in Sect. 3.1.

$$d(\pi_{|\delta}^1, \pi_{|\delta}^2) = \delta - \max_{1 \leq i \leq \delta} \{ \pi_{|i}^1 = \pi_{|i}^2 \} \tag{3}$$

**Theorem 1** *The distance function  $d$  obeys all metric axioms.*

**Proof** The axioms are identity, symmetry, and triangle inequality.

**Identity**

For any coalition structure  $\pi^1$  over  $N$ ,  $d(\pi_{|\delta}^1, \pi_{|\delta}^1) = 0$ . For any coalition structure  $\pi^2$  over  $N$ , if  $\pi_{|\delta}^1 \neq \pi_{|\delta}^2$ , then  $d(\pi_{|\delta}^1, \pi_{|\delta}^2) \neq 0$ .

<sup>1</sup> The restriction of a structure  $\pi = (C_1, C_2, \dots)$  over  $N$  to some  $s \subseteq N$  is  $\pi_{|s} = (C_1 \cap s, C_2 \cap s, \dots)$ . Note that, in Sect. 3.1, ' $|$ ' is used in a slightly different way; in  $\pi_{|k}$ ,  $k \in \{1, \dots, \delta\}$ . The meaning of ' $|$ ' should be clear from context.

- Symmetry** For any two coalition structures  $\pi^1$  and  $\pi^2$  over  $N$ ,  $d(\pi^1_{|\delta}, \pi^2_{|\delta}) = d(\pi^2_{|\delta}, \pi^1_{|\delta})$ .
- Triangle inequality** For any three coalition structures  $\pi^1$ ,  $\pi^2$ , and  $\pi^3$  over  $N$ ,  $d(\pi^1_{|\delta}, \pi^2_{|\delta}) \leq d(\pi^1_{|\delta}, \pi^3_{|\delta}) + d(\pi^2_{|\delta}, \pi^3_{|\delta})$ .

Let  $\max_{1 \leq i \leq \delta} \{\pi^1_{|i} = \pi^2_{|i}\} = k$ ,  $\max_{1 \leq i \leq \delta} \{\pi^1_{|i} = \pi^3_{|i}\} = x$ , and  $\max_{1 \leq i \leq \delta} \{\pi^2_{|i} = \pi^3_{|i}\} = y$  where  $1 \leq k \leq \delta$ ,  $1 \leq x \leq \delta$ , and  $1 \leq y \leq \delta$ . Consider each one of the following possibilities:

- $x > k$  : For this case,  $y = k$ . Thus,  $d(\pi^1_{|\delta}, \pi^3_{|\delta}) + d(\pi^2_{|\delta}, \pi^3_{|\delta}) = \delta - x + \delta - k \geq \delta - k$ , and  $d(\pi^1_{|\delta}, \pi^2_{|\delta}) = \delta - k$ .
  - $x < k$  : For this case,  $y = x$ . Thus,  $d(\pi^1_{|\delta}, \pi^3_{|\delta}) + d(\pi^2_{|\delta}, \pi^3_{|\delta}) = \delta - x + \delta - x \geq \delta - k$ , and  $d(\pi^1_{|\delta}, \pi^2_{|\delta}) = \delta - k$ .
  - $x = k$  : For this case,  $y \geq k$ . Thus,  $d(\pi^1_{|\delta}, \pi^3_{|\delta}) + d(\pi^2_{|\delta}, \pi^3_{|\delta}) = \delta - k + \delta - y \geq \delta - k$ , and  $d(\pi^1_{|\delta}, \pi^2_{|\delta}) = \delta - k$ . □
- 

### 3.3 Probabilistic monotonicity

Let  $\mathbb{S}$  be the set of all ordered pairs of coalition structures over  $N$ . Formally,

$$\mathbb{S} = \{(\pi^1, \pi^2) \mid \pi^1 \in \Pi^N \text{ and } \pi^2 \in \Pi^N \text{ and } \pi^1 \neq \pi^2\}$$

For a game of  $n$  players,  $|\mathbb{S}| = \text{Bell}(n) \times (\text{Bell}(n) - 1)$ . Then, for a player ordering, probabilistic monotonicity of an objective function  $v$  is defined as follows.

**Definition 6** (*Probabilistic monotonicity*) For a random pair  $(\pi^1, \pi^2) \in \mathbb{S}$  such that  $\pi^1 \notin \Pi_{\mathbb{O}}$  and  $\pi^2 \notin \Pi_{\mathbb{O}}$ , and any optimal structure  $\mathbb{O} \in \Pi_{\mathbb{O}}$

$$d(\mathbb{O}_{|\delta}, \pi^1_{|\delta}) < d(\mathbb{O}_{|\delta}, \pi^2_{|\delta}) \Rightarrow v(\pi^1) > v(\pi^2) \tag{4}$$

with a certain probability. Specifically, for a random pair  $(\pi^1, \pi^2) \in \mathbb{S}$  such that neither  $\pi^1$  nor  $\pi^2$  is an optimum, the probability that  $v(\pi^1) > v(\pi^2)$  conditional on  $d(\mathbb{O}_{|\delta}, \pi^1_{|\delta}) < d(\mathbb{O}_{|\delta}, \pi^2_{|\delta})$  is

$$p\left(v(\pi^1) > v(\pi^2) \mid d(\mathbb{O}_{|\delta}, \pi^1_{|\delta}) < d(\mathbb{O}_{|\delta}, \pi^2_{|\delta})\right) \leq 1$$

For a random pair  $(\pi^1, \pi^2) \in \mathbb{S}$  such that any one element of the pair, say  $\pi^1$ , is an optimum,  $v(\pi^1) > v(\pi^2)$  with probability 1. □

**Definition 7** A PFG<sup>2</sup>  $(N, v_2, v)$  is probabilistically monotonic if  $v$  is probabilistically monotone for some player ordering. □

<sup>2</sup> In the literature on coalition game theory, a PFG is defined as a pair  $(N, v_2)$ . We include the objective function  $v$  in the definition of a PFG.



Probabilistic monotonicity is modelled as follows. Let the set  $\mathbb{A}$  be defined as follows:

$$\mathbb{A} = \{(u, v) \mid u \in \{<, =, >\} \text{ and } v \in \{<, =, >\}\}.$$

Let the functions  $f : \mathbb{S} \rightarrow \mathbb{A}$ ,  $r^d : \mathbb{S} \rightarrow \{<, =, >\}$ , and  $r^v : \mathbb{S} \rightarrow \{<, =, >\}$  be defined as follows. For any  $(x, y) \in \mathbb{S}$ ,  $f(x, y) = (r^d(x, y), r^v(x, y))$  where

$$r^d(x, y) = \begin{cases} < & \text{if } d(\mathbb{O}_{|\delta}, x_{|\delta}) < d(\mathbb{O}_{|\delta}, y_{|\delta}) \\ = & \text{if } d(\mathbb{O}_{|\delta}, x_{|\delta}) = d(\mathbb{O}_{|\delta}, y_{|\delta}) \\ > & \text{if } d(\mathbb{O}_{|\delta}, x_{|\delta}) > d(\mathbb{O}_{|\delta}, y_{|\delta}) \end{cases}$$

and

$$r^v(x, y) = \begin{cases} < & \text{if } v(x) < v(y) \\ = & \text{if } v(x) = v(y) \\ > & \text{if } v(x) > v(y) \end{cases}$$

The set  $\mathbb{S}$  can be partitioned into nine pairwise disjoint subsets as follows:

1.  $\mathbb{S}_{ee} = \{(x, y) \mid (x, y) \in \mathbb{S} \text{ and } f(x, y) \text{ is } (=, =)\}$
2.  $\mathbb{S}_{eg} = \{(x, y) \mid (x, y) \in \mathbb{S} \text{ and } f(x, y) \text{ is } (=, >)\}$ .
3.  $\mathbb{S}_{el} = \{(x, y) \mid (x, y) \in \mathbb{S} \text{ and } f(x, y) \text{ is } (=, <)\}$
4.  $\mathbb{S}_{ge} = \{(x, y) \mid (x, y) \in \mathbb{S} \text{ and } f(x, y) \text{ is } (>, =)\}$
5.  $\mathbb{S}_{gg} = \{(x, y) \mid (x, y) \in \mathbb{S} \text{ and } f(x, y) \text{ is } (>, >)\}$
6.  $\mathbb{S}_{gl} = \{(x, y) \mid (x, y) \in \mathbb{S} \text{ and } f(x, y) \text{ is } (>, <)\}$
7.  $\mathbb{S}_{le} = \{(x, y) \mid (x, y) \in \mathbb{S} \text{ and } f(x, y) \text{ is } (<, =)\}$
8.  $\mathbb{S}_{lg} = \{(x, y) \mid (x, y) \in \mathbb{S} \text{ and } f(x, y) \text{ is } (<, >)\}$
9.  $\mathbb{S}_{ll} = \{(x, y) \mid (x, y) \in \mathbb{S} \text{ and } f(x, y) \text{ is } (<, <)\}$

Then  $\mathbb{S}$  is the union of these nine subsets:

$$\mathbb{S} = \mathbb{S}_{ee} \cup \mathbb{S}_{eg} \cup \mathbb{S}_{el} \cup \mathbb{S}_{ge} \cup \mathbb{S}_{gg} \cup \mathbb{S}_{gl} \cup \mathbb{S}_{le} \cup \mathbb{S}_{lg} \cup \mathbb{S}_{ll}. \tag{5}$$

Between all these nine subsets, only  $\mathbb{S}_{ee}$ ,  $\mathbb{S}_{eg}$ ,  $\mathbb{S}_{el}$ ,  $\mathbb{S}_{gl}$ , and  $\mathbb{S}_{lg}$  satisfy monotonicity. The union of these is denoted  $\mathbb{S}_{MON}$ :

$$\mathbb{S}_{MON} = \mathbb{S}_{ee} \cup \mathbb{S}_{eg} \cup \mathbb{S}_{el} \cup \mathbb{S}_{gl} \cup \mathbb{S}_{lg}. \tag{6}$$

**Definition 8** Each pair in  $\mathbb{S}_{MON}$  is called a *monotonicity-satisfying pair*.

Probabilistic monotonicity is modelled by a probability distribution (see Table 3) induced by the function  $v$  over the set  $\mathbb{A}$ . As per Table 3,

**Table 3** Probability distribution induced by  $\nu$  on  $\mathbb{A}$

Element of $\mathbb{A}$	Probability
$(=, =)$	$p_{ee} = \frac{ \mathbb{S}_{ee} }{ \mathbb{S} }$
$(=, >)$	$p_{eg} = \frac{ \mathbb{S}_{eg} }{ \mathbb{S} }$
$(=, <)$	$p_{el} = \frac{ \mathbb{S}_{el} }{ \mathbb{S} }$
$(>, =)$	$p_{ge} = \frac{ \mathbb{S}_{ge} }{ \mathbb{S} }$
$(>, >)$	$p_{gg} = \frac{ \mathbb{S}_{gg} }{ \mathbb{S} }$
$(>, <)$	$p_{gl} = \frac{ \mathbb{S}_{gl} }{ \mathbb{S} }$
$(<, =)$	$p_{le} = \frac{ \mathbb{S}_{le} }{ \mathbb{S} }$
$(<, >)$	$p_{lg} = \frac{ \mathbb{S}_{lg} }{ \mathbb{S} }$
$(<, <)$	$p_{ll} = \frac{ \mathbb{S}_{ll} }{ \mathbb{S} }$

$$p\left(\nu(\pi^1) > \nu(\pi^2) \mid d(\mathbb{O}_{|\delta}, \pi^1_\delta) < d(\mathbb{O}_{|\delta}, \pi^2_\delta)\right) = \frac{p_{lg}}{p_{le} + p_{lg} + p_{ll}}.$$

Since the elements of  $\mathbb{S}$  are ordered pairs, we have the following :

1.  $p_{eg} = p_{el}$  because  $(\pi^1, \pi^2) \in \mathbb{S}_{eg}$  iff  $(\pi^2, \pi^1) \in \mathbb{S}_{el}$ .
2.  $p_{ge} = p_{le}$  because  $(\pi^1, \pi^2) \in \mathbb{S}_{ge}$  iff  $(\pi^2, \pi^1) \in \mathbb{S}_{le}$ .
3.  $p_{gg} = p_{ll}$  because  $(\pi^1, \pi^2) \in \mathbb{S}_{gg}$  iff  $(\pi^2, \pi^1) \in \mathbb{S}_{ll}$ .
4.  $p_{gl} = p_{lg}$  because if  $(\pi^1, \pi^2) \in \mathbb{S}_{gl}$  iff  $(\pi^2, \pi^1) \in \mathbb{S}_{lg}$ .

For probabilistic monotonicity,  $p_{ge} \geq 0$ ,  $p_{gg} \geq 0$ ,  $p_{le} \geq 0$ , and  $p_{ll} \geq 0$ . In contrast, for deterministic monotonicity, each one of the probabilities  $p_{ge}$ ,  $p_{gg}$ ,  $p_{le}$ , and  $p_{ll}$  is zero. Example 2 is an illustration of probabilistic monotonicity.

**Example 2** Consider a game with  $n = 3$  players who are members of an airline crew. Let  $N = \{1, 2, 3\}$ ,  $\mathbb{P} = (1, 2, 3)$  and  $\delta = 3$ . There are 5 structures possible and 20 possible ordered pairs of structures as listed in Table 4 (the corresponding probability distribution is shown in Table 5). Suppose the optimum is  $\mathbb{O} = (\{1, 3\}, \{2\})$ , which is represented as the sequence (1, 2, 1) enclosed in oval. Consider Row 4.  $\pi^1$  is further away from the optimum than is  $\pi^2$  and yet  $\nu(\pi^1) > \nu(\pi^2)$ . This is a violation of monotonicity. Likewise, monotonicity is violated in the rows 11, 15, and 17. There is no violation in any of the remaining rows.

Clearly, the number of rows in which monotonicity is violated cannot be arbitrary. If we want to manage computational complexity, the *degree of non-monotonicity* must be bounded.

**Definition 9** (*Degree of non-monotonicity*) The degree of non-monotonicity  $D$  is the sum of the cardinalities of  $\mathbb{S}_{ge}$ ,  $\mathbb{S}_{gg}$ ,  $\mathbb{S}_{le}$ , and  $\mathbb{S}_{ll}$ .

**Table 4** An illustration of probabilistic monotonicity for a 3 player game with  $\mathbb{P} = (1, 2, 3)$ . The optimal structure  $\mathbb{O} = (\{1, 3\}, \{2\})$  is represented as  $(1, 2, 1)$ . Each row corresponds to an element of the set  $\mathbb{S}$ . Four (viz. those in rows 4, 11, 15, and 17 highlighted in grey) out of the twenty elements of  $\mathbb{S}$  violate monotonicity

	$\pi^1$	$\pi^2$	$f(\pi^1, \pi^2)$
1	(1, 1, 2)	(1, 2, 1)	(>, <)
2	(1, 1, 2)	(1, 2, 2)	(>, <)
3	(1, 1, 2)	(1, 1, 1)	(=, >)
4	(1, 1, 2)	(1, 2, 3)	(>, >)
5	(1, 2, 1)	(1, 1, 2)	(<, >)
6	(1, 2, 1)	(1, 2, 2)	(<, >)
7	(1, 2, 1)	(1, 1, 1)	(<, >)
8	(1, 2, 1)	(1, 2, 3)	(<, >)
9	(1, 2, 2)	(1, 1, 2)	(<, >)
10	(1, 2, 2)	(1, 2, 1)	(>, <)
11	(1, 2, 2)	(1, 1, 1)	(<, <)
12	(1, 2, 2)	(1, 2, 3)	(=, =)
13	(1, 1, 1)	(1, 1, 2)	(=, <)
14	(1, 1, 1)	(1, 2, 1)	(>, <)
15	(1, 1, 1)	(1, 2, 2)	(>, >)
16	(1, 1, 1)	(1, 2, 3)	(>, <)
17	(1, 2, 3)	(1, 1, 2)	(<, <)
18	(1, 2, 3)	(1, 2, 1)	(>, <)
19	(1, 2, 3)	(1, 2, 2)	(=, =)
20	(1, 2, 3)	(1, 1, 1)	(<, >)

$$D = |\mathbb{S}_{ge}| + |\mathbb{S}_{gg}| + |\mathbb{S}_{le}| + |\mathbb{S}_{ll}| \quad \square$$

Let  $\mathbb{B}$  denote  $\frac{Bel^2(n-\delta-2)}{3}$  and suppose that  $D$  satisfies the relation

$$D < \mathbb{B}$$

for some  $3 < \delta \leq n - 3$  and  $n > 7$ . Our aim then is to solve the following CSG problem:

**Table 5** Probability distribution table for the game given in Table 4

Element of $\mathbb{A}$	Probability
$(=, =)$	$p_{ee} = 0.1$
$(=, >)$	$p_{eg} = 0.05$
$(=, <)$	$p_{el} = 0.05$
$(>, =)$	$p_{ge} = 0$
$(>, >)$	$p_{gg} = 0.1$
$(>, <)$	$p_{gl} = 0.3$
$(<, =)$	$p_{le} = 0$
$(<, >)$	$p_{lg} = 0.3$
$(<, <)$	$p_{ll} = 0.1$

**CSG problem definition:** For a probabilistically monotone PFG  $(N, v_2, v)$  with the degree of non-monotonicity known to be bounded above by  $\mathbb{B}$ , find the identities of  $\mathbb{P}_1, \dots, \mathbb{P}_\delta$  and an optimal structure  $\mathbb{O}$  where

$$\mathbb{O} \in \left\{ \pi \mid \underset{\pi \in \Pi^N}{arg\ max} \ v(\pi) \right\}$$

given as input  $N$  and the function  $r^v$  induced by  $v$ .

Note that the actual values of coalition structures as given by  $v$  are not part of the input. Rather, only the relation between the values of any two structures is part of the problem input.

### 4 The proposed method

In Sect. 4.1 is a brief overview of the three key steps of the proposed method for finding  $\mathbb{P}$  and an optimal structure  $\mathbb{O}$ . Details of Step 1 are in Sect. 4.2, Step 2 in Sect. 4.3, and Step 3 in Sect. 4.4. A complete formulation of the method is given as an algorithm in Sect. 5.

#### 4.1 CSG method: overview

- Step 1 Determine who the two top priority players are, and their optimal coalitions. At the end of this step,  $\mathbb{P}_{12}$  and  $\mathbb{O}_{12}$  will become known.
- Step 2 For each  $3 \leq i \leq \delta$ , determine the identity of the player  $\mathbb{P}_i$  and its optimal coalition. At the end of this step,  $\mathbb{P}_{1\delta}$  and  $\mathbb{O}_{1\delta}$  will become known.
- Step 3 Determine  $\mathbb{P}$  and  $\mathbb{O}$ .

Consider Step 1. To begin, we know that  $\mathbb{P}_1 \in N$ , so there are  $n$  possibilities for the identity of  $\mathbb{P}_1$ . We also know that  $\mathbb{P}_1$  must belong to the first coalition (denoted  $C_1^{\mathbb{O}}$ ) in  $\mathbb{O}$ . Then, for any one possibility for the identity of  $\mathbb{P}_1$ , say  $\mathbb{P}_1 = x$ , we know that there must be  $n - 1$  possibilities for the identity of  $\mathbb{P}_2$ , i.e.,  $\mathbb{P}_2 \in N - \{x\}$ , and that there must be two possibilities for its optimal coalition, i.e.,  $\mathbb{P}_2$  must be a member of either  $C_1^{\mathbb{O}}$  or  $C_2^{\mathbb{O}}$ . Let  $Z$  be the set of all these possibilities and let each element of  $Z$  be a quadruple defined as follows:

$$Z = \{(x, y, 1, z) \mid x \in N, y \in N - \{x\}, z \in \{1, 2\}\} \tag{7}$$

The semantics of quadruple  $(x, y, 1, z)$  is that, it is possible that  $\mathbb{P}_1 = x, \mathbb{P}_2 = y, x$  belongs to the coalition  $C_1^{\mathbb{O}}$ , and  $y$  belongs to the coalition  $C_z^{\mathbb{O}}$  in  $\mathbb{O}$ . For example, the quadruple  $(2, 3, 1, 2)$  means that  $\mathbb{P}_1 = 2, \mathbb{P}_2 = 3, 2 \in C_1^{\mathbb{O}}$ , and  $3 \in C_2^{\mathbb{O}}$  is a possibility. We find it convenient to introduce terminology for referring to certain pairs of elements of  $Z$ .

**Definition 10** For any  $x \in N$  and any  $y \in N - \{x\}$ ,  $(x, y, 1, 1)$  and  $(y, x, 1, 1)$  are each others' partners, and  $(x, y, 1, 2)$  and  $(y, x, 1, 2)$  are each others' partners. A partner pair is in one of the following forms:

- $((x, y, 1, 1), (y, x, 1, 1))$
- $((x, y, 1, 2), (y, x, 1, 2))$

Lemmas 1 and 2 readily follow from the definition of  $Z$  and that of a partner pair.

**Lemma 1** *All the following assertions are true.*

- $|Z| = 2 \times n \times (n - 1)$ .
- *Every element in  $Z$  has a unique partner in  $Z$ .*
- *Every element in  $Z$  is the partner of a unique element in  $Z$ .* □

**Lemma 2** *Any two partner pairs of  $Z$  must be in one of the following forms:*

1.  $((i, t, 1, 1), (t, i, 1, 1)), ((j, s, 1, 1), (s, j, 1, 1))$
2.  $((i, t, 1, 2), (t, i, 1, 2)), ((j, s, 1, 2), (s, j, 1, 2))$
3.  $((i, t, 1, 2), (t, i, 1, 2)), ((j, s, 1, 1), (s, j, 1, 1))$
4.  $((i, t, 1, 1), (t, i, 1, 1)), ((i, t, 1, 2), (t, i, 1, 2))$

where  $i \in N, j \in N - \{i\}, s \in N - \{i, j\}$ , and  $t \in N - \{i, j\}$ . □

We use  $(i, j, x, y) \vDash \mathbb{O}$  as short for  $(\mathbb{P}_1 = i \wedge \mathbb{P}_2 = j \wedge \mathbb{P}_1 \in C_x^{\mathbb{O}} \wedge \mathbb{P}_2 \in C_y^{\mathbb{O}})$  and  $(i, j, x, y) \not\vDash \mathbb{O}$  as short for  $\neg(\mathbb{P}_1 = i \wedge \mathbb{P}_2 = j \wedge \mathbb{P}_1 \in C_x^{\mathbb{O}} \wedge \mathbb{P}_2 \in C_y^{\mathbb{O}})$ . Now, we know that there is a unique  $z \in Z$  such that  $z \vDash \mathbb{O}$  and for each  $\bar{z} \in Z, \bar{z} \not\vDash \mathbb{O}$  iff  $z \neq \bar{z}$ . This observation leads to the definition of set  $\bar{Z}$ :

$$\bar{Z} = \{\bar{z} \in Z \mid \bar{z} \not\vDash \mathbb{O}\}$$

Since only one element of  $Z$  corresponds to  $\mathbb{O}$ , we get  $|\bar{Z}| = |Z| - 1$ .

Step 1 ( details in Sect. 4.2) involves doing three different tests  $T1, T2$ , and  $T3$  with appropriate arguments in such a way that at least  $|Z| - 2$  elements of  $\bar{Z}$  can be determined. Once this is done, one of the elements in  $Z - \bar{Z}$  must correspond to  $\mathbb{O}$  and this concludes Step 1. For Step 2, we define a further test  $T4$  details of which are in Sect. 4.3.

## 4.2 CSG method: Step 1

Sect. 4.2.1 gives a definition of the tests T1, T2, and T3. Section 4.2.2 is a specification of the parameters of these tests. Section 4.2.3 describes how these tests can be used to find an element of  $\mathbb{S}_{MON}$ . Section 4.2.4 describes how these tests can be used to determine who the two top priority players are and their positions in an optimal structure.

### 4.2.1 The tests T1, T2, T3

The tests T1, T2, and T3 are defined as follows:

**T1** For any  $i \in N$  and  $j \in N - \{i\}$ , the test  $T1(i, j)$  compares the values of any two structures  $\pi_1 \in \Pi^N$  and  $\pi_2 \in \Pi^N$  such that

- \* in  $\pi_1$ , the players  $i$  and  $t$  belong to different coalitions but  $j$  and  $s$  belong to the same coalition, and
- \* in  $\pi_2$ , the players  $i$  and  $t$  belong to the same coalition but  $j$  and  $s$  belong to different coalitions

where  $t$  is an arbitrary element of  $N - \{i, j\}$  and  $s$  is an arbitrary element of  $N - \{i, j\}$ . The values of  $i, j, s$ , and  $t$  are determined on the basis of the elements in  $Z$  (see Theorem 5 for details). Let  $S_{T1}^1(i, j)$  denote the set of all those structures in which the players  $i$  and  $t$  belong to different coalitions but  $j$  and  $s$  belong to the same coalition. Let  $S_{T1}^2(i, j)$  denote the set of all those structures in which the players  $i$  and  $t$  belong to the same coalition but  $j$  and  $s$  belong to different coalitions. The test  $T1(i, j)$  uses  $r^v$  to compare the value of any structure from  $S_{T1}^1(i, j)$  to the value of any structure from  $S_{T1}^2(i, j)$ .

**T2** For any  $i \in N$  and any  $j \in N - \{i\}$ , the test  $T2(i, j)$  compares the values of any two structures  $\pi_1 \in \Pi^N$  and  $\pi_2 \in \Pi^N$  such that

- \* in  $\pi_1$ ,  $i$  and  $t$  are apart, and  $j$  and  $s$  are apart, and
- \* in  $\pi_2$ ,  $i$  and  $t$  are together, and  $j$  and  $s$  are together

where  $t$  is an arbitrary element of  $N - \{i, j\}$  and  $s$  is an arbitrary element of  $N - \{i, j\}$ . The values of  $i, j, s$ , and  $t$  are determined on the basis of the elements in  $Z$  (see Theorem 5 for details) Let  $S_{T2}^1(i, j)$  denote the set of all those structures in which the players  $i$  and  $t$  belong to different coalitions, and  $j$  and  $s$  belong to different coalitions. Let  $S_{T2}^2(i, j)$  denote the set of all those structures in which the players  $i$  and  $t$  belong to the same coalition, and  $j$  and  $s$  belong to the same coalition. The test  $T2(i, j)$  uses  $r^v$  to compare the value of any structure from  $S_{T2}^1(i, j)$  to the value of any structure from  $S_{T2}^2(i, j)$ .

**T3** For any  $i \in N$  and  $j \in N - \{i\}$ , the test  $T3(i, j)$  compares the values of any two structures  $\pi_1 \in \Pi^N$  and  $\pi_2 \in \Pi^N$  such that

- \* in  $\pi_1$ ,  $i$  and  $j$  are apart, and
- \* in  $\pi_2$ ,  $i$  and  $j$  are together.

Let  $S_{T_3}^1(i, j)$  denote the set of all those structures in which the players  $i$  and  $j$  belong to different coalitions. Let  $S_{T_3}^2(i, j)$  denote the set of all those structures in which the players  $i$  and  $j$  belong to the same coalition. The test  $T_3(i, j)$  uses  $r^v$  to compare the value of any structure from  $S_{T_3}^1(i, j)$  to the value of any structure from  $S_{T_3}^2(i, j)$ .

#### 4.2.2 The universes of T1, T2, T3

By definition, there are several different ways of doing tests  $T_1(i, j)$ ,  $T_2(i, j)$ , and  $T_3(i, j)$  for any  $i$  and  $j$ . The set of all these possibilities is defined in terms of the *universe* of a test.

**Definition 11** For each  $1 \leq a \leq 3$ , the set  $U_{T_a}(i, j) = S_{T_a}^1(i, j) \times S_{T_a}^2(i, j)$  is called the *universe* of  $T_a(i, j)$ .  $\square$

**Lemma 3** For each  $1 \leq a \leq 3$ , the sets  $S_{T_a}^1(i, j)$  and  $S_{T_a}^2(i, j)$  are disjoint.

**Proof** By definition of  $S_{T_a}^1(i, j)$  and  $S_{T_a}^2(i, j)$  given in Sect. 4.2.1.  $\square$

Let  $\mathbb{B}_1$  denote  $\frac{(Bell(n-1) - Bell(n-2))^2}{3}$ . Then as per the definition of  $\mathbb{B}$  given in Sect. 3.3,  $\mathbb{B}_1 > \mathbb{B}$ .

**Theorem 2** For any two distinct players  $i$  and  $j$ , there are

1.  $(Bell(n-1) - Bell(n-2))^2 = 3\mathbb{B}_1$  distinct ways of doing  $T_1(i, j)$ .
2.  $((Bell(n) - 2Bell(n-1) + Bell(n-2)) \times Bell(n-2)) > 3\mathbb{B}_1$  distinct ways of doing  $T_2(i, j)$ .
3.  $((Bell(n) - Bell(n-1)) \times Bell(n-1)) > 3\mathbb{B}_1$  distinct ways of doing  $T_3(i, j)$ .

**Proof**

1. By Lemma 3 and Theorem 19 (see Appendix A).
2. By Lemma 3 and Theorem 20 (see Appendix A).
3. From Lemma 3 and Theorem 21 (see Appendix A).

$\square$

For any  $n \geq 5$  and any two distinct players  $i$  and  $j$ , the cardinalities of the universes of  $T_1$ ,  $T_2$ , and  $T_3$  are lower bounded as per the following assertions (see Theorem 23 in Appendix A for proof):

1.  $|U_{T_1}(i, j)| > 3\mathbb{B}$
2.  $|U_{T_2}(i, j)| > 3\mathbb{B}$
3.  $|U_{T_3}(i, j)| > 3\mathbb{B}$

Theorem 22 (see Appendix A) gives the relation between the cardinalities of  $U_{T_1}(i, j)$ ,  $U_{T_2}(i, j)$ , and  $U_{T_3}(i, j)$ .

### 4.2.3 Monotonicity-satisfying relations for T1, T2, T3

For any  $1 \leq a \leq 3$ , the universe of  $Ta(i, j)$  can be partitioned into three disjoint subsets  $U_{Ta}^<(i, j)$ ,  $U_{Ta}^=(i, j)$ , and  $U_{Ta}^>(i, j)$  as follows:

- \*  $U_{Ta}^<(i, j) = \{(x, y) \in S_{Ta}^1(i, j) \times S_{Ta}^2(i, j) \mid v(x) < v(y)\}$
- \*  $U_{Ta}^=(i, j) = \{(x, y) \in S_{Ta}^1(i, j) \times S_{Ta}^2(i, j) \mid v(x) = v(y)\}$
- \*  $U_{Ta}^>(i, j) = \{(x, y) \in S_{Ta}^1(i, j) \times S_{Ta}^2(i, j) \mid v(x) > v(y)\}$

Then, the set of monotonicity-satisfying relations for a test is defined as follows:

**Definition 12** For any  $1 \leq a \leq 3$ , and any two distinct players  $i$  and  $j$ , the set of monotonicity-satisfying relations for  $Ta(i, j)$  is

$$MSR_{Ta}(i, j) = \{x \mid x \in \{<, =, >\}, r^v(\pi^1, \pi^2) = x, (\pi^1, \pi^2) \in U_{Ta}(i, j) \cap S_{MON}\} \quad \square$$

**Lemma 4** Consider any  $1 \leq a \leq 3$ . For any  $x \in \{<, =, >\}$ ,  $x \in MSR_{Ta}(i, j)$  if  $|U_{Ta}^x(i, j)| > \mathbb{B}$ , i.e.,

$$\forall x \in \{<, =, >\} \quad |U_{Ta}^x(i, j)| > \mathbb{B} \Rightarrow x \in MSR_{Ta}(i, j)$$

**Proof** By problem definition  $D < \mathbb{B}$ . So if  $|U_{Ta}^x(i, j)| > \mathbb{B}$  for any  $x$ , then at least one element of  $U_{Ta}^x(i, j)$  must belong to  $S_{MON}$ .  $\square$

**Theorem 3** Consider any  $1 \leq a \leq 3$ . For any two distinct players  $i$  and  $j$ , each one of the following assertions must true.

1. More than  $2\mathbb{B}$  elements of the universe of  $Ta(i, j)$  must obey monotonicity.
2. The set of monotonicity-satisfying relations for  $Ta(i, j)$  must be non-empty.

**Proof** The universe of  $T1(i, j)$  contains more than  $3\mathbb{B}$  elements (see Theorem 23 in Appendix A for proof). By the CSG problem definition given in Sect. 3,  $D < \mathbb{B}$ . So more than  $2\mathbb{B}$  elements of the universe of  $T1(i, j)$  must obey monotonicity.

Since  $|U_{T_1}(i, j)| > 3\mathbb{B}$  (see Theorem 23 in Appendix A), at least one of the three sets  $U_{T_1}^<(i, j)$ ,  $U_{T_1}^=(i, j)$ , or  $U_{T_1}^>(i, j)$  must contain  $\mathbb{B}$  or more elements. Otherwise,  $|U_{T_1}^<(i, j)| + |U_{T_1}^=(i, j)| + |U_{T_1}^>(i, j)| < 3\mathbb{B}$  which is impossible. If  $|U_{T_1}^x(i, j)| \geq \mathbb{B}$  for any  $x$ , then at least one element of  $U_{T_1}^x(i, j)$  must belong to  $S_{MON}$  because  $D < \mathbb{B}$ . So the set of monotonicity-satisfying relations for  $T1(i, j)$  must be non-empty.

By Theorem 23 (see Appendix A),  $|U_{T_2}(i, j)| > 3\mathbb{B}$  and  $|U_{T_3}(i, j)| > 3\mathbb{B}$ . Thus for each one of the two tests  $T2(i, j)$  and  $T3(i, j)$ , more than  $2\mathbb{B}$  elements of their universe must obey



**Table 6** Step 1: Probabilistic monotonicity induced implications

Label	Implication
$X_1$	$(i, t, 1, 1) \models \text{O} \vee (t, i, 1, 1) \models \text{O} \Rightarrow \left( \forall \pi^1, \pi^2 (\pi_1, \pi_2) \in S_{\text{MON}} \wedge \pi^1 \in S_{T_1}^1(i, j) \wedge \pi^2 \in S_{T_1}^2(i, j) \Rightarrow v(\pi^1) < v(\pi^2) \right)$
$X_2$	$(i, t, 1, 2) \models \text{O} \vee (t, i, 1, 2) \models \text{O} \Rightarrow \left( \forall \pi^1, \pi^2 (\pi_1, \pi_2) \in S_{\text{MON}} \wedge \pi^1 \in S_{T_1}^1(i, j) \wedge \pi^2 \in S_{T_1}^2(i, j) \Rightarrow v(\pi^1) > v(\pi^2) \right)$
$X_3$	$(j, s, 1, 1) \models \text{O} \vee (s, j, 1, 1) \models \text{O} \Rightarrow \left( \forall \pi^1, \pi^2 (\pi_1, \pi_2) \in S_{\text{MON}} \wedge \pi^1 \in S_{T_1}^1(i, j) \wedge \pi^2 \in S_{T_1}^2(i, j) \Rightarrow v(\pi^1) > v(\pi^2) \right)$
$X_4$	$(j, s, 1, 2) \models \text{O} \vee (s, j, 1, 2) \models \text{O} \Rightarrow \left( \forall \pi^1, \pi^2 (\pi_1, \pi_2) \in S_{\text{MON}} \wedge \pi^1 \in S_{T_1}^1(i, j) \wedge \pi^2 \in S_{T_1}^2(i, j) \Rightarrow v(\pi^1) < v(\pi^2) \right)$
$X_5$	$(i, t, 1, 2) \models \text{O} \vee (t, i, 1, 2) \models \text{O} \Rightarrow \left( \forall \pi^1, \pi^2 (\pi_1, \pi_2) \in S_{\text{MON}} \wedge \pi^1 \in S_{T_2}^1(i, j) \wedge \pi^2 \in S_{T_2}^2(i, j) \Rightarrow v(\pi^1) > v(\pi^2) \right)$
$X_6$	$(s, j, 1, 1) \models \text{O} \vee (j, s, 1, 1) \models \text{O} \Rightarrow \left( \forall \pi^1, \pi^2 (\pi_1, \pi_2) \in S_{\text{MON}} \wedge \pi^1 \in S_{T_2}^1(i, j) \wedge \pi^2 \in S_{T_2}^2(i, j) \Rightarrow v(\pi^1) < v(\pi^2) \right)$
$X_7$	$(i, t, 1, 2) \models \text{O} \vee (t, i, 1, 2) \models \text{O} \Rightarrow \left( \forall \pi^1, \pi^2 (\pi_1, \pi_2) \in S_{\text{MON}} \wedge \pi^1 \in S_{T_3}^1(i, t) \wedge \pi^2 \in S_{T_3}^2(i, t) \Rightarrow v(\pi^1) > v(\pi^2) \right)$
$X_8$	$(i, t, 1, 1) \models \text{O} \vee (t, i, 1, 1) \models \text{O} \Rightarrow \left( \forall \pi^1, \pi^2 (\pi_1, \pi_2) \in S_{\text{MON}} \wedge \pi^1 \in S_{T_3}^1(i, t) \wedge \pi^2 \in S_{T_3}^2(i, t) \Rightarrow v(\pi^1) < v(\pi^2) \right)$

monotonicity, and the set of their monotonicity-satisfying relations must be non-empty. □

Probabilistic monotonicity induces the implications  $X_1$  to  $X_8$  listed in Table 6. In order to make deductions on the basis of any implication, it is necessary to ensure that its antecedent is satisfied. Observe that each one of the implications  $X_1$  to  $X_8$  has  $\pi^1 \in S_{T_a}^1(i, j)$  (for each  $1 \leq a \leq 3$ ) and  $\pi^2 \in S_{T_a}^2(i, j)$  as part of the antecedent of  $\Rightarrow$ . Given the definitions of  $S_{T_a}^1(i, j)$  and  $S_{T_a}^2(i, j)$ , it is straightforward to find a pair  $(\pi^1, \pi^2)$  such that  $\pi^1 \in S_{T_a}^1(i, j)$  and  $\pi^2 \in S_{T_a}^2(i, j)$ . Observe also that each one of the implications  $X_1$  to  $X_8$  has  $(\pi^1, \pi^2) \in S_{MON}$  as part of the antecedent of  $\Rightarrow$ . But the key question is how can a pair  $(\pi^1, \pi^2)$  be found such that  $(\pi^1, \pi^2) \in S_{MON}$  when  $S_{MON}$  is not part of the problem input (as per the problem definition given in Sect. 3)? The answer lies in Theorem 4.

**Theorem 4** Consider any  $1 \leq w \leq 3$ . If  $D < \mathbb{B}$ , then for any two distinct players  $i$  and  $j$ , at least  $\mathbb{B}$  and at most  $3\mathbb{B} - 2$  invocations to  $r^v$  are needed to compute an element of the set of monotonicity-satisfying relations for  $Tw(i, j)$ .

**Proof** By Lemma 4,  $\forall x \in \{<, =, >\}$ ,  $|U_{Tw}^x(i, j)| > \mathbb{B}$  implies that  $x \in MSR_{Tw}(i, j)$  (i.e., at least one element of  $U_{Tw}^x(i, j)$  must belong to  $S_{MON}$ ). Further, if  $|U_{Tw}^x(i, j)| < \mathbb{B}$ , there is no guarantee that  $x \in MSR_{Tw}(i, j)$ . Now, suppose the elements of  $U_{Tw}(i, j)$  are arranged in some arbitrary sequence denoted  $u_1, u_2, \dots$ . Suppose the function  $r^v$  is applied to the elements of  $U_{Tw}(i, j)$  in the order  $u_1, u_2, \dots$ . Based on what is returned by  $r^v$ , one of the following scenarios must occur:

- Let  $a \in \{<, =, >\}$ . The function  $r^v$  returns  $r^v(u_i) = a$  for each  $1 \leq i \leq \mathbb{B}$ , so  $a \in MSR_{Tw}(i, j)$  and a monotonicity-satisfying relation is found with  $\mathbb{B}$  invocations to  $r^v$ . Note that this means that an element of  $U_{Tw}^a(i, j)$  must belong to  $S_{MON}$ .
- Let  $a, b$  and  $c$  be any three pairwise distinct elements of  $\{<, =, >\}$ . Suppose  $r^v$  is applied to the first  $3(\mathbb{B} - 1)$  elements of the sequence  $u_1, u_2, \dots$  and it is found that

$$r^v(u_i) = \begin{cases} a & \text{for each } i \text{ such that } \text{mod}(i, 3) = 0 \\ b & \text{for each } i \text{ such that } \text{mod}(i, 3) = 1 \\ c & \text{for each } i \text{ such that } \text{mod}(i, 3) = 2 \end{cases} \tag{8}$$

At this stage,  $|U_{Tw}^a(i, j)| = |U_{Tw}^b(i, j)| = |U_{Tw}^c(i, j)| = \mathbb{B} - 1$ . Next,  $r^v$  is applied to  $u_{3\mathbb{B}-2}$ . Then regardless of the outcome of  $r^v(u_{3\mathbb{B}-2})$ , one of the three cardinalities  $|U_{Tw}^a(i, j)|$ ,  $|U_{Tw}^b(i, j)|$  or  $|U_{Tw}^c(i, j)|$  must be  $\mathbb{B}$  and therefore a monotonicity-satisfying relation will be found with  $3\mathbb{B} - 2$  invocations to  $r^v$ . Say  $a$  is monotonicity-satisfying, then an element of  $U_{Tw}^a(i, j)$  must belong to  $S_{MON}$ .

- If it is neither of the above two scenarios, then a monotonicity-satisfying relation, say  $a$ , will be found with more than  $\mathbb{B}$  but fewer than  $3\mathbb{B} - 2$  applications of  $r^v$ . Consequently, an element of  $U_{Tw}^a(i, j)$  must belong to  $S_{MON}$ .

□

#### 4.2.4 Application of tests T1, T2, and T3

Those elements of  $Z$  that do not correspond to an optimal solution can be eliminated using the tests  $T1$ ,  $T2$ , and  $T3$  with suitable arguments, as characterized in Theorem 5.

**Theorem 5** *If  $Z$  contains at least two partner pairs, then  $\exists a$   $1 \leq a \leq 3$  such that at least one partner pair can be eliminated by doing  $Ta$  (with suitable arguments) at most  $3\mathbb{B} - 2$  times.*

**Proof** As per Lemma 2, any two partner pairs of  $Z$  must be in one of the following forms:

Case 1 :  $((i, t, 1, 1), (t, i, 1, 1)), ((j, s, 1, 1), (s, j, 1, 1))$

Case 2 :  $((i, t, 1, 2), (t, i, 1, 2)), ((j, s, 1, 2), (s, j, 1, 2))$

Case 3 :  $((i, t, 1, 2), (t, i, 1, 2)), ((j, s, 1, 1), (s, j, 1, 1))$

Case 4 :  $((i, t, 1, 1), (t, i, 1, 1)), ((i, t, 1, 2), (t, i, 1, 2))$

where  $i \in N$ ,  $j \in N - \{i\}$ ,  $s \in N - \{i, j\}$ , and  $t \in N - \{i, j\}$ .

Consider Case 1 which gives rise to two possibilities: either  $t = s$  or else  $t \neq s$ . No matter which one of these two possibilities holds, Theorem 3 guarantees that the set of monotonicity-satisfying relation for  $T1(i, j)$  must be non-empty. By Theorem 4, at most  $3\mathbb{B} - 2$  comparisons are needed to compute a monotonicity-satisfying relation for  $T1(i, j)$ .

- \* If either  $>$  or  $=$  is a monotonicity-satisfying relation for  $T1(i, j)$ , then the pair  $((i, t, 1, 1), (t, i, 1, 1))$  must be eliminated from  $Z$ . Because, by the implication  $X_1$  given in Table 6<sub>1</sub>, the antecedant of  $\Rightarrow$  is true but its consequent is false. This makes the consequent of  $\Rightarrow$  false. By contrapositive, the antecedant of  $\Rightarrow$  must be false. In other words,  $(i, t, 1, 1) \not\# \odot$  and  $(t, i, 1, 1) \not\# \odot$ . Consequently,  $((i, t, 1, 1), (t, i, 1, 1)) \notin Z$ . Equivalently,  $((i, t, 1, 1), (t, i, 1, 1)) \in \bar{Z}$ .
- \* If either  $<$  or  $=$  is a monotonicity-satisfying relation for  $T1(i, j)$ , then the pair  $((j, s, 1, 1), (s, j, 1, 1))$  must be eliminated from  $Z$ . Because, by the implication  $X_3$  given in Table 6<sub>2</sub>, the antecedant of  $\Rightarrow$  is true but its consequent is false. This makes the consequent of  $\Rightarrow$  false. By contrapositive, the antecedant of  $\Rightarrow$  must be false. In other words,  $(j, s, 1, 1) \not\# \odot$  and  $(s, j, 1, 1) \not\# \odot$ . Consequently,  $((j, s, 1, 1), (s, j, 1, 1)) \notin Z$ . Equivalently,  $((j, s, 1, 1), (s, j, 1, 1)) \in \bar{Z}$ .

Thus, regardless of what the monotonicity-satisfying relation for  $T1(i, j)$  is, the elimination of at least one pair of elements is guaranteed for Case 1.

Consider Case 2 which also gives rise to two possibilities: either  $t = s$  or else  $t \neq s$ . No matter which one of these two possibilities holds, Theorem 3 guarantees that the set of monotonicity-satisfying relation for  $T1(i, j)$  must be non-empty. By Theorem 4, at most  $3\mathbb{B} - 2$  comparisons are needed to compute a monotonicity relation for  $T1(i, j)$ .

- \* If either  $>$  or  $=$  is a monotonicity-satisfying relation for  $T1(i, j)$ , then the pair  $((j, s, 1, 2), (s, j, 1, 2))$  must be eliminated from  $Z$ . Because, by the implication  $X_4$  given in Table 6, the antecedent of  $\Rightarrow$  is true but its consequent is false. This makes the consequent of  $\Rightarrow$  false. By contrapositive, the antecedent of  $\Rightarrow$  must be false. In other words,  $(i, t, 1, 2) \not\models \bigcirc$  and  $(j, s, 1, 2) \not\models \bigcirc$ . Consequently,  $((j, s, 1, 2), (s, j, 1, 2)) \notin Z$ . Equivalently,  $((j, s, 1, 2), (s, j, 1, 2)) \in \bar{Z}$ .
- \* If either  $<$  or  $=$  is a monotonicity-satisfying relation for  $T1(i, j)$ , then the pair  $((i, t, 1, 2), (t, i, 1, 2))$  must be eliminated from  $Z$ . Because, by the implication  $X_2$  given in Table 6, the antecedent of  $\Rightarrow$  is true but its consequent is false. This makes the consequent of  $\Rightarrow$  false. By contrapositive, the antecedent of  $\Rightarrow$  must be false. In other words,  $(i, t, 1, 2) \not\models \bigcirc$  and  $(t, i, 1, 2) \not\models \bigcirc$ . Consequently,  $((i, t, 1, 2), (t, i, 1, 2)) \notin Z$ . Equivalently,  $((i, t, 1, 2), (t, i, 1, 2)) \in \bar{Z}$ .

Thus, regardless of what the monotonicity-satisfying relation for  $T1(i, j)$  is, the elimination of at least one pair of elements is guaranteed for Case 2.

Consider Case 3 which also gives rise to two possibilities: either  $t = s$  or else  $t \neq s$ . No matter which one of these two possibilities holds, Theorem 3 guarantees that the set of monotonicity-satisfying relation for  $T2(i, j)$  must be non-empty. By Theorem 4, at most  $3B - 2$  comparisons are needed to compute a monotonicity relation for  $T2(i, j)$ .

- \* If the monotonicity-satisfying relation for  $T2(i, j)$  is either " $<$ " or " $=$ ", then the pair  $((i, t, 1, 2), (t, i, 1, 2))$  must be eliminated from  $Z$ . Because, by the implication  $X_5$  given in Table 6, the antecedent of  $\Rightarrow$  is true but its consequent is false. This makes the consequent of  $\Rightarrow$  false. By contrapositive, the antecedent of  $\Rightarrow$  must be false. In other words,  $(i, t, 1, 2) \not\models \bigcirc$  and  $(t, i, 1, 2) \not\models \bigcirc$ .
- \* If the monotonicity-satisfying relation for  $T2(i, j)$  is either " $=$ " or " $>$ ", then the pair  $((j, s, 1, 1), (s, j, 1, 1))$  must be eliminated. Because, by the implication  $X_6$  given in Table 6, the antecedent of  $\Rightarrow$  is true but its consequent is false. This makes the consequent of  $\Rightarrow$  false. By contrapositive, the antecedent of  $\Rightarrow$  must be false. In other words,  $(j, s, 1, 1) \not\models \bigcirc$  and  $(s, j, 1, 1) \not\models \bigcirc$ .

Thus, regardless of what the monotonicity-satisfying relation for  $T1(i, j)$  is, the elimination of at least one pair of elements is guaranteed for Case 3.

Consider Case 4. By Theorem 3 guarantees that the set of monotonicity-satisfying relation for  $T3(i, t)$  must be non-empty. By Theorem 4, at most  $3B - 2$  comparisons are needed to compute a monotonicity relation for  $T3(i, t)$ .

- \* If the monotonicity-satisfying relation for  $T3(i, t)$  is either " $<$ " or " $=$ ", then the pair  $((i, t, 1, 2), (t, i, 1, 2))$  must be eliminated from  $Z$ . Because, by the implication  $X_7$  given in Table 6, the antecedent of  $\Rightarrow$  is true but its consequent is false. This makes the consequent of  $\Rightarrow$  false. By contrapositive, the antecedent of  $\Rightarrow$  must be false. In other words,  $(i, t, 1, 2) \not\models \bigcirc$  and  $(t, i, 1, 2) \not\models \bigcirc$ .
- \* If the monotonicity-satisfying relation for  $T3(i, t)$  is either " $=$ " or " $>$ ", then the pair  $((i, t, 1, 1), (t, i, 1, 1))$  must be eliminated. Because, by the implication  $X_8$  given in Table 6, the antecedent of  $\Rightarrow$  is true but its consequent is false. This makes the

consequent of  $\overset{L}{\Rightarrow}$  false. By contrapositive, the antecedent of  $\overset{L}{\Rightarrow}$  must be false. In other words,  $(i, t, 1, 1) \not\equiv \mathbb{O}$  and  $(t, i, 1, 1) \not\equiv \mathbb{O}$ .

Thus, regardless of what the monotonicity-satisfying relation for  $T3(i, t)$  is, the elimination of at least one pair of elements is guaranteed for Case 4.  $\square$

Theorem 5 forms the basis for Step 1, i.e., for computing the set  $\{\mathbb{P}_1, \mathbb{P}_2\}$  of the two top priority players and the coalitions to which they belong in  $\mathbb{O}$ . Specifically, Step 1 is achieved as follows:

- Initialize  $Z$  as per Equation 7.
- While  $Z$  contains more than one partner pair, choose a relevant test (i.e.,  $T1$ ,  $T2$ , or  $T3$  as per Theorem 5) and do it in order to eliminate from  $Z$  those elements that do not correspond to  $\mathbb{O}$ .

At the end of Step 1,  $Z$  contains one partner pair. At this stage, the set  $\{\mathbb{P}_1, \mathbb{P}_2\}$  is known although we do not know which one of these two players is  $\mathbb{P}_1$  and which one is  $\mathbb{P}_2$ . However, knowledge about the identities of  $\mathbb{P}_1$  and  $\mathbb{P}_2$  is unnecessary for computing  $\mathbb{O}$  as long as the set  $\{\mathbb{P}_1, \mathbb{P}_2\}$  is known.

### 4.3 CSG method: Step 2

At this stage, the set  $\{\mathbb{P}_1, \mathbb{P}_2\}$  comprised of the two top priority players is known, and it is also known if the two top priority players are together in  $\mathbb{O}$  or if they are apart. The identity of each  $\mathbb{P}_i$ , for  $3 \leq i \leq n$ , and its optimal coalition remains to be found. Step 2 is for finding this in a series of stages. In Stage  $i$ ,  $3 \leq i \leq n$ , the identity of  $\mathbb{P}_i$  and the coalition  $C_j^{\mathbb{O}}$  to which  $\mathbb{P}_i$  belongs will be found using the test  $T4$ . We will first describe  $T4$  and then explain how it is used.

In general, consider any stage  $k > 2$  where the identities and optimal positions of the top  $k$  priority players have already been found. Let  $\alpha(k)$  be the number of coalitions in  $\mathbb{O}_{|k}$ . Let  $Q(k) = N - \{\mathbb{P}_1, \dots, \mathbb{P}_k\}$ . There are  $|Q(k)| = n - k$  possibilities for the identity of  $\mathbb{P}_{k+1}$ , and  $\alpha(k) + 1$  possibilities for its optimal coalition. Let  $V(k)$  be a set of all these possibilities:

$$V(k) = \{(x, y) \mid x \in Q(k), y \in \{1, \dots, \alpha(k) + 1\}\} \quad (9)$$

The semantics of  $(x, y)$  is that, it is possible that  $\mathbb{P}_{k+1} = x$  and  $x$  belongs to the coalition  $C_y^{\mathbb{O}}$  in  $\mathbb{O}$ . Thus  $|V(k)| = (n - k) \times (\alpha(k) + 1)$ . The problem now is to find an  $(x, y) \in V(k)$  such that  $(x, y) \equiv \mathbb{O}$ . The test  $T4$  is for finding such an element by eliminating from  $V(k)$  those elements that do not correspond to  $\mathbb{O}$ .

The remainder of this section is organised as follows. Section 4.3.1 gives a definition of the test  $T4$ . Section 4.3.2 is a specification of the parameters of  $T4$ . Section describes how  $T4$  can be used to find an element of  $S_{MON}$ . Section 4.3.4 describes how  $T4$  can be used to determine the identities of players  $\mathbb{P}_3 \dots \mathbb{P}_\delta$  and their positions in an optimal structure.

#### 4.3.1 The test $T4$

The test  $T4$  is defined as follows. For any two distinct elements  $(a, b)$  and  $(c, d)$  of  $V(k)$ , the test  $T4(k, a, b, c, d)$  uses  $r^v$  to compare the values of any two structures  $\pi_1 \in \Pi^N$  and  $\pi_2 \in \Pi^N$  such that

- \* in  $\pi_1$ , each one of the  $k$  top priority players belongs to its respective optimal coalition, player  $a$  belongs to  $C_b$ , player  $c$  belongs to any coalition except  $C_d$ , and
- \* in  $\pi_2$ , each one of the  $k$  top priority players belongs to its respective optimal coalition, player  $c$  belongs to  $C_d$ , player  $a$  belongs to any coalition except  $C_b$ .

Let  $S_{T_4}^1(k, a, b, c, d)$  denote<sup>3</sup> the set of all those structures in which each one of the  $k$  top priority players belongs to its respective optimal coalition, the player  $a$  belongs to  $C_b$ , and the player  $c$  belongs to any coalition except  $C_d$ . Let  $S_{T_4}^2(k, a, b, c, d)$  denote the set of all those structures in which each one of the  $k$  top priority players belongs to its respective optimal coalition, the player  $c$  belongs to  $C_d$ , and the player  $a$  belongs to any coalition except  $C_b$ . Lemma 5 follows readily from the definitions of  $S_{T_4}^1(k, a, b, c, d)$  and  $S_{T_4}^2(k, a, b, c, d)$ .

**Lemma 5** *The sets  $S_{T_4}^1(k, a, b, c, d)$  and  $S_{T_4}^2(k, a, b, c, d)$  are disjoint.*

The test  $T_4(k, a, b, c, d)$  uses  $r^v$  to compare the value of any structure from  $S_{T_4}^1(k, a, b, c, d)$  to the value of any structure from  $S_{T_4}^2(k, a, b, c, d)$ .

### 4.3.2 The universe of T4

For any  $k, a, b, c$ , and  $d$ , there are several ways of doing  $T_4(k, a, b, c, d)$ . The set of all these possibilities is given by the *universe of  $T_4(k, a, b, c, d)$* .

**Definition 13** The set  $U_{T_4}(k, a, b, c, d) = S_{T_4}^1(k, a, b, c, d) \times S_{T_4}^2(k, a, b, c, d)$  is called the *universe of  $T_4(k, a, b, c, d)$* .

Lemma 6 characterises the number of different ways of doing  $T_4$ .

**Lemma 6** *For any  $2 \leq k \leq n - 1$ , any  $(a, b) \in V(k)$ , and any  $(c, d) \in V(k)$ ,*

$$|U_{T_4}(x, a, b, c, d)| = |S_{T_4}^1(k, a, b, c, d)| \times |S_{T_4}^2(k, a, b, c, d)|,$$

*and there are  $|U_{T_4}(x, a, b, c, d)|$  distinct ways of doing  $T_4(k, a, b, c, d)$ .*

**Proof** By Lemma 5,  $S_{T_4}^1(x, a, b, c, d)$  and  $S_{T_4}^2(x, a, b, c, d)$  are disjoint. Thus there are  $|U_{T_4}(x, a, b, c, d)| = |S_{T_4}^1(k, a, b, c, d)| \times |S_{T_4}^2(k, a, b, c, d)|$  distinct ways of doing the test  $T_4(x, a, b, c, d)$ . □

Theorem 6 characterises a bound on the cardinality of  $U_{T_4}(x, a, b, c, d)$ .

**Theorem 6** *For any  $2 < k < n - 2$ , any  $(a, b) \in V(k)$ , and any  $(c, d) \in V(k)$ , the cardinality of  $U_{T_4}(k, a, b, c, d)$  satisfies the following relation:*

$$|U_{T_4}(k, a, b, c, d)| > Bell^2(n - k - 2)$$

**Proof** We are given that  $U_{T_4}(x, a, b, c, d) = S_{T_4}^1(k, a, b, c, d) \times S_{T_4}^2(k, a, b, c, d)$ . Now, in some of the structures in  $S_{T_4}^1(k, a, b, c, d)$ , the  $k + 1$  top priority players are

<sup>3</sup> Note that  $k, a, b, c$ , and  $d$  are the parameters of  $T_4$  and hence their respective domains are as defined in Equation 9.

split into  $\alpha(k)$  coalitions, while in others they are split into  $\alpha(k) + 1$  coalitions. Let  $S_{T_4}^{11}(k, a, b, c, d) \subset S_{T_4}^1(k, a, b, c, d)$  denote the set of structures of the former type and  $S_{T_4}^{12}(k, a, b, c, d) \subset S_{T_4}^1(k, a, b, c, d)$  that of the latter type, where  $S_{T_4}^{11}(k, a, b, c, d)$  and  $S_{T_4}^{12}(k, a, b, c, d)$  are disjoint. Consider each one of these two cases:

Case 1 The  $k + 1$  top priority players are split into  $\alpha(k)$  coalitions. The set of all such structures is  $S_{T_4}^{11}(k, a, b, c, d) \subset S_{T_4}^1(k, a, b, c, d)$ . The structures in  $S_{T_4}^{11}(k, a, b, c, d)$  are similar with regard to the positions of the  $k$  top priority players, but differ in terms of the positions of the remaining  $Q(k)$  players. There are various ways of organising the players in  $Q(k)$  whilst keeping the positions of  $\mathbb{P}_1, \dots, \mathbb{P}_k$  fixed, to get coalition structures over  $N$ . The players in  $Q(k) - \{a, c\}$  can be partitioned into  $1 \leq t \leq |Q(k)| - 2$  unordered coalitions in  $Stirling(|Q(k)| - 2, t)$  different ways<sup>4</sup>. Some of these  $t$  unordered coalitions may be merged with the coalitions  $C_1, \dots, C_{\alpha(k)}$  such that no two unordered coalitions are merged with the same coalition in  $\{C_1, \dots, C_{\alpha(k)}\}$ . Let  $0 \leq w \leq \text{Min}(\alpha(k), t)$  be the number of merged coalitions. Let  $S_{T_4}^{111}(k, a, b, c, d) \subset S_{T_4}^{11}(k, a, b, c, d)$  denote the set of all those structures in which each one of the  $k$  top priority players is in its optimal coalition,  $a \in C_b$ , and for some specific  $x \neq d, c \in C_x$ . Then

$$|S_{T_4}^{111}(k, a, b, c, d)| = |Q(k)| - 2t = 1 \sum \left( Stirling(|Q(k)| - 2, t) \times \text{Min}(\alpha(k), t)^w = 0 \right. \\ \left. \sum \left\{ \binom{\alpha(k)}{w} \binom{t}{w} w! \right\} \right) \tag{10}$$

Case 2 The  $k + 1$  top priority players are split into  $\alpha(k) + 1$  coalitions. The set of all such structures is  $S_{T_4}^{12}(k, a, b, c, d) \subset S_{T_4}^1(k, a, b, c, d)$ . The structures in  $S_{T_4}^{12}(k, a, b, c, d)$  are similar with regard to the positions of the  $k$  top priority players, but differ in terms of the positions of the remaining  $Q(k)$  players. There are various ways of organising the players in  $Q(k)$  whilst keeping the positions of  $\mathbb{P}_1, \dots, \mathbb{P}_k$  fixed, to get coalition structures over  $N$ . The players in  $Q(k) - \{a, c\}$  can be partitioned into  $1 \leq t \leq |Q(k)| - 2$  unordered coalitions in  $Stirling(|Q(k)| - 2, t)$  different ways. Some of these  $t$  unordered coalitions may be merged with the coalitions  $C_1, \dots, C_{\alpha(k)}$  such that no two unordered coalitions are merged with the same coalition in  $\{C_1, \dots, C_{\alpha(k)}\}$ . Let  $0 \leq w \leq \text{Min}(\alpha(k) + 1, t)$  be the number of merged coalitions. Let  $S_{T_4}^{121}(k, a, b, c, d) \subset S_{T_4}^{12}(k, a, b, c, d)$  denote the set of all those structures in which each one of the  $k$  top priority players is in its optimal coalition,  $a \in C_b$ , and for some specific  $x \neq d, c \in C_x$ . Then

$$|S_{T_4}^{121}(k, a, b, c, d)| = |Q(k)| - 2t = 1 \sum \left( Stirling(|Q(k)| - 2, t) \times \text{Min}(\alpha(k) + 1, t)^w = 0 \sum \left\{ \binom{\alpha(k) + 1}{w} \binom{t}{w} w! \right\} \right) \tag{11}$$

<sup>4</sup> *Stirling*( $n, i$ ) is the Stirling number of the second kind [21]. It gives the number of ways of partitioning a set  $\{1, 2, \dots, n\}$  of  $n$  labeled objects into  $i$  non-empty disjoint parts. On the other hand, the Bell number *Bell*( $n$ ) is the number of all possible partitions of  $\{1, 2, \dots, n\}$ .

From Equations 10 and 11, it is evident that  $|S_{T_4}^{111}(k, a, b, c, d)| < |S_{T_4}^{121}(k, a, b, c, d)|$ . Now, since  $\alpha(k) \geq 1$ , we get

$$\text{Min}(\alpha(k), t)w = 0 \sum \left\{ \binom{\alpha(k)}{w} \times \binom{t}{w} \times w! \right\} \geq 1.$$

This implies that

$$|S_{T_4}^{111}(k, a, b, c, d)| > |Q(k)| - 2t = 1 \sum \text{Stirling}(|Q(k)| - 2, t)$$

where  $|Q(k)| = n - k$ . Then, by the definitions of Bell and Stirling numbers,  $Bell(n) = \sum_{i=1}^n \text{Stirling}(n, i)$ . Thus,

$$|S_{T_4}^{111}(k, a, b, c, d)| > Bell(n - k - 2).$$

Since  $S_{T_4}^{111}(k, a, b, c, d) \subset S_{T_4}^{11}(k, a, b, c, d) \subset S_{T_4}^1(k, a, b, c, d)$ , we get  $|S_{T_4}^1(k, a, b, c, d)| > |S_{T_4}^{111}(k, a, b, c, d)|$  and therefore

$$|S_{T_4}^1(k, a, b, c, d)| > Bell(n - k - 2).$$

Analogously,

$$|S_{T_4}^2(k, a, b, c, d)| > Bell(n - k - 2).$$

Since  $S_{T_4}^1(k, a, b, c, d)$  and  $S_{T_4}^2(k, a, b, c, d)$  are disjoint (see Lemma 5),  $S_{T_4}^{111}(k, a, b, c, d)$  and  $S_{T_4}^{121}(k, a, b, c, d)$  are disjoint, and since  $U_{T_4}(k, a, b, c, d) = S_{T_4}^1(k, a, b, c, d) \times S_{T_4}^2(k, a, b, c, d)$ , we have:

$$|U_{T_4}(k, a, b, c, d)| > Bell^2(n - k - 2).$$

□

### 4.3.3 Monotonicity-satisfying relations for T4

For any  $2 < k \leq n - 1$ , the universe of  $T_4(k, a, b, c, d)$  can be partitioned into three disjoint subsets  $U_{T_4}^<(k, a, b, c, d)$ ,  $U_{T_4}^=(k, a, b, c, d)$ , and  $U_{T_4}^>(k, a, b, c, d)$  as follows:

- \*  $U_{T_4}^<(k, a, b, c, d) = \{(x, y) \in S_{T_4}^1(k, a, b, c, d) \times S_{T_4}^2(k, a, b, c, d) \mid v(x) < v(y)\}$
- \*  $U_{T_4}^=(k, a, b, c, d) = \{(x, y) \in S_{T_4}^1(k, a, b, c, d) \times S_{T_4}^2(k, a, b, c, d) \mid v(x) = v(y)\}$
- \*  $U_{T_4}^>(k, a, b, c, d) = \{(x, y) \in S_{T_4}^1(k, a, b, c, d) \times S_{T_4}^2(k, a, b, c, d) \mid v(x) > v(y)\}$

Then the set of monotonicity-satisfying relations for  $T_4(k, a, b, c, d)$  is defined as follows:

**Definition 14** For any  $2 < k < n$ , any  $(a, b) \in V(k)$ , and  $(c, d) \in V(k)$ , the set of monotonicity-satisfying relations for  $T_4(k, a, b, c, d)$  is

$$MSR_{T_4}(k, a, b, c, d) = \{x \mid r^v(\pi^1, \pi^2) = x, (\pi^1, \pi^2) \in U_{T_4}(k, a, b, c, d) \cap S_{\text{MON}}\} \quad \square$$

For  $2 < k < \delta$ , let  $\mathbb{B}_2(k)$  be defined as follows:



**Table 7** Probabilistic monotonicity induced implication for stage  $2 < k \leq \delta$ . These are used to make deductions based on the outcome of  $T4(k, a, b, c, d)$

Label	Implication
$X_{ab}$	$\left( (a, b) \vDash \mathbb{O} \right) \stackrel{L}{\Rightarrow} \left( \forall \pi^1, \pi^2 \ (\pi^1, \pi^2) \in S_{\text{MON}} \right.$ $\wedge \ \pi^1 \in S_{T4}^1(k, a, b, c, d) \ \wedge \ \pi^2 \in S_{T4}^2(k, a, b, c, d)$ $\stackrel{R}{\Rightarrow} v(\pi^1) > v(\pi^2) \left. \right)$
$X_{cd}$	$\left( (c, d) \vDash \mathbb{O} \right) \stackrel{L}{\Rightarrow} \left( \forall \pi^1, \pi^2 \ (\pi^1, \pi^2) \in S_{\text{MON}} \right.$ $\wedge \ \pi^1 \in S_{T4}^1(k, a, b, c, d) \ \wedge \ \pi^2 \in S_{T4}^2(k, a, b, c, d)$ $\stackrel{R}{\Rightarrow} v(\pi^1) < v(\pi^2) \left. \right)$

$$\mathbb{B}_2(k) = \frac{\text{Bell}^2(n - k - 2)}{3}.$$

**Lemma 7** For each  $2 < k < \delta$ ,

$$\mathbb{B}_1 > \mathbb{B}_2(k) > \mathbb{B}.$$

**Proof** Follows readily from the definitions of  $\mathbb{B}$ ,  $\mathbb{B}_1$ , and  $\mathbb{B}_2(k)$ . □

**Theorem 7** If  $D < \mathbb{B}$ , then for each  $2 < k \leq \delta$ , each one of the following assertions must be true:

1. More than  $2\mathbb{B}_2(k)$  elements of the universe of  $T4(k, a, b, c, d)$  must obey monotonicity.
2. The set of monotonicity-satisfying relations for  $T4(k, a, b, c, d)$  must be non-empty.

**Proof** Consider any  $2 < k \leq \delta$ . By Lemma 6, there are  $|U_{T4}(k, a, b, c, d)|$  distinct ways of doing  $T4(k, a, b, c, d)$  for any  $(a, b) \in V(k)$  and any  $(c, d) \in V(k)$ . By Theorem 6,  $|U_{T4}(k, a, b, c, d)| > \text{Bell}^2(n - k - 2)$ . That is,  $|U_{T4}(k, a, b, c, d)| > 3\mathbb{B}_2$ . By Lemma 7,  $\mathbb{B}_2(k) > \mathbb{B}$ . By the CSG problem definition of Sect. 3,  $D < \mathbb{B}$ . So  $D < \mathbb{B}_2(k)$ . Therefore, more than  $2\mathbb{B}_2$  elements of the universe of  $T4(k, a, b, c, d)$  must therefore obey monotonicity.

Since  $|U_{T4}(k, a, b, c, d)| > 3\mathbb{B}_2(k)$ , it follows that at least one of the three sets  $U_{T4}^<(k, a, b, c, d)$ ,  $U_{T4}^=(k, a, b, c, d)$ , or  $U_{T4}^>(i, j)$  must contain  $\mathbb{B}_2(k)$  or more elements. Otherwise,  $|U_{T4}^<(k, a, b, c, d)| + |U_{T4}^=(k, a, b, c, d)| + |U_{T4}^>(k, a, b, c, d)| < 3\mathbb{B}_2(k)$  which is impossible. If  $|U_{T4}^x(k, a, b, c, d)| > \mathbb{B}_2(k)$  for any  $x$ , then at least one element of  $U_{T4}^x(k, a, b, c, d)$  must belong to  $S_{\text{MON}}$  because  $D < \mathbb{B}_2(k)$ . So the set of monotonicity-satisfying relations for  $T4(k, a, b, c, d)$  must be non-empty. □

Probabilistic monotonicity induces the implications  $X_{ab}$  and  $X_{cd}$  listed in Table 7. For  $2 < k \leq \delta$ , these implications will be used to determine the identity of  $\mathbb{P}_k$  and its optimal coalition, by eliminating those elements from  $V(k)$  that do not correspond to an optimum.

In order to use these implications, a pair  $(\pi^1, \pi^2)$  must be found in the universe of  $T4$  such that  $(\pi^1, \pi^2) \in S_{MON}$ . Since  $S_{MON}$  is unknown, the question ‘how to find such a pair?’ is answered in Theorem 8.

**Theorem 8** Consider any  $2 < k \leq \delta$ , any  $(a, b) \in V(k)$ , and any  $(c, d) \in V(k)$ . If  $D < \mathbb{B}$ , then at least  $\mathbb{B}$  and at most  $3\mathbb{B} - 2$  invocations to  $r^v$  are needed to compute an element of the set of monotonicity-satisfying relations for  $T4(k, a, b, c, d)$ .

**Proof** Consider any  $2 < k \leq \delta$ , any  $(a, b) \in V(k)$ , and any  $(c, d) \in V(k)$ . We are given that  $D < \mathbb{B}$ . By Lemma 7,  $D < \mathbb{B}_2(k)$ . So for any  $x \in \{<, =, >\}$ ,  $x \in MSR_{T4}(k, a, b, c, d)$  if  $|U_{T4}^x(k, a, b, c, d)| > \mathbb{B}$ . Further, if  $|U_{T4}^x(k, a, b, c, d)| < \mathbb{B}$ , there is no guarantee that  $x \in MSR_{T4}(k, a, b, c, d)$ . Suppose the elements of  $U_{T4}(k, a, b, c, d)$  are arranged in some arbitrary sequence denoted  $u_1, u_2, \dots$ . Suppose the function  $r^v$  is applied to the elements of  $U_{T4}(k, a, b, c, d)$  in the order  $u_1, u_2, \dots$ . Based on the outcomes of these tests, one of the following scenarios will occur:

- Let  $x$  denote an element of  $\{<, =, >\}$ . The outcomes of the tests are  $r^v(u_i) = x$  for each  $1 \leq i \leq \mathbb{B}$ , so  $x \in MSR_{T4}(k, a, b, c, d)$  and a monotonicity-satisfying relation is found with  $\mathbb{B}$  invocations to  $r^v$ . Note that this means that an element of  $U_{T4}^x(k, a, b, c, d)$  must belong to  $S_{MON}$ .
- Let  $x, y$  and  $z$  be any three pairwise distinct elements of  $\{<, =, >\}$ . When  $r^v$  is applied to the first  $3(\mathbb{B} - 1)$  elements of the sequence  $u_1, u_2, \dots$  it is found that

$$r^v(u_i) = \begin{cases} x & \text{for each } i \text{ such that } \text{mod}(i, 3) = 0 \\ y & \text{for each } i \text{ such that } \text{mod}(i, 3) = 1 \\ z & \text{for each } i \text{ such that } \text{mod}(i, 3) = 2 \end{cases} \tag{12}$$

At this stage,  $|U_{T4}^x(k, a, b, c, d)| = |U_{T4}^y(k, a, b, c, d)| = |U_{T4}^z(k, a, b, c, d)| = \mathbb{B} - 1$ . Next,  $r^v$  is applied to  $u_{3\mathbb{B}-2}$ . Then regardless of the outcome of  $r^v(u_{3\mathbb{B}-2})$ , one of the three cardinalities  $|U_{T4}^x(k, a, b, c, d)|$ ,  $|U_{T4}^y(k, a, b, c, d)|$  or  $|U_{T4}^z(k, a, b, c, d)|$  must be  $\mathbb{B}$  and therefore a monotonicity-satisfying relation will be found with  $3\mathbb{B} - 2$  invocations to  $r^v$ . Say  $x$  is monotonicity-satisfying, then an element of  $U_{T4}^x(k, a, b, c, d)$  must belong to  $S_{MON}$ .

- If it is neither of the above two scenarios, then a monotonicity-satisfying relation, say  $x$ , will be found with more than  $\mathbb{B}$  but fewer than  $3\mathbb{B} - 2$  applications of  $r^v$ . Consequently, an element of  $U_{T4}^x(k, a, b, c, d)$  must belong to  $S_{MON}$ .

□

### 4.3.4 Application of test T4

**Theorem 9** Consider any  $2 < k \leq \delta$  and suppose that  $D < \mathbb{B}$ . Then if  $|V(k)| \geq 2$  and  $(a, b)$  and  $(c, d)$  are any two distinct elements of  $V(k)$ , then at least one element of  $V(k)$  can be deleted by doing  $T4(k, a, b, c, d)$  at most  $3\mathbb{B} - 2$  times.

**Proof** Consider any  $2 < k \leq \delta$ . Consider any two distinct elements  $(a, b)$  and  $(c, d)$  of  $V(k)$ . By Theorem 7, the set of monotonicity-satisfying relations for  $T4(k, a, b, c, d)$  must be

non-empty. By Theorem 8, at most  $3\mathbb{B} - 2$  invocations to  $r^v$  are needed to compute a monotonicity-satisfying relation for  $T4(k, a, b, c, d)$ .

- \* If either  $<$  or  $=$  is a monotonicity-satisfying relation for  $T4(k, a, b, c, d)$ , then element  $(a, b)$  must be eliminated from  $V(k)$ . Because, by the implication  $X_{ab}$  given in Table 7, the antecedent of  $\Rightarrow$  is true but its consequent is false. This makes the consequent of  $\Rightarrow$  false. By contrapositive, the antecedent of  $\Rightarrow$  must be false. In other words,  $(ca, b) \notin \mathbb{O}$  and  $(a, b)$  must therefore be eliminated from  $V(k)$ .
- \* If either  $>$  or  $=$  is a monotonicity-satisfying relation for  $T4(k, a, b, c, d)$ , then element  $(c, d)$  must be eliminated from  $V(k)$ . Because, by the implication  $X_{cd}$  given in Table 7, the antecedent of  $\Rightarrow$  is true but its consequent is false. This makes the consequent of  $\Rightarrow$  false. By contrapositive, the antecedent of  $\Rightarrow$  must be false. In other words,  $(c, d) \notin \mathbb{O}$  and  $(c, d)$  must therefore be eliminated from  $V(k)$ .

Thus, regardless of what the monotonicity-satisfying relation for  $T4(k, a, b, c, d)$  is, at least one element of  $V(k)$  can be deleted by doing  $T4(k, a, b, c, d)$  at most  $3\mathbb{B} - 2$  times.  $\square$

#### 4.4 The CSG method: Step 3

At this stage,  $\mathbb{P}_1, \dots, \mathbb{P}_\delta$  and  $\mathbb{O}_{1\delta}$  have been determined. An element of  $\mathbb{O}_{1\delta}^E$  with the highest value remains to be found. As per Definition 6, for non-optimal structures, probabilistic monotonicity does not hold beyond  $\delta$ . An exhaustive search is therefore needed over the space of all those structures in  $\mathbb{O}_{1\delta}^E$ .

### 5 CSG algorithm

The complete CSG method (described in Sect. 4) is summarised as Algorithm 1, the input to which is a set of players  $N$ , a mapping  $r^v$ , and a bound  $\delta$  such that

$$\mathbb{B} = \frac{Bell^2(n - \delta - 2)}{3}$$

Lines 1 to 8 constitute Step 1 (described in Sect. 4.2) of the method during which  $\mathbb{P}_1, \mathbb{P}_2$ , and  $\mathbb{O}_{12}$  are determined. In Line 1,  $Z$  is initialized. In the *while* loop of Line 2, elements from  $Z$  are eliminated by doing relevant tests. In Line 3, any two elements of  $Z$  are considered to choose a test  $Ta$  (where  $a \in 1, 2, 3$  as given in Theorem 5) to do. Then in Line 4,  $3\mathbb{B}$  arbitrary elements of the universe of  $Ta$  are generated using the method described in Appendix B. In Line 5, a monotonicity-satisfying relation is computed for  $Ta$ , and, on the basis of this relation, elements of  $Z$  are eliminated as per as per Theorem 4. Once the *while* loop is exited,  $\mathbb{P}_1, \mathbb{P}_2$ , and  $\mathbb{O}_{12}$  become known in Line 8.

Lines 9 to 18 constitute Step 2 (described in Sect. 4.3) of the method. Within the *for* loop of Line 9,  $V(k)$  is initialized. The *while* loop in Line 11 is for eliminating elements of  $V(k)$  by doing  $T4$ . In Line 12, any two elements of  $V(k)$  are considered to do  $T4$  (as given in Theorem 5). Then in Line 13,  $3\mathbb{B}$  arbitrary elements of the universe of  $T4$  are generated using the method described in Appendix B. In Line 14, a monotonicity-satisfying relation is computed for  $T4$  as per Theorem 8. Then, on the basis of this relation, elements of  $V(k)$

are eliminated as per as per Theorem 9. When the *for* loop of Line 9 is exited,  $\mathbb{P}_{|\delta}$  and  $\mathbb{O}_{|\delta}$  become known in Line 17.

Line 19 is Step 3 (described in Sect. 4.4), i.e., an exhaustive search over the space of all those structures in  $\mathbb{O}_{|\delta}^E$ .

---

**Algorithm 1** Computes optimal coalition structure

**Input:**  $N = \{1, \dots, n\}, r^v, \delta$

**Output:**  $\mathbb{P}$  and  $\mathbb{O}$

---

- 1: Initialize  $Z$
  - 2: **while**  $|Z| > 2$  **do**
  - 3: Consider any two elements of  $Z$  and choose a test  $T_a$  (where  $a \in \{1, 2, 3\}$ ) as given in Theorem 5) to do
  - 4: Generate  $3\mathbb{B}$  arbitrary elements of  $U_{T_a}$  (as described in Appendix B)
  - 5: Compute  $MSR_{T_a}$  as per Theorem 4
  - 6: Perform eliminations from  $Z$  as per Theorem 5
  - 7: **end while**
  - 8:  $\mathbb{P}_1 \leftarrow x, \mathbb{P}_2 \leftarrow y, \mathbb{O}[1] \leftarrow 1, \mathbb{O}[2] \leftarrow z$  ▷ where  $Z = \{(x, y, 1, z)\}$
  - 9: **for**  $k \leftarrow 3, \delta$  **do**
  - 10: Initialize  $V(k)$
  - 11: **while**  $|V(k)| > 1$  **do**
  - 12: Consider any two elements of  $V(k)$  and do the test  $T_4$  (as indicated in Theorem 9)
  - 13: Generate  $3\mathbb{B}$  arbitrary elements of  $U_{T_4}$  (as described in Appendix B)
  - 14: Compute  $MSR_{T_4}$  as per Theorem 8
  - 15: Perform eliminations from  $V(k)$  as per Theorem 9
  - 16: **end while**
  - 17:  $\mathbb{P}_k \leftarrow a, \mathbb{O}[k] \leftarrow b,$  ▷ where  $(a, b) \in V(k)$  is what remains in  $V(k)$  after eliminations
  - 18: **end for**
  - 19:  $\mathbb{O} \leftarrow ExhaustiveSearch(\mathbb{O}_{|\delta}^E)$
  - 20: **return**  $(\mathbb{P}_1, \dots, \mathbb{P}_\delta, \mathbb{O})$
- 

**Theorem 10** *If the degree of non-monotonicity  $D$  satisfies  $D < \mathbb{B}$  where*

$$\mathbb{B} = \frac{Bell^2(n - \delta - 2)}{3}$$

*for some  $3 < \delta \leq n - 3$  and any  $n > 7$ , then the time complexity of searching the part of the search space that satisfies probabilistic monotonicity is  $\mathcal{O}(n^3 \times ((n - \delta)!)^2)$ .*

**Proof** The time complexity of **Step 1**: Lines 1 to 8 of Algorithm 1 constitute Step 1. By Lemma 1, the set  $Z$  initially contains  $2n(n - 1)$  elements. The time taken for Line 1 is  $\mathcal{O}(n^2)$ . Since any two elements of  $Z$  may be considered, Line 3 takes constant time. Line 4,

**Table 8** Illustration of  $\mathbb{B}$  for various values of  $\delta$ 

$\delta$	$n - \delta$	$\mathbb{B}$
$n - 3$	3	1/3
$n - 4$	4	4/3
$\vdots$	$\vdots$	$\vdots$
5	$n - 5$	$\frac{Bell^2(n-7)}{3}$
4	$n - 4$	$\frac{Bell^2(n-6)}{3}$

as per Appendix B, takes  $\Theta(\mathbb{B})$  time. Line 5 (by Theorem 4 involves at most  $3\mathbb{B}$  invocations to  $r^\nu$ ), will take  $\mathcal{O}(\mathbb{B})$  time. Line 6 takes constant time. So the time to run the entire while loop of Line 2 will be  $\mathcal{O}(n^2\mathbb{B})$  (since initially  $|Z| = 2n(n - 1)$ , and each computed  $MSR_{T_a}$  will result in the elimination of at least one partner pair from  $Z$ ).

The time complexity of **Step 2**: Lines 9 to 18 of Algorithm 1 constitute Step 2. Consider any  $3 \leq k \leq \delta$ , any  $(a, b) \in V(k)$ , and any  $(c, d) \in V(k)$ . By the definition of  $V(k)$  given in Equation 9,  $|V(k)| = (n - k) \times (\alpha(k) + 1)$ . Since  $|V(k)| = (n - k) \times (\alpha(k) + 1)$ , the time taken by Line 10 will be  $\mathcal{O}(n^2)$ . Since any to elements of  $V(k)$  may be considered, Line 12 will take constant time. As per Appendix B, the time taken by Line 13 will be  $\Theta(\mathbb{B})$ . By Theorem 8, at most  $3\mathbb{B}$  invocations to  $r^\nu$  are needed to compute  $MSR_{T_4}$ . So the time to run Line 14 will be  $\mathcal{O}(\mathbb{B})$ . Once  $MSR_{T_4}$  is computed, eliminations in Line 15 (as per Theorem 9) can be performed in constant time. Since  $k < n$ ,  $\alpha(k) \leq n$ , and  $\delta < n$ , at most  $n^3 \times (3\mathbb{B} - 2)$  invocations to  $r^\nu$  will be needed to complete all iterations of the for loop. The time complexity of Step 2 is therefore  $\mathcal{O}(n^3 \times \mathbb{B})$  which is  $\mathcal{O}(n^3 \times ((n - \delta)!)^2)$ .

The time to search the search space that satisfies probabilistic monotonicity, i.e., the time to run Step 1 and Step 2 is  $\mathcal{O}(n^3 \times ((n - \delta)!)^2)$ .  $\square$

As shown in Theorem 10, the time to run the CSG algorithm over the probabilistically monotonic part of the search space is decreasing in  $\delta$ , or equivalently, increasing in the degree of non-monotonicity. Depending on  $\delta$ , the time complexity may or may not be polynomial in  $n$  (see Table 8  $\mathbb{B}$ ).

## 6 Literature review

The problem of optimally partitioning a set of players has been studied from various perspectives. From a system-wide perspective, the aim is to maximize a social welfare function. From the perspective of individual players, the aim is to find solutions that are stable [10, 11], or those that are Pareto optimal [1]. We have taken the societal perspective. Regardless of whether optimization is for individuals or for the society, the literature on optimal partitioning can broadly be divided into two categories: partitioning for deterministic environments and partitioning for stochastic environments.

*Deterministic environments*: Finding a socially optimal structure for CFGs where the value of a structure is the sum of the values of its coalitions is NP-complete [35]. Numerous approaches such as dynamic programming [30, 42], branch-and-bound [33], and hybrid methods [25] have been used to solve the CSG problem for CFGs. Ueda et al. [41] showed how concise representation schemes for characteristic functions can be used to efficiently

solve the CSG problem. Within the context of CFGs, Dang et al. [13], Chalkiadakis et al. [8], and Habib et al. [23] addressed the CSG problem for overlapping coalitions.

Compared to CFGs, there are relatively fewer solutions for the CSG problem for PFGs. Rahwan et al. [31] showed how to find optimal partitions for PFGs by restricting externalities to positive only or negative only but not mixed. Their method prunes the search space on the basis of bounds on the values of groups of coalitions. Epstein et al. [15] used a distributed approach to solve the CSG problem for PFGs with positive only or negative only externalities. Banerjee and Kraemer [4] used a branch-and-bound approach to solve the CSG problem for PFGs by restricting externalities on the basis of agent types. In [15, 31], and [4], the value of a partition is the sum of the values of its coalitions. In contrast, we consider probabilistically monotone PFGs for which the value of a structure does not necessarily have to be the sum but can be any function of the values of its coalitions.

For CFGs where the value of a coalition depends both on its members and the tasks the members execute, and the value of a structure is the sum of the values of its constituent coalitions, Prantare and Heintz [29] used a branch-and-bound approach to find an anytime solution to the CSG problem.

*Stochastic environments:* Compared to deterministic cooperative games, the literature on stochastic games is rather small. Charnes and Granot [10, 11] addressed the problem of finding stable solutions to stochastic CFGs for which the values of coalitions are not deterministic but rather random variables with given distribution functions. Suijs et al. [38] showed how certain insurance scenarios can be modelled as CFGs with stochastic pay-offs and stable solutions determined. This framework was later applied for modelling water resources [14]. A key distinction between all this work on stochastic models and ours is that they take an economic perspective and pay attention to finding stable solutions. In contrast, we take an algorithmic perspective and address the problem of finding coalition structures that are socially optimal.

For CFGs with the value of a coalition structure given by the sum of the values of its coalitions, Matsumara et al. [24] proposed a framework for probabilistic coalition structure generation. In this model, each agent can belong to no more than one coalition and there is uncertainty about the membership of an agent in a coalition. For this framework, approximation algorithms are given for finding an optimal structure. The key differences between their work and ours is that they consider CFGs, while we consider PFGs. Another difference is that they assume that the value of a coalition structure is the sum of the values of its coalitions, while we allow the value of a coalition structure to be any function of the values of its coalitions. Yet another difference is that their method requires the values of coalitions as an input. In contrast, our method does not require the values of coalitions as input nor does it require the values of coalition structures. Rather, our method only requires the relation (given by  $r^v$ ) between the values of structures.

For stochastic environments, a different strand of work on coalition formation has dealt with situations requiring coalitions to form repeatedly. This opens up the possibility of introducing learning. For CFGs, Chalkiadakis and Boutilier [7] considered scenarios where agents are typed and there is uncertainty about agent types, and proposed a Bayesian learning framework for coalition formation.

In summary, the key distinguishing features of our work are as follows. Unlike existing work, we solve the CSG problem for PFGs in a stochastic setting. Further, in the existing work on stable solutions for stochastic CFGs, the values of coalitions are random variables with known probability distribution functions. However, our CSG algorithm does not require a known probability distribution function; it is sufficient to know that the degree of non-monotonicity is within a certain bound. Yet another difference is that, unlike existing

CSG methods for PFGs, we do not restrict the value of a partition to be the sum of the values of its coalitions, but allow it to be any function of the values of its coalitions. Finally, unlike the existing CSG methods for both CFGs and PFGs, our CSG algorithm does not require the values of structures to be known; rather it is sufficient to know the relation (given by  $r^v$ ) between the values of structures.

## 7 Conclusions

In this paper, we considered the problem of optimally partitioning a set of players into disjoint exhaustive coalitions. We focussed on solving this problem in the context of probabilistically monotone partition function games with priority-ordered players. For such games, we showed how an optimum can be found knowing just that a priority ordering exists but without knowing the actual ordering. We presented a greedy algorithm for solving the problem. The time complexity of the algorithm depends on the degree of non-monotonicity. We showed how the time complexity varies with the degree of non-monotonicity.

There are various avenues for further research. We considered PFGs where the property of monotonicity is violated with a certain non-zero probability. In the future, it will be interesting to consider situations where the probability that any two random structures violate monotonicity depends on the distance between them and their closest optima; those structures that are further away from an optimum are more likely to violate monotonicity than those that are closer. Another possibility is to consider situations where a player can be a member of multiple coalitions as opposed being a member of a single coalition.

## Appendix A: Theorems and proofs

**Theorem 11** *Let  $a$  and  $b$  be any two distinct players. Let  $\Pi_0$  be the set of all those structures of  $n$  players such that  $a$  and  $b$  belong to different coalitions, i.e.,*

$$\Pi_0 = \{\pi \in \Pi^N \mid \beta_\pi^a \neq \beta_\pi^b\}$$

*Then  $|\Pi_0| = \text{Bell}(n) - \text{Bell}(n - 1)$ .*

**Proof** The set  $\Pi_0$  can be written as

$$\Pi_0 = \Pi^N - \Pi_{0a}$$

where  $\Pi_{0a}$  is the set of all structures in which  $a$  and  $b$  belong to the same coalition.

$$\Pi_{0a} = \{\pi \in \Pi^N \mid \beta_\pi^a = \beta_\pi^b\}$$

Since  $a$  and  $b$  are in the same coalition, these two players may be regarded as a single player. So there will  $n - 1$  players to be arranged in coalitions. We therefore have  $|\Pi_{0a}| = \text{Bell}(n - 1)$ . Since  $|\Pi^N| = \text{Bell}(n)$ ,  $|\Pi_0| = \text{Bell}(n) - \text{Bell}(n - 1)$ .  $\square$

**Theorem 12** *Let  $K \subseteq N$  be a set of  $k$  players. Let  $\Pi_1$  be the set of all those structures of  $n$  players such that all the players in  $K$  belong to the same coalition, i.e.,*

$$\Pi_1 = \{\pi \in \Pi^N \mid a \in K, b \in K, a \neq b, \beta_\pi^a = \beta_\pi^b\}$$

Then  $|\Pi_1| = Bell(n - k + 1)$ .

**Proof** Since all the players in  $K$  belong to the same coalition, this set may be treated as a single player. This means we have  $n - k + 1$  players and therefore  $|\Pi_1| = Bell(n - k + 1)$ . □

**Theorem 13** Let  $a, b, c$ , and  $d$  be any four distinct players. Let  $\Pi_2$  be the set of all those structures of  $n$  players such that  $a$  and  $c$  are in the same coalition,  $b$  and  $d$  are in the same coalition, and  $a$  and  $b$  in different coalitions, i.e.,

$$\Pi_2 = \{\pi \in \Pi^N \mid \beta_\pi^a = \beta_\pi^c, \beta_\pi^b = \beta_\pi^d, \beta_\pi^a \neq \beta_\pi^b\}$$

Then  $|\Pi_2| = Bell(n - 2) - Bell(n - 3)$ .

**Proof** The set  $\Pi_2$  has the following composition:

$$\Pi_2 = \Pi_{2a} - \Pi_{2b}$$

where

$$\Pi_{2a} = \{\pi \in \Pi^N \mid \beta_\pi^a = \beta_\pi^c, \beta_\pi^b = \beta_\pi^d\}$$

and

$$\Pi_{2b} = \{\pi \in \Pi^N \mid \beta_\pi^a = \beta_\pi^b = \beta_\pi^c = \beta_\pi^d\}.$$

Treating  $a$  and  $c$  as a single player, and  $b$  and  $d$  as another single player, we get  $|\Pi_{2a}| = Bell(n - 4 + 2) = Bell(n - 2)$ . Then, treating  $a, b, c$ , and  $d$  as a single player, we get  $|\Pi_{2b}| = Bell(n - 4 + 1) = Bell(n - 3)$ . Thus,  $|\Pi_2| = Bell(n - 2) - Bell(n - 3)$ . □

**Theorem 14** Let  $K \subseteq N$  be a set of  $k$  players, and let  $i \notin K$ . Let  $\Pi_3$  be the set of all those structures of  $n$  players such that all the players in  $K$  belong to the same coalition, and  $i$  does not belong to that coalition, i.e.,

$$\Pi_3 = \{\pi \in \Pi^N \mid a \in K, b \in K, a \neq b, \beta_\pi^a = \beta_\pi^b, \beta_\pi^a \neq \beta_\pi^i\}$$

Then  $|\Pi_3| = Bell(n - k + 1) - Bell(n - k)$ .

**Proof** Let  $\Pi_{3a}$  be the set of all structures in which all the players in  $K$  belong to the same coalition. Let  $\Pi_{3b}$  be the set of all structures in which all the players in  $K \cup \{i\}$  belong to the same coalition.

$$\Pi_{3a} = \{\pi \in \Pi^N \mid a \in K, b \in K, a \neq b, \beta_\pi^a = \beta_\pi^b\}$$

$$\Pi_{3b} = \{\pi \in \Pi^N \mid a \in K, b \in K, a \neq b, \beta_\pi^a = \beta_\pi^b = \beta_\pi^i\}$$

Then

$$\Pi_3 = \Pi_{3a} - \Pi_{3b}.$$



By Theorem 12,  $|\Pi_{3a}| = Bell(n - k + 1)$  and  $|\Pi_{3b}| = Bell(n - k)$ . Thus  $|\Pi_3| = Bell(n - k + 1) - Bell(n - k)$ . □

**Theorem 15** *Let  $a, b,$  and  $c$  be any three distinct players. Let  $\Pi_4$  be the set of all those structures of  $n$  players such that  $a, b,$  and  $c$  belong to three different coalitions, i.e.,*

$$\Pi_4 = \{\pi \in \Pi^N \mid \beta_\pi^a \neq \beta_\pi^b, \beta_\pi^a \neq \beta_\pi^c, \beta_\pi^b \neq \beta_\pi^c\}$$

Then,  $|\Pi_4| = Bell(n) - 3Bell(n - 1) + 2Bell(n - 2)$ .

**Proof** The set  $\Pi_4$  has the following composition:

$$\Pi_4 = \Pi_{4a} - \Pi_{4b} - \Pi_{4c}$$

where

$$\begin{aligned} \Pi_{4a} &= \{\pi \in \Pi^N \mid \beta_\pi^a \neq \beta_\pi^b\}, \\ \Pi_{4b} &= \{\pi \in \Pi^N \mid \beta_\pi^a \neq \beta_\pi^b, \beta_\pi^a = \beta_\pi^c\} \\ \Pi_{4c} &= \{\pi \in \Pi^N \mid \beta_\pi^a \neq \beta_\pi^b, \beta_\pi^b = \beta_\pi^c\} \end{aligned}$$

By Theorem 11,  $|\Pi_{4a}| = Bell(n) - Bell(n - 1)$ . By substituting  $K = \{a, c\}$  and  $i = b$  in Theorem 14, we get  $|\Pi_{4b}| = Bell(n - 1) - Bell(n - 2)$ . By substituting  $K = \{b, c\}$  and  $i = a$  in Theorem 14, we get  $|\Pi_{4c}| = Bell(n - 1) - Bell(n - 2)$ . Thus,  $|\Pi_4| = Bell(n) - 3Bell(n - 1) + 2Bell(n - 2)$ . □

**Theorem 16** *Let  $a, b, c,$  and  $d$  be any four distinct players. Let  $\Pi_5$  be the set of all those structures of  $n$  players such that  $a$  and  $b$  belong to the same coalition,  $c$  and  $d$  belong to different coalitions, and neither  $c$  nor  $d$  belongs to  $a$ 's coalition, i.e.,*

$$\Pi_5 = \{\pi \in \Pi^N \mid \beta_\pi^a = \beta_\pi^b, \beta_\pi^c \neq \beta_\pi^d, \beta_\pi^a \neq \beta_\pi^c, \beta_\pi^a \neq \beta_\pi^d\}$$

Then,  $|\Pi_5| = Bell(n - 1) - 3Bell(n - 2) + 2Bell(n - 3)$ .

**Proof** Treat  $a$  and  $b$  as a single player. Then we have  $n - 1$  players, three of whom should be in three different coalitions. By Theorem 15, we get  $|\Pi_5| = Bell(n - 1) - 3Bell(n - 2) + 2Bell(n - 3)$ . □

**Theorem 17** *Let  $a, b, c,$  and  $d$  be any four distinct players. Let  $\Pi_6$  be the set of all those structures of  $n$  players such that  $a, b, c,$  and  $d$  belong to four different coalitions, i.e.,*

$$\Pi_6 = \{\pi \in \Pi^N \mid \beta_\pi^a \neq \beta_\pi^b, \beta_\pi^a \neq \beta_\pi^c, \beta_\pi^a \neq \beta_\pi^d, \beta_\pi^b \neq \beta_\pi^c, \beta_\pi^b \neq \beta_\pi^d, \beta_\pi^c \neq \beta_\pi^d\}$$

Then  $|\Pi_6| = Bell(n) - 6Bell(n - 1) + 11Bell(n - 2) - 6Bell(n - 3)$ .

**Proof** The set  $\Pi_6$  has the following composition:

$$\Pi_6 = \Pi_{6a} - \Pi_{6b} - \Pi_{6c} - \Pi_{6d}$$

where

$$\begin{aligned} \Pi_{6a} &= \{\pi \in \Pi^N \mid \beta_\pi^a \neq \beta_\pi^b, \beta_\pi^a \neq \beta_\pi^c, \beta_\pi^b \neq \beta_\pi^c\}, \\ \Pi_{6b} &= \{\pi \in \Pi^N \mid \beta_\pi^a \neq \beta_\pi^b, \beta_\pi^a \neq \beta_\pi^c, \beta_\pi^b \neq \beta_\pi^c, \beta_\pi^a = \beta_\pi^d\}, \\ \Pi_{6c} &= \{\pi \in \Pi^N \mid \beta_\pi^a \neq \beta_\pi^b, \beta_\pi^a \neq \beta_\pi^c, \beta_\pi^b \neq \beta_\pi^c, \beta_\pi^b = \beta_\pi^d\}, \end{aligned}$$

and

$$\Pi_{6d} = \{\pi \in \Pi \mid \beta_\pi^a \neq \beta_\pi^b, \beta_\pi^a \neq \beta_\pi^c, \beta_\pi^b \neq \beta_\pi^c, \beta_\pi^c = \beta_\pi^d\}.$$

By Theorem 15,  $|\Pi_{6a}| = Bell(n) - 3Bell(n - 1) + 2Bell(n - 2)$ . Since in  $\Pi_{6b}$ ,  $a$  and  $d$  are in the same coalition, view them as a single player called  $ad$ . Then the players  $ad$ ,  $b$  and  $c$  must be in three different coalitions. Applying Theorem 15,  $|\Pi_{6b}| = Bell(n - 1) - 3Bell(n - 2) + 2Bell(n - 3)$ . In the same way,  $|\Pi_{7c}| = |\Pi_{7d}| = Bell(n - 1) - 3Bell(n - 2) + 2Bell(n - 3)$ . Thus,  $|\Pi_6| = Bell(n) - 6Bell(n - 1) + 11Bell(n - 2) - 6Bell(n - 3)$ .  $\square$

**Theorem 18** *Let  $a, b, c$ , and  $d$  be any four distinct players. Let  $\Pi_7$  be the set of all those structures of  $n$  players such that  $a$  and  $b$  belong to different coalitions, and  $c$  and  $d$  belong to different coalitions, i.e.,*

$$\Pi_7 = \{\pi \in \Pi^N \mid \beta_\pi^a \neq \beta_\pi^b, \beta_\pi^c \neq \beta_\pi^d\}$$

Then  $|\Pi_7| = Bell(n) - 2Bell(n - 1) + Bell(n - 2)$ .

**Proof** The set  $\Pi_7$  has the following composition:

$$\Pi_7 = \Pi_{7a} \cup \Pi_{7b} \cup \Pi_{7c} \cup \Pi_{7d} \cup \Pi_{7e} \cup \Pi_{7f} \cup \Pi_{7g}.$$

where

$$\begin{aligned} \Pi_{7a} &= \{\pi \in \Pi^N \mid \beta_\pi^a = \beta_\pi^c, \beta_\pi^b = \beta_\pi^d, \beta_\pi^a \neq \beta_\pi^b\}. \\ \Pi_{7b} &= \{\pi \in \Pi^N \mid \beta_\pi^a = \beta_\pi^c, \beta_\pi^b \neq \beta_\pi^d, \beta_\pi^a \neq \beta_\pi^b, \beta_\pi^a \neq \beta_\pi^d\}. \\ \Pi_{7c} &= \{\pi \in \Pi^N \mid \beta_\pi^a = \beta_\pi^d, \beta_\pi^b = \beta_\pi^c, \beta_\pi^a \neq \beta_\pi^b\}. \\ \Pi_{7d} &= \{\pi \in \Pi^N \mid \beta_\pi^a = \beta_\pi^d, \beta_\pi^b \neq \beta_\pi^c, \beta_\pi^a \neq \beta_\pi^b, \beta_\pi^a \neq \beta_\pi^c\}. \\ \Pi_{7e} &= \{\pi \in \Pi^N \mid \beta_\pi^a \neq \beta_\pi^b, \beta_\pi^a \neq \beta_\pi^d, \beta_\pi^b = \beta_\pi^c, \beta_\pi^b \neq \beta_\pi^d\}. \\ \Pi_{7f} &= \{\pi \in \Pi^N \mid \beta_\pi^a \neq \beta_\pi^b, \beta_\pi^a \neq \beta_\pi^c, \beta_\pi^b = \beta_\pi^d, \beta_\pi^b \neq \beta_\pi^c\}. \\ \Pi_{7g} &= \{\pi \in \Pi^N \mid \beta_\pi^a \neq \beta_\pi^b, \beta_\pi^a \neq \beta_\pi^c, \beta_\pi^a \neq \beta_\pi^d, \beta_\pi^b \neq \beta_\pi^c, \beta_\pi^b \neq \beta_\pi^d, \beta_\pi^c \neq \beta_\pi^d\}. \end{aligned}$$

Then

$$\Pi_7 = \Pi_{7a} + \Pi_{7b} + \Pi_{7c} + \Pi_{7d} + \Pi_{7e} + \Pi_{7f} + \Pi_{7g}$$

By Theorem 13,  $|\Pi_{7a}| = |\Pi_{7c}| = Bell(n - 2) - Bell(n - 3)$ . By Theorem 16,  $|\Pi_{7b}| = |\Pi_{7d}| = |\Pi_{7e}| = |\Pi_{7f}| = Bell(n - 1) - 3Bell(n - 2) + 2Bell(n - 3)$ . By Theorem 17,  $|\Pi_{7g}| = Bell(n) - 6Bell(n - 1) + 11Bell(n - 2) - 6Bell(n - 3)$ . Thus,  $|\Pi_7| = Bell(n) - 2Bell(n - 1) + Bell(n - 2)$ .  $\square$

**Theorem 19** *For any  $i \in N$ , and any  $j \in N - \{i\}$ , the cardinalities of  $S_{T1}^1(i, j)$ ,  $S_{T1}^2(i, j)$ , and  $U_{T1}(i, j)$  are:*

$$\begin{aligned}
 |S_{T_1}^1(i, j)| &= Bell(n - 1) - Bell(n - 2) \\
 |S_{T_1}^2(i, j)| &= Bell(n - 1) - Bell(n - 2) \\
 |U_{T_1}(i, j)| &= \left( Bell(n - 1) - Bell(n - 2) \right)^2
 \end{aligned}$$

**Proof** For any  $i \in N$  and  $j \in N - \{i\}$ , the test  $T_1(i, j)$  (see Sect. 4.2.1) compares the values of any two structures  $\pi_1$  and  $\pi_2$  such that

- \* in  $\pi_1$ , the players  $i$  and  $t$  (where  $t$  is an arbitrary element of  $N - \{i, j\}$  and is determined on the basis of the elements in  $Z$ ) belong to different coalitions but  $j$  and  $s$  (where  $s$  is an arbitrary element of  $N - \{i, j\}$  and is determined on the basis of the elements in  $Z$ ) belong to the same coalition, and
- \* in  $\pi_2$ , the players  $i$  and  $t$  belong to the same coalition but  $j$  and  $s$  belong to different coalitions.

$S_{T_1}^1(i, j)$  denotes the set of all those structures in which the players  $i$  and  $t$  belong to different coalitions but  $j$  and  $s$  belong to the same coalition.  $S_{T_1}^2(i, j)$  denotes the set of all those structures in which the players  $i$  and  $t$  belong to the same coalition but  $j$  and  $s$  belong to different coalitions.

Since there is no constraint on whether  $s = t$  or  $s \neq t$ , must consider both cases.

- **The case  $s \neq t$ :** The set  $S_{T_1}^1(i, j)$  has the following composition:

$$S_{T_1}^1(i, j) = \Pi_{T_{1a}} \cup \Pi_{T_{1b}} \cup \Pi_{T_{1c}}$$

where

$$\begin{aligned}
 \Pi_{T_{1a}} &= \{ \pi \in \Pi^N \mid \beta_\pi^i \neq \beta_\pi^t, \beta_\pi^i \neq \beta_\pi^j, \beta_\pi^t \neq \beta_\pi^j, \beta_\pi^j = \beta_\pi^s \} \\
 \Pi_{T_{1b}} &= \{ \pi \in \Pi^N \mid \beta_\pi^i \neq \beta_\pi^t, \beta_\pi^i = \beta_\pi^j = \beta_\pi^s \} \\
 \Pi_{T_{1c}} &= \{ \pi \in \Pi^N \mid \beta_\pi^i \neq \beta_\pi^t, \beta_\pi^i = \beta_\pi^j = \beta_\pi^s \}
 \end{aligned}$$

By Theorem 16,  $|\Pi_{T_{1a}}| = Bell(n - 1) - 3Bell(n - 2) + 2Bell(n - 3)$ .  $|\Pi_{T_{1b}}|$  is the number of structures in which  $t, j$ , and  $s$  belong the same coalition, less the number of structures in which  $i, t, j$ , and  $s$  all belong to the same coalition. By Theorem 12 we get  $|\Pi_{T_{1b}}| = Bell(n - 2) - Bell(n - 3)$ . Analogously,  $|\Pi_{T_{1c}}| = Bell(n - 2) - Bell(n - 3)$ . Thus,  $S_{T_1}^1(i, j) = Bell(n - 1) - Bell(n - 2)$ .

Analogously,  $S_{T_1}^2(i, j) = Bell(n - 1) - Bell(n - 2)$ , so  $|U_{T_1}(i, j)| = (Bell(n - 1) - Bell(n - 2))^2$ .

- **The case  $s = t$ :** The set  $S_{T_1}^1(i, j)$  has the following composition:

$$S_{T_1}^1(i, j) = \{ \pi \in \Pi^N \mid \beta_\pi^i \neq \beta_\pi^s, \beta_\pi^j = \beta_\pi^s \}$$

By Theorem 14,  $|S_{T_1}^1(i, j)| = Bell(n - 1) - Bell(n - 2)$ . Next, the set  $S_{T_1}^2(i, j)$  has the following composition:

$$S_{T_1}^2(i, j) = \{ \pi \in \Pi^N \mid \beta_\pi^i = \beta_\pi^s, \beta_\pi^j \neq \beta_\pi^s \}$$

By Theorem 14,  $|S_{T_1}^2(i, j)| = Bell(n - 1) - Bell(n - 2)$ . Thus  $|U_{T_1}(i, j)| = (Bell(n - 1) - Bell(n - 2))^2$ . □

**Theorem 20** For any  $i \in N$ , and any  $j \in N - \{i\}$ , the cardinalities of  $S_{T_2}^1(i, j)$ ,  $S_{T_2}^2(i, j)$ , and  $U_{T_2}(i, j)$  are:

$$\begin{aligned} |S_{T_2}^1(i, j)| &= Bell(n) - 2Bell(n - 1) + Bell(n - 2) \\ |S_{T_2}^2(i, j)| &= Bell(n - 2) \\ |U_{T_2}(i, j)| &= (Bell(n) - 2Bell(n - 1) + Bell(n - 2)) \times Bell(n - 2) \end{aligned}$$

**Proof** For any  $i \in N$  and  $j \in N - \{i\}$ , the test  $T2(i, j)$  (see Sect. 4.2.1) compares the values of any two structures  $\pi_1$  and  $\pi_2$  such that

- \* in  $\pi_1$ , the players  $i$  and  $t$  are apart, and the players  $j$  and  $s$  are apart, and
- \* in  $\pi_2$ , the players  $i$  and  $t$  are together, and the players  $j$  and  $s$  are together.

$S_{T_2}^1(i, j)$  denotes the set of all those structures in which the players  $i$  and  $t$  are apart, and  $j$  and  $s$  are apart.  $S_{T_2}^2(i, j)$  denotes the set of all those structures in which the players  $i$  and  $t$  belong to the same coalition, and  $j$  and  $s$  belong to the same coalition. Since there is no constraint on whether  $s = t$  or  $s \neq t$ , must consider both cases.

• **The case  $s \neq t$ :**

By Theorem 18,  $|S_{T_2}^1(i, j)| = Bell(n) - 2Bell(n - 1) + Bell(n - 2)$ .

The set  $S_{T_2}^2(i, j)$  has the following composition:

$$S_{T_2}^2(i, j) = \Pi_{T_{2a}} \cup \Pi_{T_{2b}}$$

where

$$\begin{aligned} \Pi_{T_{2a}} &= \{\pi \in \Pi^N \mid \beta_\pi^i = \beta_\pi^t = \beta_\pi^j = \beta_\pi^s\} \\ \Pi_{T_{2b}} &= \{\pi \in \Pi^N \mid \beta_\pi^i = \beta_\pi^t, \beta_\pi^j = \beta_\pi^s, \beta_\pi^i \neq \beta_\pi^j\}. \end{aligned}$$

Consider  $|\Pi_{T_{2a}}|$ . Substiting  $k = 4$  in Theorem 12, we get  $|\Pi_{T_{2a}}| = Bell(n - 3)$ . Consider  $|\Pi_{T_{2b}}|$ . By Theorem 13,  $|\Pi_{T_{2b}}| = Bell(n - 2) - Bell(n - 3)$ . Add  $|\Pi_{T_{2a}}|$  and  $|\Pi_{T_{2b}}|$  to get  $|S_{T_2}^2(i, j)| = Bell(n - 2)$ . So  $|U_{T_2}(i, j)| = (Bell(n) - 2Bell(n - 1) + Bell(n - 2)) \times Bell(n - 2)$ .

• **The case  $s = t$ :** The set  $S_{T_2}^1(i, j)$  has the following composition:

$$S_{T_2}^1(i, j) = \Pi_{T_{2a}} \cup \Pi_{T_{2b}}$$

where

$$\begin{aligned} \Pi_{T_{2a}} &= \{\pi \in \Pi^N \mid \beta_\pi^i \neq \beta_\pi^s, \beta_\pi^i \neq \beta_\pi^j, \beta_\pi^s \neq \beta_\pi^j\} \\ \Pi_{T_{2b}} &= \{\pi \in \Pi^N \mid \beta_\pi^i = \beta_\pi^j, \beta_\pi^i \neq \beta_\pi^s\}. \end{aligned}$$

By Theorem 15, we have  $|\Pi_{T_{2a}}| = Bell(n) - 3Bell(n - 1) + 2Bell(n - 2)$ . By Theorem 14, we have  $|\Pi_{T_{2b}}| = Bell(n - 1) - Bell(n - 2)$ . So

$|S_{T_2}^1(i, j)| = Bell(n) - 2Bell(n - 1) + Bell(n - 2)$ . Next, the set  $S_{T_2}^2(i, j)$  has the following composition:

$$S_{T_2}^2(i, j) = \{ \pi \in \Pi^N \mid \beta_\pi^i = \beta_\pi^j = \beta_\pi^n \}$$

By Theorem 12  $k = 3$ , we have  $|S_{T_2}^2(i, j)| = Bell(n - 2)$ . So  $|U_{T_2}(i, j)| = (Bell(n) - 2Bell(n - 1) + Bell(n - 2)) \times Bell(n - 2)$ . □

**Theorem 21** For any  $i \in N$ , and any  $j \in N - \{i\}$ , the cardinalities of  $S_{T_3}^1(i, j)$ ,  $S_{T_3}^2(i, j)$ , and  $U_{T_3}(i, j)$  are:

$$\begin{aligned} |S_{T_3}^1(i, j)| &= Bell(n) - Bell(n - 1) \\ |S_{T_3}^2(i, j)| &= Bell(n - 1) \\ |U_{T_3}(i, j)| &= (Bell(n) - Bell(n - 1)) \times Bell(n - 1) \end{aligned}$$

**Proof** For any  $i \in N$  and  $j \in N - \{i\}$ , the test  $T_2(i, j)$  (see Sect. 4.2.1) compares the values of any two structures  $\pi_1$  and  $\pi_2$  such that

- \* in  $\pi_1$ , the players  $i$  and  $j$  are apart, and
- \* in  $\pi_2$ , the players  $i$  and  $j$  are together.

$S_{T_3}^1(i, j)$  denotes the set of all those structures in which the players  $i$  and  $j$  are apart.  $S_{T_3}^2(i, j)$  denotes the set of all those structures in which the players  $i$  and  $j$  are together.

By Theorem 11,  $|S_{T_3}^1(i, j)| = Bell(n) - Bell(n - 1)$ . Substituting  $k = 2$  in Theorem 12,  $|S_{T_3}^2(i, j)| = Bell(n - 1)$ . Thus,  $|U_{T_3}(i, j)| = (Bell(n) - Bell(n - 1)) \times Bell(n - 1)$ . □

**Theorem 22** The cardinalities of the universes of  $T_1$ ,  $T_2$ , and  $T_3$  are related as follows:

1. For any  $n$  and any two distinct players  $i$  and  $j$ ,  $|U_{T_2}(i, j)| > |U_{T_1}(i, j)|$ .
2. For any  $n$  and any two distinct players  $i$  and  $j$ ,  $|U_{T_3}(i, j)| > |U_{T_1}(i, j)|$ .
3. For any  $n$ , any two distinct players  $i$  and  $j$ ,  $|U_{T_3}(i, j)| > |U_{T_2}(i, j)|$ .
4. For any  $n > 4$ , any  $2 < k < n - 2$ , and any two distinct players  $i$  and  $j$ ,  $|U_{T_1}(i, j)| > Bell^2(n - k - 2)$ .

□

**Proof** \* For any  $n$ , here is proof that  $|U_{T_2}(i, j)| > |U_{T_1}(i, j)|$ : By Theorem 19:

$$|U_{T_1}(i, j)| = (Bell(n - 1) - Bell(n - 2))^2.$$

By Theorem 20:

$$|U_{T_2}(i, j)| = (Bell(n) - 2Bell(n - 1) + Bell(n - 2)) \times Bell(n - 2).$$

Thus,  $|U_{T_2}(i, j)| - |U_{T_1}(i, j)| = Bell(n) \times Bell(n - 2) - Bell^2(n - 1)$ . Since  $Bell(n) \times Bell(n - 2) > Bell^2(n - 1)$ ,  $|U_{T_2}(i, j)| > |U_{T_1}(i, j)|$ . Since

\* For any  $n$ , here is proof that  $|U_{T_3}(i, j)| > |U_{T_1}(i, j)|$ : By Theorem 19:

$$|U_{T_1}(i, j)| = (Bell(n - 1) - Bell(n - 2))^2.$$

By Theorem 21,

$$|U_{T_3}(i, j)| = (Bell(n) - Bell(n - 1)) \times Bell(n - 1).$$

Clearly  $|U_{T_3}(i, j)| > |U_{T_1}(i, j)|$ .

\* For any  $n$ , here is proof that  $|U_{T_3}(i, j)| > |U_{T_2}(i, j)|$ : By Theorem 20:

$$|U_{T_2}(i, j)| = (Bell(n) - 2Bell(n - 1) + Bell(n - 2)) \times Bell(n - 2).$$

By Theorem 21,

$$|U_{T_3}(i, j)| = (Bell(n) - Bell(n - 1)) \times Bell(n - 1).$$

Clearly,  $|U_{T_3}(i, j)| > |U_{T_2}(i, j)|$ .

\* For any  $n$  and any  $2 < k < n - 2$ , here is proof that  $|U_{T_1}(i, j)| > Bell^2(n - k - 2)$ . Since  $2 < k < n - 2$  and since  $Bell^2(n - k - 2)$  is decreasing in  $k$ , it is sufficient to prove the above inequality for  $k = 2$ , i.e., it is sufficient to prove that  $|U_{T_1}(i, j)| > Bell^2(n - 4)$ . We know by Theorem 19 that  $|U_{T_1}(i, j)| = (Bell(n - 1) - Bell(n - 2))^2$ . Since  $Bell(n)$  is exponentially increasing in  $n$ , for  $n > 4$ , we get

$$Bell(n - 1) - Bell(n - 2) > Bell(n - 4).$$

Thus, we get  $|U_{T_1}(i, j)| > Bell^2(n - 4)$ . This proves that  $|U_{T_1}(i, j)| > Bell^2(n - k - 2)$ .

□

**Theorem 23** For any  $n \geq 5$ , any  $k \leq \delta$ , and any two distinct players  $i$  and  $j$ :

1.  $|U_{T_1}(i, j)| > 3\mathbb{B}$
2.  $|U_{T_2}(i, j)| > 3\mathbb{B}$
3.  $|U_{T_3}(i, j)| > 3\mathbb{B}$
4.  $|U_{T_4}(k, a, b, c, d)| > 3\mathbb{B}$  for any  $(a, b) \in V(k)$ , any  $(c, d) \in V(k)$ , and any  $k \leq \delta$ .

**Proof** By Theorem 22, for any  $n \geq 5$ , any  $i$  and  $j$ , and any  $2 < k < n - 2$ ,  $|U_{T_1}(i, j)| > Bell^2(n - k - 2)$ . Further, by Theorem 22,  $|U_{T_2}(i, j)| > |U_{T_1}(i, j)|$  and  $|U_{T_3}(i, j)| > |U_{T_1}(i, j)|$ . Since

$$\mathbb{B} = \frac{Bell^2(n - \delta - 2)}{3}$$

where  $3 < \delta \leq n - 3$ , we get  $|U_{T_1}(i, j)| > 3\mathbb{B}$ ,  $|U_{T_2}(i, j)| > 3\mathbb{B}$ , and  $|U_{T_3}(i, j)| > 3\mathbb{B}$ .

By Theorem 6,  $|U_{T_4}(k, a, b, c, d)| > Bell^2(n - k - 2)$  for any  $2 < k < n - 2$ , any  $(a, b) \in V(k)$ , and any  $(c, d) \in V(k)$ . Thus, for any  $(a, b) \in V(k)$  and any  $(c, d) \in V(k)$ ,  $|U_{T_4}(k, a, b, c, d)| > 3\mathbb{B}$  for any  $k \leq \delta$ . □

## Appendix B: Generating the universes of $T_1, T_2, T_3,$ and $T_4$

By Definition 11 and Definition 13,  $U_{Ta} = S_{Ta}^1 \times S_{Ta}^2$  for each  $1 \leq a \leq 4$ . For each  $1 \leq a \leq 3$ ,  $S_{Ta}^1$  and  $S_{Ta}^2$  are as defined in Sect. 4.2.1, and  $S_{T4}^1$  and  $S_{T4}^2$  are as defined in Sect. 4.3.2. By Theorem 4, a monotonicity-satisfying relation for  $Ta$  (for any  $1 \leq a \leq 3$ ) can be found with at most  $3\mathbb{B} - 2$  invocations to  $r^\nu$ . Then by Theorem 5, at least one element of  $Z$  can be deleted by doing  $Ta$  (with  $a$  and the arguments to  $Ta$  as defined in Theorem 5). Also, by Theorem 8, a monotonicity-satisfying relation for  $T4$  can be found with  $3\mathbb{B} - 2$  invocations to  $r^\nu$ . Then by Theorem 9, at least one element of  $V(k)$  can be deleted by doing  $T4$  (with arguments as defined in Theorem 9).

Since  $U_{Ti} = S_{Ti}^1 \times S_{Ti}^2$  for each  $1 \leq i \leq 4$ , it is sufficient to generate  $\lceil \sqrt{3\mathbb{B}} \rceil$  arbitrary elements of  $S_{Ti}^1$  and  $\lceil \sqrt{3\mathbb{B}} \rceil$  arbitrary elements of  $S_{Ti}^2$  to achieve  $|U_{Ti}| = 3\mathbb{B}$ .

In order to generate the elements of  $S_{Ti}^1$  or  $S_{Ti}^2$  for any  $i$ , it is convenient to first choose a suitable representation for coalition structures. To this end, we will represent the set of players as  $N = \{1, \dots, n\}$ . Any coalition structure over  $\{1, \dots, n\}$  will have at most  $n$  coalitions. Each coalition structure will be represented by a codeword ( $code_{cs}$ ). The codeword for a coalition structure is a vector of the form  $(C_1, C_2, \dots, C_n)$  such that player  $1 \leq x \leq n$  is in the coalition  $C_x$  in the structure. Without any loss of generality, assume that the coalitions in a structure  $\pi$  are ordered as follows:

coalition  $C_a$  will precede a coalition  $C_b$  in  $\pi$  if the smallest element of  $C_a$  is less than the smallest element of  $C_b$ .

Observe that, with coalitions ordered in this way, coalition structures have the following property. In any coalition structure, player 1 must belong to the first coalition, player 2 must belong to one of the first two coalitions, and so on. In general, if the players  $1, \dots, k$  ( $1 \leq k < n$ ) belong to the first  $1 \leq m \leq k$  non-empty coalitions, then player  $k + 1$  must belong to one of the first  $m + 1$  coalitions. For example, for a game of five players, the coalition structure  $\{\{1, 2\}, \{3\}, \{4, 5\}\}$  has the codeword:

$$code_{cs}(\{\{1, 2\}, \{3\}, \{4, 5\}\}) = (1, 1, 2, 3, 3).$$

The following notation will be used for subwords of the codeword of a coalition structure. For a codeword, the subword beginning at index  $a$  and ending at index  $b$  will be denoted  $code_{cs|a,b}$ . For example, we have:

$$code_{cs|1,3}(\{\{1, 2\}, \{3\}, \{4, 5\}\}) = (1, 1, 2).$$

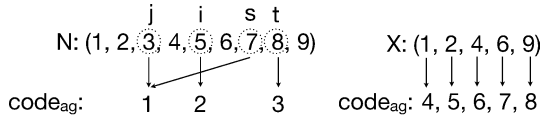
With a representation for coalition structures in place, we are now ready to describe a method (see Algorithm 2) for generating coalition structures in this representation. We will describe the procedure for generating elements of  $S_{T1}^1(i, j)$  as codewords, given  $i, j, s$ , and  $t$  as inputs (the elements of  $S_{T1}^2(i, j)$  can be generated analogously). As per Sect. 4.2.1, the set  $S_{T1}^1(i, j)$  has the following composition:

$$S_{T1}^1(i, j) = \Pi_{T1a} \cup \Pi_{T1b} \cup \Pi_{T1c}$$

and the constraints that must be satisfied by elements of  $\Pi_{T1a}$ ,  $\Pi_{T1b}$  and  $\Pi_{T1c}$  are as follows:

- $\Pi_{T1a} = \{\pi \in \Pi^N \mid \beta_\pi^i \neq \beta_\pi^t, \beta_\pi^i \neq \beta_\pi^j, \beta_\pi^t \neq \beta_\pi^j, \beta_\pi^j = \beta_\pi^s\}$
- $\Pi_{T1b} = \{\pi \in \Pi^N \mid \beta_\pi^i \neq \beta_\pi^t, \beta_\pi^i = \beta_\pi^j = \beta_\pi^s\}$

**Fig. 1** An illustration of encoding for  $N = \{1, \dots, 9\}$ ,  $i = 5$ ,  $j = 3$ ,  $s = 7$ , and  $t = 8$



	$j$	$i$	$s$	$t$
$code_{cs}$ :	(1, 2, 3, 3,	2, 4, 4, 5)		
Transformed:	(3, 2, 1, 4,	2, 4, 1, 3, 5)		

- $\Pi_{T1c} = \{\pi \in \Pi^N \mid \beta_\pi^i \neq \beta_\pi^j, \beta_\pi^i = \beta_\pi^j = \beta_\pi^s = \beta_\pi^t\}$

where  $i \in N$ ,  $j \in N - \{i\}$ ,  $s$  is an arbitrary element of  $N - \{i, j\}$  determined on the basis of the elements in  $Z$ , and  $t$  is an arbitrary element of  $N - \{i, j\}$  determined on the basis of the elements in  $Z$ .

The procedure we are going to describe is an adaptation of the method proposed in [16] for generating all possible partitions of a set  $\{1, \dots, n\}$ . Since [16] generates all possible partitions, it must be adapted to generate only those that satisfy the above listed constraints for  $\Pi_{T1a}$ ,  $\Pi_{T1b}$  and  $\Pi_{T1c}$ . The required adaptation is done as follows.

Note that, in the above listed constraints,  $s$  may or may not equal  $t$ , so need to consider both cases. However, regardless of whether  $s = t$  or not, the players  $j$  and  $s$  are required to belong to the same coalition in any coalition structure that belongs to  $\Pi_{T1a}$ ,  $\Pi_{T1b}$ , or  $\Pi_{T1c}$ . Thus,  $j$  and  $s$  may be treated as a single player. Consequently, the total number of players will now be reduced to  $n - 1$ . So coalition structures over these  $n - 1$  players must be generated such that every generated coalition structure satisfies each of the above listed constraints for  $\Pi_{T1a}$ ,  $\Pi_{T1b}$ , and  $\Pi_{T1c}$ .

Without any loss of generality, suppose  $j < s$ . Since  $j$  and  $s$  are treated as a single player, the set of players will be  $\{1, \dots, j, \dots, s - 1, s + 1, \dots, n\}$ . For convenience, we will encode the players in this set such that the numbers are all consecutive. The encoding is done by the mapping  $code_{ag} : \{1, \dots, n\} \rightarrow \{1, \dots, n - 1\}$  defined as follows (distinguish between codes ( $code_{ag}$ ) for agents and codewords ( $code_{cs}$ ) for partitions):

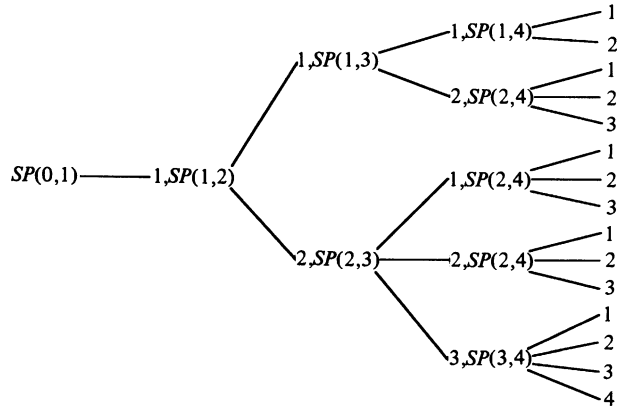
- $code_{ag}(j) = code_{ag}(s) = 1$
- $code_{ag}(i) = 2$
- $code_{ag}(t) = 3$
- $code_{ag}(X_y) = y + 3$

where  $X = N - \{i, j, s, t\}$  and the elements of  $X$  are in ascending order with  $X_y$  denoting the  $y$ th element of  $X$ . Figure 1 is an illustration of this encoding for the example  $N = \{1, \dots, 9\}$ ,  $i = 5$ ,  $j = 3$ ,  $s = 7$ , and  $t = 8$ .

With the description of encoding of agents and coalition structures in place, we are now ready to describe a procedure for generating codewords for the coalition structures in  $S_{T1}^1(i, j)$ . The method in [16] is used to generate codewords for the partitions of the  $n - 1$  element set of coded players, i.e., the set  $\{code_{ag}(1), \dots, code_{ag}(n)\}$  where  $|\{code_{ag}(1), \dots, code_{ag}(n)\}| = n - 1$ . Each generated codeword is then checked to ensure it satisfies each one of the constraints listed above for  $\Pi_{T1a}$ ,  $\Pi_{T1b}$ , and  $\Pi_{T1c}$ . The complete method is presented as Algorithm 2.



**Fig. 2** An illustration of the generation of partitions of  $N = \{1, \dots, 4\}$



In Algorithm 2, the procedure GENERATE-UNIVERSE is the main routine. The inputs to it are  $i, j, s, t, n$ , and  $count$  (the number of elements of  $S_{T1}^1$  to generate, i.e.  $count = \sqrt{3\mathbb{B}}$ ). Within GENERATE-UNIVERSE is defined a recursive routine  $SP$  (Lines 7 to 40) for generating the required codewords.

To begin, in Line 41, the list  $X$  is set to  $\{1, \dots, n\} - \{i, j, s, t\}$  (note that elements of  $X$  are to be in ascending order. The variable  $index$ , used to track the number of elements of  $S_{T1}^1$  generated so far, is initialised to zero. Then the procedure  $SP$  is invoked in Line 43.

This procedure  $SP$  (defined in Lines 7 to 40) is for generating the elements of  $S_{T1}^1$  recursively: the codewords of all partitions of the set  $\{1, \dots, n\}$  are obtained from the codewords of all partitions of the set  $\{1, \dots, n - 1\}$  by appending  $C_n$  to the respective codewords. The range of values that  $C_n$  may assume is  $1, \dots, \max(C_1, C_2, \dots, C_{n-1}) + 1$ . Note that, for  $SP(m, p)$ , the parameter  $m = \max(C_1, \dots, C_{p-1})$ , and the parameter  $p$  indicates the current index of the codeword under consideration. Figure 2 (taken from [16]) is an illustration of the recursive generation codewords for the set  $\{1, 2, 3, 4\}$ .

The *If* statement from Line 9 to 39 is for tracking the number of codewords generated so far. In the *If* statement between Lines 10 and 29, the elements of a codeword are generated one-by-one, starting from the first element and going to the  $(n - 1)$ th element. Once a complete  $n - 1$  element codeword is generated, it is transformed to a corresponding  $n$  element codeword by now treating the players  $j$  and  $s$  as distinct. The transformed codeword is saved in  $Table[index]$  (see Lines 31 to 37).

In Lines 12 to 23, checks are done to ensure that a codeword satisfies the required constraints. This check is done in Lines 12 to 16 for the case  $s \neq t$ , and in Lines 19 to 23 for the case  $s = t$ . For the case  $s \neq t$ , a constraint is violated if any one of the following conditions is true:

- o  $code_{cs|1,3} = (1, 1, 1)$
- o  $code_{cs|1,3} = (1, 2, 1)$
- o  $code_{cs|1,3} = (1, 2, 2)$

For the case  $s = t$ , a constraint is violated if the following condition is true:

- o  $code_{cs|2,2} \neq 2$

**Algorithm 2** Generates *count* number of arbitrary elements of  $S_{T1}^1$ .

**Input:**  $i, j, s, t, n, count$

**Output:** *count* number of arbitrary elements of  $S_{T1}^1$  are output in Table

---

```

1: procedure GENERATE-UNIVERSE
2:   var int  $code_{cs}[n - 1]$             $\triangleright$  Codeword for a coalition structure
3:   var int  $Table[count][n]$   $\triangleright$  This table contains the generated elements of
    $S_{T1}^1(i, j)$  as codewords
4:   var int  $i, j, s, t, n$             $\triangleright$  The inputs
5:   var int  $X[n]$             $\triangleright$  Used to contain  $\{1, \dots, n\} - \{i, j, s, t\}$  in ascending order
6:   var int  $index$             $\triangleright$  To track the number of codewords generated
7:   procedure SP(int  $m$ , int  $p$ )
8:     var int  $x$ 
9:     if  $index < count$  then
10:      if  $p \leq n - 1$  then
11:        for  $x \leftarrow 1$  to  $m$  do
12:          if  $(s \neq t)$  then
13:            if  $(p = 3)$  then
14:              if  $(code_{cs}[1 \dots 3] = (1, 1, 1)) \vee (code_{cs}[1 \dots 3] =$ 
 $(1, 2, 1)) \vee (code_{cs}[1 \dots 3] = (1, 2, 2))$  then
15:                continue            $\triangleright$  A constraint has been violated
16:              end if
17:            end if
18:          else
19:            if  $(p = 2)$  then
20:              if  $(code_{cs}[2] \neq 2)$  then
21:                continue            $\triangleright$  A constraint has been violated
22:              end if
23:            end if
24:          end if
25:           $code_{cs}[p] \leftarrow x$ 
26:          SP( $m, p + 1$ )
27:        end for
28:         $code_{cs}[p] \leftarrow m + 1$ 
29:        SP( $m + 1, p + 1$ )
30:      else
31:         $index++$ 
32:         $Table[index][j] \leftarrow Table1[index][s] \leftarrow code_{cs}[1]$ 
33:         $Table[index][i] \leftarrow code_{cs}[2]$ 
34:         $Table[index][t] \leftarrow code_{cs}[3]$ 
35:        for  $x \leftarrow 4$  to  $n - 1$  do
36:           $Table[index][X[x - 3]] \leftarrow code_{cs}[x]$ 
37:        end for
38:      end if
39:    end if
40:  end procedure SP
41:   $X \leftarrow \{1, \dots, n\} - \{i, j, s, t\}$ 
42:   $index \leftarrow 0$ 
43:  SP( $0, 1$ )            $\triangleright$  Generate constraint-satisfying partitions of  $\{1, \dots, n - 1\}$ 
44: end procedure SETPARTITIONS

```

---

Once an  $n - 1$  element constraint-satisfying codeword is generated, it is transformed to a corresponding  $n$  element codeword that represents a coalition structure over  $n$  players as follows. Let  $tcode_{cs}$  denote a transformed codeword. Transformation involves treating  $j$  and  $s$  as distinct players, and is done as follows:

- $tcode_{cs}(j) = tcode_{cs}(s) = code_{cs|1,1}$
- $tcode_{cs}(i) = code_{cs|2,2}$
- $tcode_{cs}(t) = code_{cs|3,3}$
- $tcode_{cs}(X_{x-3}) = code_{cs|x,x}$  for each  $4 \leq x \leq n - 1$ .

Codeword transformation is illustrated in Figure for the example  $N = \{1, \dots, 9\}$ ,  $i = 5$ ,  $j = 3$ ,  $s = 7$ ,  $t = 8$ , and  $code_{cs} = (1, 2, 3, 3, 2, 4, 4, 5)$ .

Each transformed codeword is saved in Table (see Lines 31 to 37). Each codeword in Table will be an  $n$  element codeword.

Elements of  $U_{Ta}(i, j)$  for each  $2 \leq a \leq 4$  can also be generated by using Algorithm 2 with constraint checking conditions suitably modified for each respective case.

We will now analyse the time complexity of generating  $3\mathbb{B}$  arbitrary elements of  $U_{T1}(i, j) = S_{T1}^1(i, j) \times S_{T1}^2(i, j)$  using Algorithm 2.

**Theorem 24** For any given  $i, j, s,$  and  $t$ ,  $3\mathbb{B}$  arbitrary elements of  $U_{Ta}(i, j)$  (for each  $1 \leq a \leq 4$ ) can be generated in  $\Theta(\sqrt{\mathbb{B}})$  time.

**Proof** Consider  $a = 1$  for which  $U_{T1}(i, j) = S_{T1}^1(i, j) \times S_{T1}^2(i, j)$ . Algorithm 2 generates  $\lceil \sqrt{3\mathbb{B}} \rceil$  elements of  $S_{Ta}^1(i, j)$ . Now, as per [16], the average time to generate a partition of  $\{1, \dots, n\}$  using Algorithm 2 is  $\Theta(1.6)$ . Since Algorithm 2 generates  $\lceil \sqrt{3\mathbb{B}} \rceil$  partitions, it will take  $\Theta(\sqrt{\mathbb{B}})$  time.

The definition of  $S_{T1}^2$  is analogous to that of  $S_{T1}^1$  (see Sect. 4.2.1). Thus,  $\lceil \sqrt{3\mathbb{B}} \rceil$  elements of  $S_{T1}^2(i, j)$  can be generated in  $\Theta(\sqrt{\mathbb{B}})$  time. Since  $U_{T1}(i, j) = S_{T1}^1(i, j) \times S_{T1}^2(i, j)$ , the time taken to generate  $3\mathbb{B}$  arbitrary elements of  $U_{T1}(i, j)$  will be  $\Theta(\mathbb{B})$ .

Elements of  $U_{Ta}(i, j)$  for each  $2 \leq a \leq 4$  can also be generated by using Algorithm 2 with constraint checking conditions suitably modified for each respective case. Thus, the time taken to generate  $3\mathbb{B}$  distinct elements of  $U_{Ta}(i, j)$  will be  $\Theta(\mathbb{B})$  for each  $2 \leq a \leq 4$ .  $\square$

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Aziz, H., & de Keijzer, B. (2011). Complexity of coalition structure generation. In *Proceedings of the 10th international joint conference on AAMAS*, pp. 191–198.

2. Aziz, H., Brandt, F., & Harrenstein, P. (2013). Pareto optimality in coalition formation. *Games and Economic Behavior*, 82, 562–581.
3. Bachrach, Y., Meir, R., Jung, K., & Kohli, P. (2010). Coalitional structure generation in skill games. In *Proceedings of AAAI*, pp. 703–708.
4. Banerjee, B., & Kraemer, L. (2010). Coalition structure generation in multi-agent systems with mixed externalities. In *Proceedings of AAMAS*, pp. 175–182.
5. Bell, E. T. (1934). Exponential numbers. *The American Mathematical Monthly*, 41(7), 411–419.
6. Chalkiadakis, G., Elkind, E., & Wooldridge, M. (2011). *Computational aspects of cooperative game theory*. Morgan & Claypool.
7. Chalkiadakis, G., & Boutilier, C. (2012). Sequentially optimal repeated coalition formation under uncertainty. *Autonomous Agents and Multi-Agent Systems*, 24, 441–484.
8. Chalkiadakis, G., Elkind, E., Markakis, E., & Jennings, N. (2010). Cooperative games with overlapping coalitions. *Journal of AI Research*, 39(1), 179–216.
9. Charnes, A., & Granot, D. (1976). Coalitional and chance-constrained solutions to  $n$ -person games I. *SIAM Journal of Applied Mathematics*, 31, 358–367.
10. Charnes, A., & Granot, D. (1977). Coalitional and chance-constrained solutions to  $n$ -person games II. *Operations Research*, 25, 1013–1019.
11. Curiel, I. (1997). *Cooperative game theory and applications*. B. V: Springer.
12. Dang, V., Dash, R., Rogers, A., & Jennings, N. (2006). Overlapping coalition formation for efficient data fusion in multi-sensor networks. In *Proceedings of the 21st National Conference on artificial intelligence (AAAI)*, pp. 635–640.
13. de Bruijn, N. (1988). *Asymptotic methods in analysis*. Dover.
14. Dinar, A., Moretti, S., Patrone, F., & Zara, S. (2006). Application of stochastic cooperative games in water resources. In R. Goetz, D. Berga (Eds.) *Frontiers in water resource economics*. Natural Resource Management and Policy, vol. 29. Springer. [https://doi.org/10.1007/0-387-30056-2\\_1](https://doi.org/10.1007/0-387-30056-2_1)
15. Epstein, D., & Bazzan, A. (2013). Distributed coalition structure generation with positive and negative externalities. In *16th Proceedings of Portuguese conference on AI*, pp. 408–419. Springer.
16. Er, M. C. (1988). A fast algorithm for generating set partitions. *The Computer Journal*, 31(3), 283–284.
17. Fatima, S., & Wooldridge, M. (2019). Computing optimal coalition structures in polynomial time. *Journal of Autonomous Agents and Multiagent Systems*, 33, 35–83.
18. Fink, A. (2006). Supply chain coordination by means of automated negotiations between autonomous agents. In B. Chaib-draa, J. Muller (Eds.) *Multiagent based supply chain management*, pp. 351–372. Springer.
19. Gawiejnowicz, S. (2020). A review of four decades of time-dependent scheduling: Main results, new topics, and open problems. *Journal of Scheduling*, 23, 3–47.
20. Gordon, V., Potts, C., Strusevich, V., & Whitehead, J. (2008). Single machine scheduling models with deterioration and learning: Handling precedence constraints via priority generation. *Journal of Scheduling*, 11, 357–370.
21. Graham, R., Knuth, D., & Patashnik, O. (1994). *Concrete mathematics*. Addison Wesley.
22. Granot, D., & Granot, F. (1992). On some network flow games. *Mathematics of Operations Research*, 17, 792–841.
23. Habib, F., Polukarov, M., & Gerding, E. (2017). Optimising social welfare in multi-resource threshold task games. In *Proceedings of principles and practice of multi-agent systems – 20th international conference*.
24. Matsumara, K., Kodric, B., Okimoto, T., & Hirayama, K. (2020). Two approximation algorithms for probabilistic coalition structure generation with quality bound. *Journal of Autonomous Agents and Multiagent Systems*, 34(Article number: 25). <https://doi.org/10.1007/s10458-020-09449-8>.
25. Michalak, T., Rahwan, T., Elkind, E., & Wooldridge, M. (2016). A hybrid exact algorithm for complete set partitioning. *AI Journal*, 230, 139–174.
26. Moyaux, T., Chaib-draa, B., & D'Amours, S. (2006). Supply chain management and multiagent systems: An overview. In B. Chaib-draa, J. Muller (Eds.) *Multiagent based supply chain management*, pp. 1–27. Springer.
27. Myerson, R. (1977). Values of games in partition function form. *International Journal of Game Theory*, 6, 23–31.
28. Nowak, A., & Radzik, T. (1994). The shapley value for  $n$ -person games in generalized characteristic function form. *Games and Economic Behavior*, 6, 150–161.
29. Prántare, F., & Heintz, F. (2020). An anytime algorithm for optimal simultaneous coalition structure generation and assignment. *Journal of Autonomous Agents and Multiagent Systems*, 34. <https://doi.org/10.1007/s10458-020-09450-1>.

30. Rahwan, T., & Jennings, N. R. (2008). An improved dynamic programming algorithm for coalition structure generation. In *Proceedings of the seventh international joint conference on autonomous agents and multiagent systems*, pp. 1417–1420.
31. Rahwan, T., Ramchurn, S., Jennings, N., & Giovanucci, A. (2009). An anytime algorithm for optimal coalition structure generation. *Journal of AI Research*, 34, 521–567.
32. Rahwan, T., Michalak, T., Wooldridge, M., & Jennings, N. (2012). Anytime coalition structure generation in multi-agent systems with positive or negative externalities. *AI Journal*, 186, 95–122.
33. Rahwan, T., Michalak, T., Wooldridge, M., & Jennings, N. (2015). Coalition structure generation: A survey. *AI Journal*, 229, 139–174.
34. Ray, D. (2007). *A game-theoretic perspective on coalition formation*. Oxford University Press.
35. Sandholm, T., Larson, K., Andersson, A., Shehory, O., & Tohmé, F. (1999). Coalition structure generation with worst case guarantees. *AI Journal*, 111, 209–238.
36. Shapley, L. (1953). A value for  $n$ -person games. In: H. Kuhn, A. Tucker (Eds.) *Contributions to the theory of games, volume II*, pp. 307–317. Princeton University Press.
37. Shrot, T., Auman, Y., & Kraus, S. (2010). On agent types in coalition formation problems. In *Proceedings of AAMAS*, pp. 757–764.
38. Suijs, J., Waegenaere, A., & Borm, P. (1998). Stochastic cooperative games in insurance. *Insurance: Mathematics and Economics*, 22(3–4), 209–228.
39. Tanaev, V., Gordon, V., & Shafransky, Y. (1994). *Scheduling theory: Single-stage systems*. Dordrecht: Kluwer.
40. Thrall, R., & Lucas, W. (1963).  $n$ -person games in partition function form. *Naval Research Logistics Quarterly*, 10, 281–298.
41. Ueda, S., Iwasaki, A., Conitzer, V., Ohta, N., Sakurai, Y., & Yokoo, M. (2018). Coalition structure generation in cooperative games with compact representations. *Journal of Autonomous Agents and Multiagent Systems*, 32, 503–533.
42. Yeh, D. (1986). A dynamic programming approach to the complete set partitioning problem. *BIT Numerical Mathematics*, 26(4), 467–474.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.