siwn.org.uk

Communications of SIWN
June 2008 Volume 4 pp. 116-122

Adaptive Coalition Structure Generation in Cooperative Multi-agent Systems

Giovanni Rossi and Gabriele D'Angelo

Department of Computer Science, University of Bologna
Mura Anteo Zamboni 7, 40126 Bologna, Italy
Email: {giorossi, gdangelo}@cs.unibo.it

Abstract: In multiagent systems a coalition structure is a collection of pair-wise disjoint subsets of agents whose union yields the entire population. Given a characteristic function quantifying the worth of agent subsets, searching for optimal coalition structures (i.e. where the sum of subsets' worth is maximal) is a well-known NP-hard combinatorial optimization problem. While existing algorithms (either deterministic or stochastic) deal with time-invariant goal functions, the focus here is on dynamic settings, where the worth of agent subsets possibly varies over time in an unknown and unpredictable fashion. The aim is to design an adaptive dynamic process generating coalition structures with high worth most of the times. To this end, detecting variations in the worth of agent subsets becomes crucial. The proposed method takes into account such (possible) changes by intensifying the exploration activity whenever they are detected. The performance with respect to the worth of optimal coalition structures is evaluated through simulations.

Keywords: Adaptive Coalition Structure Generation, Coalitional Game, Simulation, Dynamic and Non-superadditive Environment, Cooperative Multiagent System.

1. Introduction

MAS (multiagent systems) are said to be *cooperative* when agents are assumed to collaborate in order to achieve some optimal outcome of the overall system [15] [20] [21]. In this setting, a great deal of attention has been paid to coalition structure generation, where outcomes are *partitions* of agents, that is, collections of disjoint coalitions or subsets of agents, called blocks, whose union yields the entire population. Given a characteristic function CF or coalitional game, assigning a worth to each coalition, the worth of coalition structures obtains as the sum of their blocks' worth, and optimality attains where such a global worth is maximal. Searching for optimal coalition structures is a NP-hard combinatorial optimization problem [16] [17], whose generic instance consists of the 2^m-dimensional real-valued vector specifying the worth of (non-empty) coalitions, where $m \in \mathbb{N}$ is the (finite) number of agents (while $\mathbb{N}$ is the set of naturals).

A main aim of this paper is to formally organize and mathematically approach coalition structure generation in dynamic settings, where the worth of coalitions varies over time in an unknown and unpredictable fashion. In the static scenario, searching amounts to (efficiently) explore the space of candidate solutions (namely, the lattice of partitions of agents), identifying the optimal ones. Conversely, in the dynamic scenario the set of optimal coalition structures changes over time. In this case, any solver can only aim at generating as often as possible near-optimal coalition structures. In fact, a main assumption shall be that the initial goal function is not known and that no information is available concerning if, when and how such a goal function shall vary, the only available information at each time being simply the worth of (the blocks of) the coalition structure generated at that time. Hence, any solver crucially has to detect changes of the goal function and allow for re-generating coalition structures previously found to be poor when such changes do (or seem to) occur.

While in the static formulation any stochastic search method can be compared with some deterministic one [11] [16] [18], the dynamic formulation lacks benchmarks for comparisons. There is no univocally defined solution, and different solvers can only be compared, through simulation results, in terms of some performance index. The proposed mechanism identifies coalitions as decisional units, and thus it is distributed, that is, without a central authority. Although it is adjustable for coalition structure generation in the non-CF form [18], this paper focuses on the CF form. In particular, the time-varying worth of coalitions is chosen in a way such that at each time the set of optimal coalition structures and the associated maximal worth are easily determined. This is used for comparing, at each time, the worth of generated coalition structures with the worth of optimal coalition structures. The contribution is methodological, providing a solver for a novel dynamic setting, which is therefore dealt with in essentially abstract terms. Such a solver is tested through simulations in the challenging scenario where the time-varying CF is double-peaked (i.e. displaying two maxima, a global and a local one) and bi-symmetric (i.e. with the population partitioned into two types and the worth of coalitions depending only on members' type).

Although this paper addresses dynamic environments, it seems worth recalling that searching for optimal partitions of a (finite) set with given CF is a problem arising in a variety of applications. Mainly, in *combinatorial auctions* (where agents are to be interpreted as goods to sell and the CF gets determined by the available bids), maximizing the revenue amounts to optimally partition the goods and sell the blocks. Similarly, in *task allocation mechanism design*, if the system has to perform a set of basic tasks each of which may be performed more or less efficiently by different coalitions, then global task performance is maximized when both agents and tasks to be performed are optimally partitioned with a bijection between these two partitions such that each block of agents performs exactly one block of tasks [7] [19]. In fact, this issue is embedded within the more general *strategic multi-agent systems* framework, where *negotiation protocols* play a fundamental role, and such an (alternative) scenario is briefly dealt with in the sequel as related work. But perhaps the broader framework where optimal partitions crucially enter the picture is *hard* (or, more generally, *fuzzy*) *clustering*, that is, a data set is to be optimally partitioned so to have, roughly speaking, maximum (minimum) diversity between (within) blocks.

2. Preliminaries

The agent set is $M := \{0, 1, \ldots, m-1\} \subset \mathbb{Z}_+$, where $\mathbb{Z}_+ = \mathbb{N} \cup \{0\}$ denotes the set of positive integers. That is, the first $m-1$ positive integers are the identifiers of agents. The set of coalitions (or subsets of agents) is $2^M = \{A : A \subseteq M\}$. A coalition structure is a partition of M, that is, a (nonempty) set P of (nonempty) and pair wise disjoint coalitions, called blocks, whose union yields M. Formally, $P = \{A_1, \ldots, A_{|P|}\}$ such that $\emptyset \neq A_h \in 2^M$ for $1 \leq k \leq |P|$ and $\emptyset = A_h \cap A_k$ for $1 \leq h < k \leq |P|$, with $|\cdot|$ denoting cardinality as usual.

Let $\mathcal{P}^M$ denote the set of coalition structures. In fact, $(\mathcal{P}^M, \wedge, \vee)$ is a *lattice*, ordered by *coarsening* $\geqslant$ and with *meet* $\wedge$ and $join$ $\vee$. That is, if $P, Q \in \mathcal{P}^M$ and $P \geqslant Q$, then every block of Q is a subset of some block of P. Also, $P \wedge Q$ is the coarsest partition finer than both P and Q, while $P \vee Q$ is the finest partition coarser than both P and Q [3]. Recursion $\mathcal{B}_0 := 1$, $\mathcal{B}_m = \sum_{0 \leq k \leq m-1} \binom{m-1}{k} \mathcal{B}_k$ yields the (m-th *Bell*) number $|\mathcal{P}^M| = \mathcal{B}_m$ of partitions of m-sets [3] [9].

A CF or coalitional game $v : 2^M \rightarrow \mathbb{R}_+, v(\emptyset) = 0$ quantifies the worth of coalitions. The worth $w(P)$ of coalition structures P is given by the sum of their blocks' worth:

$$w(P) = \sum_{A \in P} v(A) \text{ for all } P \in \mathcal{P}^M. \tag{1}$$

Let $\mathcal{P}^M_* = \{P \in \mathcal{P}^M : w(P) \geq w(Q) \text{ for all } Q \in \mathcal{P}^M\}$ contain all optimal coalition structures. Coalitional games v are 2^m-dimensional vectors $v \in \mathbb{R}^{2^m}_+$. They are *superadditive* if $v(A \cup B) \geq v(A) + v(B)$ for all $A, B \in 2^M$ such that $A \cap B = \emptyset$, while if $\geq$ is replaced with $\leq$, then v is subadditive. Apart from the chosen dynamic scenario, the concern here is with optimal coalition structures when v is neither superadditive nor subadditive [20], as if v is superadditive, then $v(M) \geq \sum_{A \in P} v(A)$ for all $P \in \mathcal{P}^M$, while if v is subadditive, then $\sum_{i \in M} v(i) \geq \sum_{A \in P} v(A)$ for all $P \in \mathcal{P}^M$. That is to say, if v is superadditive, then the coarsest (top) coalition structure $P^\top = \{M\}$ is optimal, while if v is subadditive, then the finest (bottom) coalition structure $P_\perp = \{\{0\}, \ldots, \{m-1\}\}$ is optimal. A coalitional game is *symmetric* if the worth $v(A)$ of any coalition depends only on its cardinality $|A|$. That is to say, there is some function $\gamma : \{0, 1, \ldots, m\} \rightarrow \mathbb{R}_+$ such that $v(A) = \gamma(|A|)$ for all $A \in 2^M$. In cooperative game theory, partition functions $w : \mathcal{P}^M \rightarrow \mathbb{R}_+$ are known as *global games*. If the worth of coalition structures is given by expression (1) above for some coalitional game v, then w is *additively separable* [8].

The *class* of partitions $P \in \mathcal{P}^M$ is the m-vector of positive integers $c^P = \{c^P_1, \ldots, c^P_m\} \in \mathbb{Z}^m_+$ where c^P_k is the number of k-cardinal blocks of P [3]. It is not hard to see that if global game w is *additively separated* by coalitional game v which, in turn, is symmetric (through some γ), then $c^P = c^Q \Rightarrow w(P) = w(Q)$ for all pairs $P, Q \in \mathcal{P}^M$. The number of optimal coalition structures then is large, making it easier to search on the one hand, but also more demanding to coordinate in massive systems on the other (see below).

In a static environment the worth of coalition structures (that is, the goal function) does not change in time, while in a dynamic one such a main time-invariance assumption is relaxed. Yet, apart from different possible magnitudes and frequencies of changes in the goal function, environments can be dynamic in rather different ways. In particular, a main distinction seems the following: either changes of the goal function, when they occur, constitute a novel input of the search problem, or else they have to be detected by the solver itself. In the former case, no matter how often and/or unpredictably changes do occur, the solver knows the worth of each coalition at each time. In turn, this crucially implies that over any time interval during which the $2^m - 1$ worths of coalitions are known not to vary one may well (and should reasonably) use any available static search method. Conversely, in the latter case none of such methods seems very useful, because at each time one only knows the worth of those coalitions that are blocks of the partition prevailing at that time (that is, one never knows the worth of all conceivable coalitions). Put it differently, the goal function shall vary, in general, while any static search is performed. This latter is precisely the setting under concern here.

Deterministic (static) search methods for optimal coalition structures in CF form essentially work as follows. Firstly, they explore the top, $m-1$-th, and the $m-2$-th levels of partition lattice $(\mathcal{P}^M, \wedge, \vee)$ (whose bottom level is level 0, and there are m levels). That is to say, they check $w(\{M\})$ and all the $2^{m-1} - 1$ worths $w(\{A, M \backslash A\}), \emptyset \subset A \subset M$ of 2-cardinal partitions. This (exponential) search provides an upper bound (which is m) for the ratio of the worth of optimal coalition structures and the highest checked worth. Next, searching proceeds to the lower levels of the partition lattice, as long as there is time available, and the upper bound decreases more or less rapidly depending on the chosen method [11] [16]. On the other hand, stochastic methods are compared with deterministic ones in terms of the worthiest coalition structure produced for any fixed amount of time available for the search [18]. In this respect, the former methods seem to outperform the latter, and the gap is greater when concerned with the non-CF form, where the partition function $w : \mathcal{P}^M \rightarrow \mathbb{R}_+$ is *not additively separable*, that is, admitting no CF $v : 2^M \rightarrow \mathbb{R}_+$ such that $w(P) = \sum_{A \in P} v(A)$ for all $P \in \mathcal{P}^M$ (see above).

3. Dynamic coalition structure generation

Available coalition structure generation methods explore $\mathcal{P}^M$, treated as the search space, toward optimality, along some time-pattern of partitions $P^0, P^1, \ldots, P^t, \ldots, P^T \in \mathcal{P}^M$, which may be determined through a deterministic or else stochastic rule. At each t, the worth $w(P^t)$ of the currently generated partition structure is explored (i.e. computed), and the output, at any (possibly constrained) time t, is the latest found coalition structure of maximal worth. Whatever pattern is chosen, the goal function is crucially assumed to remain fixed over time. This may be called the *static search setting*.

As for the situation where the worth v of coalitions (or, more generally, the worth w of partitions) may vary unpredictably over time, let $\mathcal{I}^t, 0 \leq t < T$ denote the information available for generating the coalition structure P^{t+1} that will prevail at time $t+1$. Minimally, the solver should be endowed with all previously generated coalition structures and associated (global) worth. That is, minimally, $\mathcal{I}^t \ni P^{t'}, w_{t'}(P^{t'})$ for $0 \leq t' \leq t$, where $w_{t'}(P^{t'})$ denotes the worth attained by the coalition structure $P^{t'}$ generated at t'. Then, a *dynamic coalition structure generation* method is a sequence $f^t, t > 0$ of mappings or rules specifying how to obtain P^t given $\mathcal{I}^{t-1}$. Concretely, such rules induce some probability distribution $f^t_{\mathcal{I}} : \mathcal{P}^M \rightarrow [0,1]$ over the partition lattice, that is, $\sum_{P \in \mathcal{P}^M} f^t_{\mathcal{I}}(P) = 1$. Finally, any such a dynamic generation method has the goal to maximize $\sum_{0 \leq t \leq T} w_t(P^t)\delta^t$, where $\delta \in (0,1]$ is some discount factor, and the whole time horizon T may or may not be known. In the latter case the goal is to maximize the $\underset{T\rightarrow\infty}{lim}$ of this summation of all the worths of generated coalition structures. In fact, let $\delta = 1$ for simplicity.

With this formalization, the static search setting corresponds to the situation where the sets $\mathcal{I}^t, t \geq 0$, together with the above minimal information, also encode that there exists a unique (i.e. time-invariant) coalitional game or CF $v : 2^M \rightarrow \mathbb{R}_+$ such that $w_t(P^t) = \sum_{A \in P^t} v(A)$ for all $0 \leq t \leq T$. This is here named the *static CF form scenario*. More sophisticatedly, one may only have the weaker information that for all $t \geq 0$ there is some $v_t : 2^M \rightarrow \mathbb{R}_+$ such that $w_t(P^t) = \sum_{A \in P^t} v_t(A)$. This is here named the *dynamic CF form scenario*, as the coalitional game v_t that additively separates the global game w_t (which, in turn, assigns a worth to coalition structure P^t) may be time-varying. Next, there is the *static non-CF form scenario*, where one knows that there exists a unique (i.e. time invariant) global game $w : \mathcal{P}^M \rightarrow \mathbb{R}_+$ such that $w_t(P^t) = w(P^t)$ for all $0 \leq t \leq T$. Finally, the *dynamic non-CF form scenario* corresponds precisely the case where the information sets $\mathcal{I}^t, t \geq 0$ are minimal in the above sense: one only knows the previously generated coalition structures and their realized worth. Hence, in dynamic scenarios (whether in CF form or not and independently from T) the available information is not sufficient for computing the worth of optimal coalition structures, even from a fully *ex post* viewpoint.

While no static search will ever evaluate twice any partition, in dynamic scenarios some constant exploration of, virtually, the whole space $\mathcal{P}^M$ seems needed, although it may be more or less intense depending on previous performance. In fact, dynamic generation algorithms should detect changes of the goal function, and allow for each partition to be generated with strictly positive probability after each such detection. Also, deterministic methods correspond to sequences of generation rules $f^t, t > 0$ inducing probabilities that place the whole unitary mass on one coalition structure: $f^t_{\mathcal{I}}(P) = 1$ for a unique $P \in \mathcal{P}^M, P = P^t$, while $f^t_{\mathcal{I}}(Q) = 0$ for all $Q \in \mathcal{P}^M$, $Q \neq P$. Otherwise, if $|\{P \in \mathcal{P}^M : f^t_{\mathcal{I}}(P) > 0\}| > 1$, that is, if more then one partition has a strictly positive probability of being generated, then the method is stochastic.

Although *metaheuristics* apply to combinatorial optimization problems and the dynamic scenarios (in CF form or not) described above do not yield a typical instance for any such a problem, a common issue is how to dynamically balance between *diversification* (i.e. exploration of the search space) and *intensification* (i.e. exploitation of accumulated search experience), so to quickly identify regions of the search space with high quality solutions and, on the other side, not to waste too much time in poor regions [5]. Two main philosophies are *trajectory methods*, which are sophistications of local search algorithms (such as tabu search, iterated local search, variable neighborhood search and simulated annealing), and *population-based methods* (such as ant colony optimization and evolutionary computation), that incorporate a learning component. In evolutionary computation, this latter component is based on recombination of previously evaluated solutions [18]. Here, basic iterated local search empowered with strictly positive probabilities of re-visiting previously generated partitions (which shall depend, in turn, on some time-varying and performance-dependent level of exploration activity) seems a suitable approach to the dynamic problem addressed here.

Roughly speaking, iterated local search for coalition structure generation outputs a sequence $\left(P^t, \mathcal{P}^M_{P_t}\right), t \geq 0$ of pairs of a partition $P^t \in \mathcal{P}^M$ and some associated neighborhood $\mathcal{P}^M_{P^t} \subset \mathcal{P}^M$ within which a search, whether deterministic or stochastic, exhaustive or partial, is conducted. Crucially, apart from the very first generated partition P^0 (and associated neighborhood), all subsequent partitions $P^t, t > 0$ (and associated neighborhoods $\mathcal{P}^M_{P^t}$) are not generated through pure randomization, but according to some higher level criterion.

3.1 Related work

When conceiving multistage coalition structure generation resulting from individual agents' decisions, many times such decisions are to be assumed driven by selfishness, in which case models often disregard the CF form, while emphasizing stability and/or bounded computational capability. For example, agents may be assumed to periodically receive tasks whose performance requires to join coalitions, and thereby to join those coalitions where their individual task is more likely to be performed and/or where they receive higher payoffs for contributing to other's tasks performance (and perhaps e-commerce is the main application of such models). From such a viewpoint, the focus is on coalition formation resulting from agents' use of *negotiation protocols* [4] [7] [10] [12] [13] [20] [22]. "A self-interested agent will choose the best strategy for itself, which cannot be explicitly imposed form outside. Therefore, the protocols need to be designed using a *noncooperative, strategic* perspective: the main question is what social outcomes follow given a protocol which *guarantees that each agent's desired local strategy is best for that agent - and thus the agent will use it*" (see [15], pp. 201-202). Conversely, in cooperative systems no negotiation whatsoever needs to be modeled, as there is no selfish behavior. Borrowing from game theory, the former setting is concerned with *conflict games*, where, roughly speaking, any outcome leaves some players unsatisfied, while the latter one is concerned with *coordination games*, where there is a non-empty set of outcomes which are the most desirable by *all* agents, and thus the issue is how to coordinate all agents so to get any of such socially optimal outcomes.

4. *Merge & split* generation

Focusing on coalition structure generation in the dynamic CF form scenario, a main feature of the proposed merge & split method is that the intensity of the exploration activity is controlled (and, in particular, dynamically fine-tuned) through a main *exploration parameter* $\alpha(t) \in (0,1)$, which shall depend on the $H \in \mathbb{N}$ global worths $w(P^{t-h}), 1 \leq h \leq H$, where H identifies the *memory* of the model. These worths are used precisely for detecting whether the goal function is likely to have recently changed or not. In fact, within each cycle $t > 0$, the performed simulations firstly compute the worth $w(P^{t-1})$ of the coalition structure inherited from the past. Secondly, they compute such a parameter $\alpha(t)$, next used as the basis upon which all randomization is carried out, depending on whether cycle t is devoted to *merging*, or else to *splitting*, or else to *pure randomization*. A merging cycle always ends by defining a coalition structure $P^t \geqslant P^{t-1}$, while a splitting cycle always ends by defining a coalition structure $P^t \leqslant P^{t-1}$ and a pure randomization cycle may well generate a coalition structure such that $P^t \ngeqslant P^{t-1} \ngeqslant P^t$. The generation process is stochastic: given the available information $\mathcal{I}^{t-1}$, all cycles t produce coalition structure P^t as the realization of a random partition according to a probability distribution $f^t_{\mathcal{I}} : \mathcal{P}^M \rightarrow [0,1]$.

Concerning $\alpha(t)$, let $t > 2H$ and consider the last H generated coalition structures $P^{t-H}, P^{t-H+1}, \ldots, P^{t-1}$. After evaluating $w(P^{t-1})$, cycle t proceeds computing $\alpha(t)$ as a function of the $2H + 2$ values $w(P^{t-h})$ and $\alpha(t-h)$ for $1 \leq h \leq H + 1$, as follows. Set $w^*_t := \max\{w(P^{t'}) : t' \leq t\}$,

$$\begin{aligned}\Delta w(P^t) &:= \Big(\frac{w(P^t)}{w^*_t} - \frac{w(P^{t-1})}{w^*_{t-1}}\Big)\Big/\Big(\frac{w(P^{t-1})}{w^*_{t-1}}\Big),\\ \Delta\alpha(t) &:= \frac{\alpha(t)-\alpha(t-1)}{\alpha(t-1)}.\end{aligned}$$

As $\sum_{1\leq h\leq H} h = \binom{H+1}{2}$ [9], let $\gamma(h,H) := \frac{\binom{H+1}{2}}{H-h+1}$ and then choose $\alpha(t)$ so to stay as close as possible to verifying

$$\sum_{1\leq h\leq H} \frac{\Delta w(P^{t-h})}{\gamma(h,H)} = \sum_{1\leq h\leq H} \frac{1-\Delta\alpha(t-h+1)}{\gamma(h,H)}. \qquad (2)$$

The exploration parameter is used in a way (described below) such that the greater $\alpha(t)$, the more the generated partition P^t is allowed to differ from the previously generated one P^{t-1}. In fact, the above expression requires that the latest value $\alpha(t)$ is determined so that the weighted average of $(1-\alpha(t-H+1)), \ldots, (1-\alpha(t))$ is as close as possible to the weighted average of $\Delta w(P^{t-H}), \ldots, \Delta w(P^{t-1})$, with more recent values having more weight than older ones. Nevertheless, while this latter average may take any value (depending on the observed worth of generated partitions), the way the exploration parameter is used requires this latter to be bounded, that is, $0 < \underline{\alpha} \leq \alpha(t) \leq \overline{\alpha} < 1$. Accordingly, if the $\alpha(t)$ that satisfies expression (2) above (there always exist a unique one) exceeds $\overline{\alpha}$, then $\alpha(t) = \overline{\alpha}$. On the other hand, if it turns out to be less than $\underline{\alpha}$, then $\alpha(t) = \underline{\alpha}$. In this way, $\alpha(t)$ is constantly updated in a way such that, roughly speaking, the more increasing the worth of the last H generated partitions, the less intense the exploration activity over the last H cycles. In particular, $\underline{\alpha} = 0.05 = 1 - \overline{\alpha}$ and, concerning initialization, $\alpha(k) = \overline{\alpha}$ for $-H \leq k \leq 1$ and $\Delta w(P^k) := 0$ for $-H \leq k \leq 0$.

The proposed method basically alternates two transitions, split and merge, but if the recent performance turns out to be particularly poor (which could well be due, in fact, to a change in the CF assigning a worth to each coalition), then a pure randomization cycle is used. In this respect, note that by construction the larger $\alpha(t)$, the smaller the worth of the most recently generated partitions $P^{t-1}, \ldots, P^{t-H}$. Accordingly, let $\hat{\alpha}$ be some threshold value such that if $\alpha(t) \geq \hat{\alpha}$, then cycle t proceeds to pure randomization, as described hereafter. Otherwise, it proceeds to splitting or else to merging depending on whether cycle $t-1$ has been a merging or else a splitting one, as detailed below. In practice, $\hat{\alpha} = 0.5$; this threshold value has been chosen (among others that have been tested), on the basis of its observed performance in preliminary simulations. In fact, defining this threshold value as a function of the available information, that is, $\hat{\alpha} = \hat{\alpha}_t = \hat{\alpha}_t(\mathcal{I}_{t-1})$, and according to a higher level criterion is a conceivable a future development.

Pure randomization. Starting from some exogenous initial partition P^0, the first cycle (i.e. cycle 1) is a pure randomization one. In fact, given initialization, $\alpha(1) > 0.5$ (see above). Also, as initially there is no available information, this seems the most natural and reasonable choice. Simulations always start with the finest partition $P^0 = \{\{0\}, \{1\}, \ldots, \{m-1\}\}$. All successive pure randomization cycles $t > 1$ work exactly the same as cycle 1, inheriting some partition P^{t-1} from the past. Hence, this cycle type is described for generic t, given $P^{t-1} = \{A^{t-1}_1, \ldots, A^{t-1}_{|P^{t-1}|}\}$. After computing global worth $w(P^{t-1})$ and $\alpha(t) \in [\underline{\alpha}, 1-\underline{\alpha}]$ accordingly (see above), if $\alpha(t) \geq \hat{\alpha}$, then the pure randomization cycle firstly generates a random subset $S \subseteq P^{t-1}$ of P^{t-1}'s blocks within which some *random splitting* shall occur. Conversely, within the complement set $P^{t-1}\backslash S$ of remaining blocks some *random merging* shall occur. Aiming to have larger blocks more likely in S and smaller ones more likely in the complement $P^{t-1}\backslash S$, a random real $\alpha_i \in [0,1]$ is generated for each agent $i \in M$. Then, $S = \{A_1, \ldots, A_{|S|}\} =$

$$S = \big\{A \in P^{t-1} : |\{i \in A : \alpha_i \in [1-\alpha(t), \alpha(t)]\}| > 0\big\},$$

where the labeling $1, \ldots, |S|$ of coalitions $A \in S$ is random. In words, the larger $\alpha(t)$ (and, in any case, $\alpha(t) \geq 0.5$), the larger the interval $[1-\alpha(t), \alpha(t)]$. Also, for each block $A \in P^{t-1}$, the larger $|A|$, the more likely at least one agent $i \in A$ realizes some $\alpha_i \in [1-\alpha(t), \alpha(t)]$, in which case $A \in S$. Clearly, $P^{t-1}\backslash S = S^c = \{B_1, \ldots, B_{|S^c|}\} =$

$$= \big\{B \in P^{t-1} : |\{i \in B : \alpha_i \in [1-\alpha(t), \alpha(t)]\}| = 0\big\},$$

where, again, the labeling $1, \ldots, |S^c|$ of coalitions $B \in S^c$ is random. In this way, P^{t-1} (which is a collection of disjoint coalitions) gets partitioned into S and its complement S^c.

Within S random splitting occurs by means of the α_i's above: for each block $A \in S$, another set $A' \subseteq A$ is created as $A' = \{i \in A : \alpha_i < 0.5\}$. On the other hand, within S^c random merging occurs as follows. Firstly, again, S^c splits in two disjoint sets, $S_1, S^c_1 = S^c\backslash S_1$, by generating a random real $\alpha_k \in [0,1]$ for each $B_k \in S^c$ (i.e. $1 \leq k \leq |S^c|$) and setting $S_1 = \{B_k \in S^c : \alpha_k < 0.5\}$. Next, for each $B_k \in S_1$, a random integer $\beta_k \in \{0, 1, \ldots, |S_1^c|\}$ is generated. The newly generated partition P^t can now be defined as follows.

Random merging: if $\beta_k = 0$, then $B_k \in P^t$; otherwise, if $\beta_k > 0$, then $(B_k \cup B_{\beta_k}) \subseteq B' \in P^t$; in words, if $\beta_k = 0$, then B_k, which is a block of the inherited partition P^{t-1}, remains a block of the newly generated partition P^t as well. Conversely, if $\beta_k > 0$, then B_k merges with block B_{β_k} from S_1^c, which, in turn, also merges with all other blocks $B_h \in S_1$ such that $\beta_h = \beta_k$. That is to say, a generic block B' (resulting from random merging) of the newly generated partition P^t has form $B' = (B_k \cup B_{\beta_k}) \bigcup_{h:\beta_h = \beta_k > 0} B_h$.

Random splitting: if $\emptyset \subset A' \subset A$, then both A' and $A \backslash A'$ are blocks of the newly generated partition: $A', A \backslash A' \in P^t$. Otherwise, A being a block of the inherited partition P^{t-1}, it also remains a block of the newly generated one: $A \in P^t$.

If the α_i's, $i \in M$, are such that $S = \emptyset \Rightarrow S^c = P^{t-1}$, then the pure randomization cycle only performs random merging within P^{t-1}. Similarly, if $S^c = \emptyset \Rightarrow S = P^{t-1}$, then pure randomization occurs by only performing random splitting within P^{t-1}. In general, $\emptyset \subset S, S^c \subset P^{t-1}$, in which case the newly generated coalition structure P^t is *non-intersecting* the preceding one P^{t-1} [8]. That is, define agent subset $M' \subset M$ by $M' = \{i \in M : i \in A \in S\}$; then P^t is finer (coarser) than P^{t-1} over M' ($M \backslash M'$).

As already mentioned, the proposed method mainly alternates merging and splitting. In particular, consecutive pure randomization cycles are not allowed. This means that if cycle t is a pure randomization one, then either cycle $t-1$ is a merging one and cycle $t+1$ is a splitting one, or else the opposite, i.e. $t-1$ is a splitting cycle and $t+1$ is a merging one. Hence, if cycle $t-1$ was *not* a pure randomization one, then $\alpha(t) \geq \hat{\alpha}$ implies that cycle t is a pure randomization one. But if cycle $t-1$ was a pure randomization one, then independently from the value of $\alpha(t)$ cycle t shall be a merging or else a splitting one depending on whether cycle $t-2$ was, respectively, a splitting or else a merging one.

Both merging and splitting cycles also heavily rely upon randomization, but in a very biased fashion, where the bias obtains by means of the available information: the (dynamic) CF form and the worth of blocks of generated partitions (which completely disregarded by pure randomization cycles).

Merging. Merging cycles t start as above, that is, by computing the global worth $w(P^{t-1})$ of partition P^{t-1} inherited from the past and $\alpha(t) \in [\epsilon, 1-\epsilon]$ accordingly. Next, a random subset $S \subseteq P^{t-1}$ of P^{t-1}'s blocks is obtained by generating a random real $\alpha_k \in [0,1]$ for each $A_k^{t-1} \in P^{t-1}$ and then setting $S = \{A_k^{t-1} \in P^{t-1} : \alpha_k \leq \alpha(t)\}$. For notational convenience, $S = \{A_1, \ldots, A_{|S|}\}$, with random labeling $1, \ldots, |S|$ of coalitions $A \in S$. At this point, superadditivity of the underlying coalitional games or CFs $v_t, t \geq 0$, (assigning worths to coalitions) over the *field* 2^S is used in order to assess the more profitable merging within S. After computing $\frac{v_t(A_1)}{|A_1|}$, if both $v_t(A_1 \cup A_2) > v_t(A_1) + v_t(A_2)$ and $\frac{v_t(A_1 \cup A_2)}{|A_1|+|A_2|} > \frac{v_t(A_1)}{|A_1|}$ are true, then $B_1 = A_1 \cup A_2$; otherwise, $B_1 = A_1$. In any case, for A_3, if both $v_t(B_1 \cup A_3) > v_t(B_1) + v_t(A_3)$ and $\frac{v_t(B_1 \cup A_3)}{|B_1|+|A_3|} > \frac{v_t(B_1)}{|B_1|}$ are true, then $B_1 = B_1 \cup A_3$. In any case, it is next checked whether merging $B_1 \cup A_4$ is profitable or not in the same way, and this continues until reaching $A_{|S|}$, at which point some $\emptyset \neq B_1 \subseteq S$ (with, possibly, $B_1 = A_1$) will be given. This is reiterated within $S_1 = S \backslash \{A_k : A_k \subseteq B_1\}$, and then again within $S_2 = S_1 \backslash \{A_k : A_k \subseteq B_2\}$, and so on, until B_h is such that $S_{h-1} \backslash \{A_k : A_k \subseteq B_h\} = \emptyset$. The new partition P^t is now generated as follows: if $A_k^{t-1} \in P^{t-1} \backslash S$, then $A_k^{t-1} \in P^t$; otherwise, each $B_h \subseteq S$ resulting from the above reiterated procedure satisfies $B_h \in P^t$. Hence, P^t coincides with P^{t-1} over agent subset $M \backslash \Big(\bigcup_{1 \leq k \leq |S|} A_k \Big)$, while each block of the partition of $\bigcup_{1 \leq k \leq |S|} A_k$ is some B_h resulting from the above reiterated procedure.

Splitting. Splitting cycles t start as above, by computing the global worth $w(P^{t-1})$ of partition P^{t-1} inherited from the past and $\alpha(t) \in [\underline{\alpha}, 1-\underline{\alpha}]$ accordingly, and then generating a random subset $S \subseteq P^{t-1}$ of P^{t-1}'s blocks exactly like for pure randomization cycles, with larger blocks are more likely to enter $S = \{A_1, \ldots, A_{|S|}\}$, where the labeling $1, \ldots, |S|$ is random again. For each coalition $\{i_1, \ldots, i_{|A|}\} = A \in S$, with random ordering $1, \ldots, |A|$ of agents $i \in A$, set $B = A$ and for $1 \leq j \leq |A|$, if $v_t(B) - v_t(B \backslash i_j) \leq 0$, then update $B = B \backslash i_j$; otherwise, stop. Hence, for some random order of agents in A, it is iteratively checked if their *marginal contribution* yields $v_t\big(A \backslash \big(\bigcup_{1 \leq h < j} i_h \big)\big) \leq v_t\big(A \backslash \big(\bigcup_{1 \leq h \leq j} i_h \big)\big)$. When reaching the very first agent i_{j_*} whose marginal contribution is strictly positive the procedure stops. Next, the newly generated partition P^t coincides with P^{t-1} over the subset $M \backslash \Big(\bigcup_{1 \leq k \leq |S|} A_k \Big)$ of agents (as for the previous merging cycle), while each block of the partition of $\bigcup_{1 \leq k \leq |S|} A_k$ is either of the form $A \backslash \Big(\bigcup_{1 \leq h < j_*} i_h \Big)$, or else it is a 1-cardinal block $\{i_h\}$ with $i_h \in A$ and $h < j_*$, for some $A \in S$.

5. Simulation environment

Merge & split being designed for the dynamic CF form scenario, in order to test it through our simulation model COALA [2], some time-varying CFs $v_t : 2^M \rightarrow \mathbb{R}_+, t \geq 0$ is needed. It is chosen to be such that at each t the worth of optimal coalition structures is easily determined. This is achieved by means of a double-peaked (i.e. displaying two maxima, a global and a local one) and bi-symmetric (i.e. with the population partitioned into two types and the worth of coalitions depending only on members' type) CF. Recall once again that such a maximum worth is not an information available for coalition structure generation, but it is crucial for evaluating the performance.

Double-peaked and bi-symmetric CFs. Consider the set $D_m = \{m^* : \frac{m}{m^*} \in \mathbb{N}\} \neq \emptyset$ of divisors of the number m of agents. For the simulations, two (random) sequences $m_t^*, m_t^{**} \in D_m$ are generated, with $m_t^* \leq m_t^{**}$ for every t. For $A \in 2^N$ and $m_t^*, m_t^{**} \in D_m$, let $A_t^* := \{i \in A : i < \frac{m}{m_t^*}\}$ and $A_t^{**} := \{i \in A : i < \frac{m}{m_t^{**}}\}$, where $m_t^* \leq m_t^{**}$ clearly implies $A_t^{**} \subseteq A_t^*$. The worth $v_t(A), t \geq 0$ is

$$v_t(A) = \frac{|A|^2}{1 + \min\{\lambda_t^*(A), \lambda_t^{**}(A)\}} \text{ for all } A \in 2^M, \qquad (3)$$

where $\lambda_t^*(A) := (|A| - m_t^*)^2 + (|A_t^*| - 1)^2$ and, analogously, $\lambda_t^{**}(A) := (|A| - m_t^{**})^2 + (|A_t^{**}| - 1)^2$.

The setting is dynamic, of course, as m_t^*, m_t^{**} change in time. In particular, at any t optimal coalition structures are

$\frac{m}{m_t^{**}}$-cardinal, each of their blocks is m_t^{**}-cardinal and contains precisely one agent with identifier $i < \frac{m}{m_t^{**}}$, so that the associated global worth is $\frac{m}{m_t^{**}} \cdot (m_t^{**})^2 = m \cdot m_t^{**}$. This CF form is useful to test the generation process in the tricky dynamic scenario where changes of the CF, occurring at generic times t', obtain by setting $m_{t'}^{**} = m_{t'+1}^{*}$. In words, whenever they occur, changes do not lead to a fall in the the worth of previously optimal coalition structures. Yet, after changes occur such previously optimal coalition structures correspond to local rather than global maxima. Thus, CF changes (and novel optimal coalition structures) are harder to detect (and find).

At any step t, the number of optimal coalition structures is

$$\frac{\prod_{0\leq k\leq \frac{m}{m_t^{**}}-1} \binom{m-\frac{m}{m_t^{**}}-k(m_t^{**}-1)}{m_t^{**}-1}\left(\frac{m}{m_t^{**}}-k\right)}{\left(\frac{m}{m_t^{**}}\right)!}, \qquad (4)$$

which might seem large, making it simpler to generate near-optimal coalition structures. Still, from a game-theoretical perspective the case with many (Pareto) optimal outcomes is not trivial at all, modeling the situation where achieving optimality is not hard because of strategic matters but for the likelihood of coordination failure [15].

Performance evaluation. The designed and implemented simulation model is based on the GAIA framework and the ARTÌS middleware [1]. The approach relies on the parallel and distributed simulation technique, specifically the ARTÌS middleware provides the simulation engine and the GAIA framework, that is based on dynamic reallocation of simulated model entities, enhances the simulation efficiency, increasing the simulator's scalability and performance [6]. These high performance tools allow for a wide variety of simulations, with large number of agents and very complex simulated models. In any case, dynamic coalition structure generation constitutes a novel complex setting where to work with such tools.

The following two figures present, respectively, the worth of generated coalition structures and the average cardinality of blocks. In particular, the worth of each generated coalition structure is divided by the worth of optimal coalition structures, that is known being $m \cdot m_t^{**}$ at each step t. This ratio is named *normalized Global Worth* (GW), which seems most naturally plotted together with the exploration parameter α, so to check whether changes of the CF are actually detected and how the algorithm behaves accordingly. Moreover, in the same figure the historical mean GW is also plotted, which is useful to depict the trend of the simulated scenario. In the companion figure two further ratios appear: the current m^* and m^{**} both divided by the maximum m^{**} generated during the simulation itself. This enables to check the speed at which coalitions tend toward the optimal cardinality and, when such an optimal cardinality is not reached, on what average cardinality coalitions tend to stabilize. In this simulation environment a population of $m = 10^4$ agents is considered; more populated environments will be considered in the future. Figures 1 and 2 show a simulation of the dynamic CF form scenario defined by expression (3) above. The two randomly generated sequences of divisors of 10^4 (with $m_t^* \leq m_t^{**}$ as desired, see above) are $m_t^* = 4, 10, 25, 10, 20, 40, 100, 20, 10, 25, 50, 20, 100, 10, 25, 50, 4, 10, 25, 50$ and $m_t^{**} = 10, 20, 50, 25, 40, 100, 500, 25, 100, 50, 100, 100, 500, 20, 100, 200, 100, 25, 50, 100$, with each value applying to a 500 step interval. In the former figure the continuous black line shows the behavior displayed by $\alpha(t) \in [\underline{\alpha}, \overline{\alpha}] \subset [0, 1]$. In order to obtain a $[0, 1]$-ranged plot for the worth of generated coalition structures $P^t \in \mathcal{P}^M, t \geq 0$ as well, the dotted light gray line shows

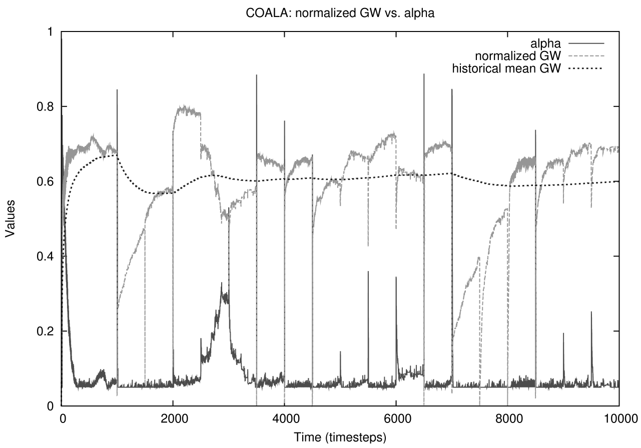

Fig. 1. *Normalized GW and α with 10^4 agents and m^*, m^{**} randomly varying every 500 time steps.*

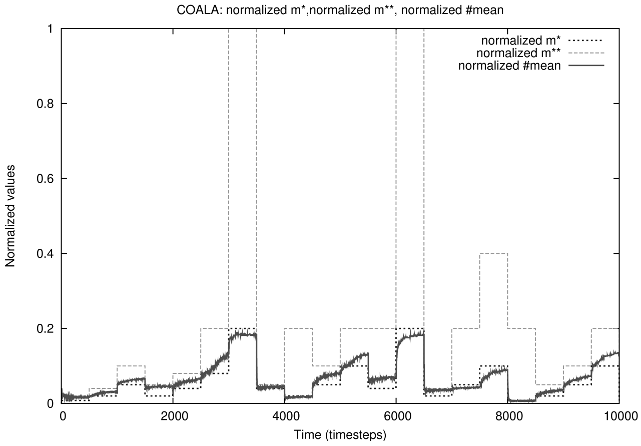

Fig. 2. *Normalized average cardinality of coalitions and normalized m^*, m^{**} with 10^4 agents and m^*, m^{**} randomly varying every 500 time steps.*

$$\text{normalized GW}(t) := \sum_{A \in P^t} \frac{v_t(A)}{m \cdot m_t^{**}} = \frac{w_t(P^t)}{m \cdot m_t^{**}}, \qquad (5)$$

with $v_t(A), A \in 2^M$ defined by expression (3) above.

It may be immediately noticed that the historical mean GW (that is, the average performance over time) is steadily around 60% of the worth of optimal coalition structures (see Figure 1, dotted dark gray line). This seems satisfactory given that the system is rather dynamic and in several simulation intervals the average cardinality of coalitions is closer to m^* than to m^{**} (see Figure 2, continuous black line). Inspection of step interval 2000-3000 reveals that the exploration parameter α behaves properly: as long as the GW (see Figure 1, dotted light gray line) falls over time, α keeps increasing until an optimum is reached, although this latter is only a local one (see Figure 2, 3000-3500 time-steps).

An overall inspection of normalized GW vs $\alpha(t)$ leads to conclude that the noise around optimal coalition structures is higher when the cardinality of optimal coalitions (i.e. blocks) is smaller. In fact, there is a trade-off: larger optimal blocks take longer to be reached but the generation process is also more stable, while when optimal coalition structures consist of many small blocks (i.e. when m_t^{**} is smaller) generated partitions get closer to optimal ones faster, but their volatility is higher. In general the overall performance seems satisfactory given the rather challenging scenario under concern.

6. Concluding remarks and future developments

In dynamic coalition structure generation, at any time the solver can choose what partition to generate at that time, but only knows the worth of all previously generated partitions, and aims at maximizing the generated stream of worth over a finite or (countably) infinite time horizon. Such a minimal available information identifies the dynamic non-CF form scenario, while in the dynamic CF case the worth of partitions is known to be given by the sum of their blocks' worth according to some (unknown) time-varying underling CF or coalitional game. This latter is the environment where the merge & split adaptive coalition structure generation process has been designed and tested through simulations.

In the chosen setting, the available information is intrinsically incomplete and, in particular, never sufficient for observing the actual CF v_t assigning worths to coalitions or blocks at time t. In fact, this latter is a 2^m-dimensional vector, but one always observes only the $|P^t|$ worths of P^t's blocks, while the remaining $2^m - |P^t|$ coordinates of v_t remain unknown. Hence, no attempt is made to learn the time-behavior of the underlying CF.

Given any partition P^t at any t, merge & split basically devotes cycle $t+1$ to searching for some best partition P^{t+1} in a neighborhood of P^t. This search is always stochastic, in that the collection of partitions in P^t's neighborhood that are actually explored is determined through (biased) randomization, alternating two main exploration directions: coarsening or else refining the partition generated by the preceding cycle. Merging cycles only explore some partitions (weakly) coarser than the previous one, and the output is a new partition worth no less than the previous one. Splitting cycles only explore some partitions (weakly) finer than the previous one, and the output is a new partition worth no less than the previous one. This alternation is interrupted by sufficiently large falls in generated worth, as these latter lead to pure randomization cycles, within which no search whatsoever is performed: the output is some new partition (in general neither coarser nor finer that the preceding one) generated independently from the attained worth.

The general idea behind merge & split seems adaptable to the dynamic non-CF form scenario, where one only observes the worth of generated partitions, and not the worth of their blocks as well, although this would definitely require some relevant changes. Yet, borrowing from cooperative game theory, an intermediate possibility (i.e. still more similar to the CF form) is represented by games in PFF (partition function form). They are (poset) functions taking values on *embedded* pairs, namely, pairs consisting of a partition and a coalition such that the former embeds the latter as one of its blocks [14]. Concretely, this sophisticates the dynamic CF-form scenario considered thus far by assuming that the solver may observe the worth of generated blocks although such a worth may well vary depending on what generated partition embeds the block (static coalition structure generation in PFF is addressed in [18]).

References

[1] ARTÌS: Advanced RTI System Homepage, http://pads.cs.unibo.it/artis, 2008.

[2] COALA: COAlitions Learn and Adapt, http://pads.cs.unibo.it/coala, 2008.

[3] M Aigner, Combinatorial Theory, Springer-Verlag, 1979.

[4] B Blankenburg, R K Dash, S D Ramchurn, M Klusch and N R Jennings, Trusted kernel-based coalition formation. AAMAS2005 Proceedings 2005, ACM Press, pp. 989–996.

[5] C Blum and C Roli, Metaheuristics in combinatorial optimization: overview and conceptual comparison. ACM Computing Surveys, Vol. 35, No. 3, 2003.

[6] L Bononi, M Bracuto, G D'Angelo and L Donatiello, An adaptive load balancing middleware for distributed simulation. ISPA 2006 Workshops 2006, Springer.

[7] V D Dang and N R Jennings, Coalition structure generation in task-based settings. ECAI2006 Proceedings 2006, pp. 567–571.

[8] I Gilboa and E Lehrer, Global games. International Journal of Game Theory, Vol. 20, 1991, pp. 129–147.

[9] R Graham, D Knuth and O Patashnik, Concrete Mathematics - A Foundation for Computer Science, Addison-Wesley, 1994.

[10] M Klusch and A Gerber, Dynamic coalition formation among rational agents. IEEE Intelligent Systems, Vol. 17(3), 2002, pp. 42–47.

[11] K Larson and T Sandholm, Anytime coalition structure generation: an average case study. Journal of Experimental and Theoretical Artificial Intelligence, Vol. 12, No. 1, 2000, pp. 23–42.

[12] K Lerman and O Shehory, Coalition formation for large-scale electronic markets. ICMAS2000 Proceedings 2000, pp. 167–174.

[13] C Li and K Sycara, Algorithm for combinatorial coalition formation and payoff division in an electronic marketplace. AAMAS2002 Proceedings 2002, pp. 120–127.

[14] R Myerson, Values of games in partition function form. International Journal of Game Theory, Vol. 6, 1977, pp. 23–31.

[15] T Sandholm, Distributed rational decision making. Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence 1999, G Weiss, Ed., MIT Press, pp. 201–258.

[16] T Sandholm, K Larson, M Andersson, O Shehory and F Tohmè, Coalition structure generation with worst case guarrantees. Artificial Intelligence, Vol. 111, 1999, pp. 209–238.

[17] T Sandholm and S Suri, BOB: Improved winner determination in combinatorial auctions and generalizations. Artificial Intelligence, Vol. 145, 2003, pp. 33–58.

[18] S Sen and P Dutta, Searching for optimal coalition structures. Proceedings of the The Fourth International Conference on Multi-Agent Systems (ICMAS) 2000, pp. 287–292.

[19] O Shehory and S Kraus, Methods for task allocation via agent coalition formation. Artificial Intelligence, Vol. 101, 1998, pp. 165–200.

[20] O Shehory and S Kraus, Feasible formation of coalitions among autonomous agents in non-superadditive environments. Computational Intelligence, Vol. 15, 1999, pp. 218–251.

[21] K Sycara, K Decker, A Pannu, M Williamson and D Zeng, Distributed intelligent systems. IEEE Expert-Intelligent Systems and their Applications, Vol. 11, No. 6, 1996, pp. 36–45.

[22] J Yamamoto and K Sycara, A stable and efficient buyer coalition formation scheme for e-marketplaces. ICAA2001 Proceedings 2001, pp. 576–583.