A theory to devise dependable cooperative encounters

Humbert Fiorino, Damien Pellier

A theory to devise dependable cooperative encounters

Humbert Fiorino and Damien Pellier

Univ. Grenoble Alpes, LIG, F-38000 Grenoble, France
humbert.fiorino@imag.fr, damien.pellier@imag.fr

Abstract. In this paper, we investigate the question of how to characterize "fault tolerance" in cooperative agents. It is generally admitted that cooperating agents can achieve tasks that they could not achieve without cooperation. Nevertheless, cooperating agents can have "Achilles' heels", a *cooperative encounter* can eventually fail to achieve its tasks because of the collapse of a single agent. The contribution of this paper is the study of how cooperating agents are affected by *dependability* issues. Specifically, our objectives are twofold: to formally define the concepts of dependability in cooperative encounters, and to analyze the computational complexity of devising dependable cooperative encounters.

Keywords: Collaboration & Coordination, Teamwork, Cooperation Theory.

1 Introduction

In this paper, we investigate the question of how to characterize "fault tolerance" in cooperative agents. It is generally admitted that cooperating agents can achieve tasks that they could not achieve without cooperation. For instance, a group of agents is committed in achieving a common task that none of them is able to fulfill. As a consequence, they decompose the initial common task into subtasks which are easier to handle separately. With this divide-and-conquer strategy, they can examine different alternatives, that is, redundant ways of achieving their subtasks [7, 10, 11, 17]. In this framework, the risk of failures is disseminated over the multiagent system because agents commit to tasks corresponding to their skills.

Nevertheless, cooperative encounters have "Achilles' heels". Some agents are much more involved in the encounter's outcome than others, and thus, deserve a closer consideration. A whole group can eventually breakdown because of the collapse of a single agent. Much work has been done in investigating cooperation representations, dependency relations between agents' activities [7, 10, 11, 17], conflict resolution [4] and task allocation [15]. Acting coherently despite partial or erroneous knowledge, partner failures and unpredictable events is central to multi-agent research works. But, they provide few arguments to identify and anticipate strengths and weaknesses in a group of cooperating agents. The main contribution of this paper is the study of how cooperating agents are affected by dependability issues. Specifically, our objectives are twofold:

- to formally define the concept of *dependability in cooperative encounters*,
- to characterize the computational complexity of achieving dependable encounters.

We introduce the fundamental concepts for our framework and define the *Cooperative Encounter Problem* as a decision problem combining task decomposition and allocation (section 2). Then, we prove that this problem is intractable in principle (section 3). Section 4 presents the related works, and some conclusions and future works are proposed in section 5.

2 COOPERATIVE ENCOUNTER FORMALIZATION

In this section, we define the fundamental concepts on which our search procedures are based.

2.1 Definitions

Cooperative encounters involve a set of agents, $\mathcal{A} = \{A_0, \ldots, A_n\}$ and a set of tasks, $\mathcal{T} = \{T_0, \ldots, T_m\}$. A task is either *primitive* or *composite*. This is represented as a set of *decomposition rules* $\mathcal{R}$ such that $(op, T_i, \rho) \in \mathcal{R}$: $\rho = [\tau_1, \ldots, \tau_k]$ is the decomposition of the composite task T_i into a list of k subtasks $\tau_i \in \mathcal{T}$, and $op \in \{AND, OR\}$: the AND operator means that a task is achieved if and only if all its subtasks are achieved; the OR operator signifies that at least one subtask has to be achieved in order to realize the composite task. We require that composite tasks appear only once in decomposition rules and we strictly forbid recursion. The primitive tasks do not have decomposition rules.

Each agent A_i has a set of $S_i \subseteq \mathcal{T}$ of **primitive** tasks that it can achieve, i.e. its *skills*: $\mathcal{S} = \{S_1, \ldots, S_n\}$ represents what each agent can do. Furthermore, we consider that agents can be *mutually exclusive* (mutex) in order to take into account conflicts, incompatible interests or unwillingness to work together, etc. $\nabla_i \subseteq \mathcal{A} - \{A_i\}$ denotes the set of A_i's mutually exclusive agents: $A_j \in \nabla_i$ if and only if $A_i \in \nabla_j$. Let $\mathcal{O} = \{\nabla_1, \ldots, \nabla_n\}$ be the set of mutually exclusive agents. (Ω_i, T_i) is the *assignment* of a set of agents $\Omega_i \subseteq \mathcal{A}$ to a task T_i. A task T_i is *achievable* if and only if there is no mutually exclusive agent in Ω_i and T_i is a skill of all the agents of Ω_i (otherwise it is *unachievable*). We use the term "achievable" rather than "achieved" on purpose: it means that T_i can be achieved if, at least, one of the agents of Ω_i does not collapse. Or, equivalently, T_i is not achieved if all the agents collapse. Our formalization does not constrain the meaning of the agent's collapse in any sense: it can be a rational decision to abandon, a failure, a malicious attack etc.

Now, we give a formal definition of the problem we want to address:

Definition 1. *A Cooperative Encounter Framework is a tuple* $CEF = (\mathcal{A}, \mathcal{T}, \mathcal{S}, \mathcal{R}, \mathcal{O})$. *A Cooperative Encounter Problem is a tuple* $\wp = (CEF, \lambda)$ *with* $\lambda = [T_0]$. T_0 *is the initial task. A cooperative encounter* $\Delta = [(\Omega_0, T_0), \ldots, (\Omega_k, T_k)]$ *is a list of assignments.*

In the following defintions, $e \cdot S$ stands for "in the list $e \cdot S$, e is the head and S is the tail", $R + S$ is the concatenation of R and S lists, $|A|$ is the cardinality of the set A and

$A \bigotimes B = \{a \cup b \mid a \in A, b \in B\}$ is the cartesian product: $\{\{x\}, \{x'\}\} \bigotimes \{\{x''\}\} = \{\{x, x''\}, \{x', x''\}\}$. Then, we define how cooperative encounters are solution for Cooperative Encounter Problems as follows:

Definition 2. *A cooperative encounter Δ is a solution for a Cooperative Encounter Problem $\wp = (CEF, \lambda)$ if and only if either:*

1. *$\Delta = []$ and $\lambda = []$, or*
2. *Given $\Delta = (\Omega, T).\Delta'$ and $\lambda = T.\lambda'$, at least one of the following conditions is satisfied:*
   (a) *T is primitive:*
   T is achievable by Ω and Δ' is a solution of (CEF, λ');
   (b) *$\exists(AND, T, \rho) \in \mathcal{R}$:*
   Δ' is a solution of $(CEF, \rho + \lambda')$. That is, T is a composite task and all its subtasks have solutions;
   (c) *$\exists(OR, T, \rho) \in \mathcal{R}$:*
   $\exists t \in \rho$ such that Δ' is a solution of $(CEF, t.\lambda')$. That is, T is a composite task and at least one of its subtasks has a solution.

Definition 2 is recursive: it defines the achievement of the initial task T_0 as a decomposition process of T_0 into achievable primitive tasks. From here, the expression "cooperative encounter" will stand for "a cooperative encounter that is solution of a Cooperative Encounter Problem".

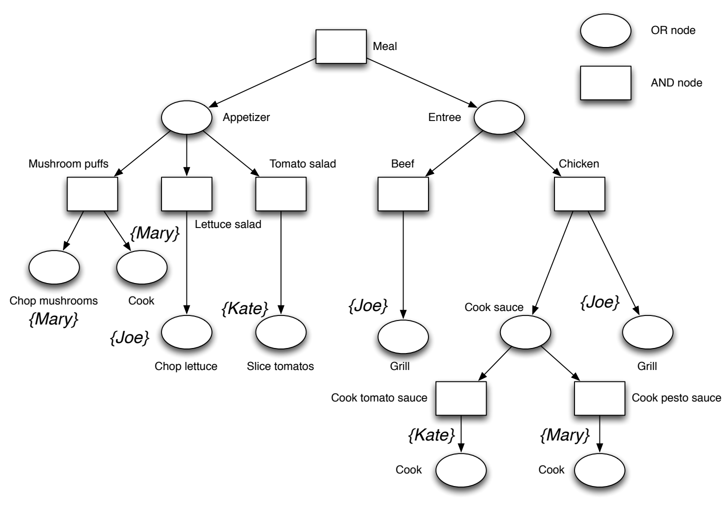

Fig. 1. A Cooperative Encounter Tree for the meal preparation.

Figure 1 represents a cooperative encounter for the preparation of a meal consisting of an appetizer and an entree [10]. In this specification, cooking chicken means cooking

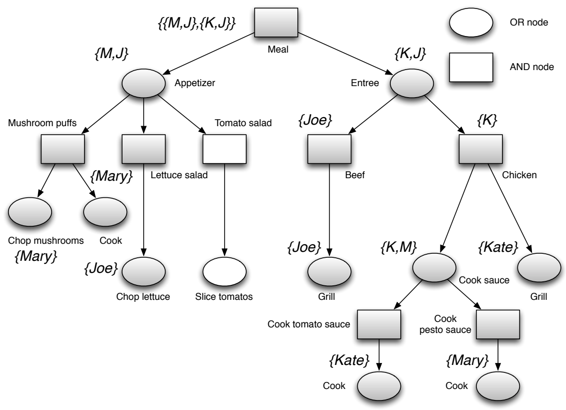

Fig. 2. Maximizing the conspiracy. The explored nodes appear in grey. {...} represents the conspiracies.

a sauce and grilling the chicken. There are two alternatives for the sauce: either a tomato sauce or a pesto sauce. The tomato and pesto sauces are respectively performed by Kate and Mary; Joe is in charge of grilling the beef or the chicken. With respect to the *dependability* of their encounter, this role distribution is not the most appropriate because if Joe eventually does not attend the dinner (whatever the reason), the entree will not be done and the meal preparation will fail. A more adequate role distribution regarding the encounter's dependability, assuming that Joe and Kate have equivalent skills, is to assign the beef grilling task to Kate. As a consequence, whatever the failure of one of the invitees, the meal will be done. In this new role distribution, at least two agents (Joe and Kate) must fail to cause the meal to collapse. Hence, the resulting encounter is more dependable because the simultaneous failure of two agents is more improbable than one isolated failure (assuming that the probabilities of failure are independent).
The key idea of this paper is that role distributions define critical set of agents that are responsible for the vulnerability of cooperative encounters to failures. We name *conspiracy* the set of agents that must *simultaneously* fail to prevent the encounter's success. Thus, *the bigger the conspiracy, the more dependable the cooperative encounter*. Therefore, we define a dependable cooperative encounter is an optimization problem consisting in calculating a role distribution maximizing the conspiracy's size. In the meal scenario, the participating agents are Kate, Mary and Joe. However, the largest possible conspiracy is composed of Joe and Kate.

Then, how do we determine the conspiracies in a cooperative encounter? At this point, we know that the initial task is achievable by a decomposition into subtasks and agents assigned to these subtasks. But, not all the agents have the possibility to form a conspiracy because of the existence of various alternatives (OR nodes). Intuitively, some agents are more important than others with respect to the dependability of the cooperative encounter. To formalize this intuition, we need two "helper functions", $\top$ ("top") and $\perp$ ("bottom"), which will be used later in the computation of the cooperative encounters and the conspiracies.

Given CEF $= (\mathcal{A}, \mathcal{T}, \mathcal{S}, \mathcal{R}, \mathcal{O})$, $\wp = (\text{CEF}, [T_0])$ and Δ, we define $\perp$ and $\top$ as follows:

Definition 3.

$$\top(T) = \begin{cases} \bigotimes_{t \in \rho} \top(t) & \textit{if } (\textit{AND}, T, \rho) \in \mathcal{R}, \\ \bigcup_{t \in \rho} \top(t) & \textit{if } (\textit{OR}, T, \rho) \in \mathcal{R}, \\ \{\emptyset\} & \textit{if } T \textit{ is primitive and achievable by } \Omega, \\ \{T\} & \textit{if } T \textit{ is primitive and unachievable.} \end{cases}$$

In Definition 3, $\top(T)$ represents the set of tasks that have to be achieved in order to achieve T, and $\emptyset$ means that there is nothing to do to achieve T as shown by the following theorem:

Theorem 1. *Let $\wp = (CEF, [T_0])$ be a Cooperative Encounter Problem. $\Delta = [(\Omega_0, T_0), \ldots, (\Omega_k, T_k)]$ is a solution of $\wp$ if and only if $\emptyset \in \top(T_0)$.*

Proof idea: the proof is by induction on the task decomposition depth k. For a single node tree ($k = 0$), it is easy to see from the definition 3 that the theorem is true: proof sets of achievable leaf nodes contain the empty set element. Hence, the theorem is admitted for trees whose depth is inferior or equal to k. The theorem is proved for $k + 1$ depth trees by showing that achievable roots have k depth subtrees – all subtrees being achievable if that root is an AND node and at least one otherwise. Because these subtrees verify the theorem by induction hypothesis, it is not difficult to conclude from the definition 3 that the proof sets of achievable $k + 1$ depth trees also contain $\emptyset$.

Definition 4.

$$\perp(T) = \begin{cases} \bigcup_{t \in \rho} \perp(t) & \textit{if } (\textit{AND}, T, \rho) \in \mathcal{R}, \\ \bigotimes_{t \in \rho} \perp(t) & \textit{if } (\textit{OR}, T, \rho) \in \mathcal{R}, \\ \{\Omega\} & \textit{if } T \textit{ is primitive and achievable by } \Omega, \\ \{\emptyset\} & \textit{if } T \textit{ is primitive and unachievable.} \end{cases}$$

Definition 4 recursively computes the set of agents committed to the achievement of T. The smallest elements of $\perp(T)$ are "the most critical sets of agents", i.e. the agents that have the possibility to form conspiracies and collapse T (if $\emptyset \in \perp(T)$ then there is no set of agents able to achieve T). Then we define the **conspiracy set** $\chi(t)$ as follows:

Definition 5. $\chi(T) = \{x \in \perp(T) | \forall x' \in \perp(T), |x| < |x'|\}$

In Figure 1, the meal is achievable because all its subtasks ("Appetizer" and "Entree") are achievable. Consider for instance the appetizer preparation: this is a composite taks, which is achievable unless {Mary, Joe, Kate} (the only conspiracy in the conspiracy set of the task "Appetizer") do not realize their tasks.

3 THE COOPERATIVE ENCOUNTER PROBLEM COMPLEXITY

Now, consider the question of the Cooperative Encounter Problem satisfiability: given a CEP, does it admit a cooperative encounter Δ?

Let CE-SAT= $\{(CEF, [T_0])|\ T_0$ is achievable$\}$. Not surprisingly,

Theorem 2. CE-SAT *is NP-complete.*

Proof: To show that CE-SAT is NP-complete, we must show that it is in NP and that all NP-problems are polynomial time reductible to it [16]. The first part consists in showing that, given a cooperative encounter Δ, there is a polynomial time algorithm that verifies that it is a solution of $(CEF, [T_0])$. The last part of the proof is based on a polynomial time reduction from 3SAT to CE-SAT.

CE-SAT is in NP. The proof by induction is based on the length k of the cooperative encounter Δ: $|\Delta| = k$.

- Basis: Proving that Δ is a solution for $k = 0$ is immediate. Here is a procedure that runs in polynomial time:
  1. If $k = 0$, test whether $\lambda = []$.
  2. If the test passes, *accept*; otherwise, *reject*.
- Induction step: For each $k \geq 0$, assume that CE-SAT satisfaction is in P for k (induction hypothesis) and show that it is also true for $k+1$. If $|\Delta| = k+1$, Δ is a solution iff case (2) in definition 2 is true. We specifically analyze condition (2–a), the other conditions are similar. We give the following procedure:
  1. Test whether $(\Omega, T) = head(\Delta)$ and $T = head(\lambda)$.
  2. Test whether $tail(\Delta)$ is a solution of $(CEF, tail(\lambda))$.
  3. If both pass, *accept*; otherwise, *reject*.

The first test is decidable in polynomial time, so is the second because $|tail(\Delta)| = k$.

Here are the details of the reduction from 3SAT to CE-SAT that operates in polynomial time. Let $\phi = C_1 \wedge C_2 \wedge \cdots \wedge C_n$ where C_1 is a clause of formal parameters (for instance $a_1 \vee b_1 \vee c_1$) and each parameter corresponds to a propositional variable $a_1 = a$, $b_1 = \neg b$ etc. The reduction maps a Boolean formula ϕ to a CEP $\wp = (CEF, [\phi])$. The set of agents $\mathcal{A}$ in the $CEF = (\mathcal{A}, \mathcal{T}, \mathcal{S}, \mathcal{R}, \mathcal{O})$ contains all the propositional variables. The set of tasks $\mathcal{T}$ is $\{a_1, b_1, c_1, \ldots, a_n, b_n, c_n\}$. Each agent's skill in $\mathcal{S}$ is defined by the mapping between the formal parameters and the propositional variables. The decomposition rules $\mathcal{R}$ are as follows: $(AND, \phi, [C_1, \ldots, C_n])$, $(OR, C_1, [a_1, b_1, c_1])$, $\ldots$, $(OR, C_n, [a_n, b_n, c_n])$. The mutex of an agent a in $\mathcal{O}$ is the negation of the corresponding propositional variable. For instance, $\nabla_a = \{\neg a\}$ etc.

We show that ϕ is satisfiable iff $\wp$ has a solution. If ϕ is satisfiable, there exists at least a true variable in each clause and an assignment of non contradictory variables. As a consequence, the corresponding agents are not mutexes. Let each of these agents commit to the leaves of the OR nodes and $\Delta = [(\Omega_0, \phi), (\Omega_1, C_1), \ldots, (\Omega_n, C_n), (\{a\}, a_1), (\{\neg b\}, b_1), \ldots]$. This is a solution of $\wp = (CEF, [\phi])$ because at least one agent commits to one of the leaves of all OR-type rules. Conversely, if $\wp = (CEF, [\phi])$ admits a solution Δ, by construction, at least one agent commits to a leaf in each OR node. We then assign *true* to each corresponding propositional variable. This assignment is consistant because agents are not mutexes in cooperative encounters (the corresponding variables are not contradictory) and at least one literal is true in each clause. Hence, ϕ is satisfiable.

4 RELATED WORK

The Cooperative Encounter Tree is a cooperative structure very similar to those used elsewhere. The main difference is that CET are the result of a decision process when pre-defined cooperative structures are used in the literature as support for group activity. In their work on collaborative plans for complex group action [7], B. J. Grosz and S. Kraus rely on *recipes* to represent actions at different levels of abstraction, agents commit to them etc. In this framework, agents must decide what recipes to use and, if an agent is unable to perform an assigned action then the group revises its recipe. Recipes have been then extended to *Probabilistic Recipe Trees* [10] where each branch of an OR node to one of its children has an associated probability representing the likelihood of being selected. As a consequence, the agents implement decision-making strategies about the relevance of communicating information and perform actions helpful to their partners.
STEAM framework [17] focus is on devising general models of *teamwork* for the agents. Such models give them the ability to have appropriate behaviors whenever they discover unexpected opportunities or unexpectedly fail in fulfilling responsabilities. Group activity is represented by a hierarchy of team or individual *operators* that have rules of application and terminaison. Quite similarily to our approach, a *role* is an abstract specification of the set of activities an individual or a subteam undertakes in service of the team's overall activity; operators are connected to their sub-operators by AND-combination, OR-combination and role dependency relations.
Generalized Partial Global Planning (GPGP) [11] is also associated with a Hierarchical Task Network representation. This representation called TÆMS is an AND/OR goal tree with relations to data and resources that are needed to solve specific subgoals. Furthermore, interpendencies relations among goals are allowed in order to indicate that one goal may facilitate the achievement of another goal or may hinder it. TÆMS representation allows the agents to reason on how their local decisions influence other agents' activites and help them to schedule tasks in the most appropriate way.
More generally, mathematical treatment of cooperation are based on either game-theoric or modal logic formulations. M. d'Inverno and al. [6] have defined a graph structure of goals and discussed its properties for representing cooperation. Then, they have shown that the problem of determining whether cooperation structures are avalaible to achieve an agent's goal is NP-complete.

Contingent planning is the task of generating a conditional plan given uncertainty about the initial state and action effects, but with the ability to observe some aspects of the current world state. Contingent planning can be transformed into an And-Or search problem in belief space, the space whose elements are sets of possible worlds [8, 1, 13]. In online contingent planning under partial observability, an agent decides at each time step on the next action to execute, given its initial knowledge of the world, the actions executed so far, and the observation made. Such agents require some representation of their belief state to determine which actions are valid, or whether the goal has been achieved. Efficient maintenance of a belief state is, given its potential exponential size, a key research challenge [2]. In [5], the authors consider a general concept of undoability, asking whether a given action can always be undone, no matter which state it is applied to. This generalizes previous concepts of invertibility, and is relevant for search as well as applications.

Another related research area is multi-agent planning [3, 12]. Multi-agent planning deals with the problem of classical planning for multiple cooperative agents who have private information about their local state and capabilities they do not want to reveal [14]. Two main approaches have recently been proposed to solve this type of problem: one is based on reduction to distributed constraint satisfaction, and the other on partial-order planning techniques. In classical single-agent planning, constraint-based and partial-order planning techniques are currently dominated by heuristic forward search. The question arises whether it is possible to formulate a distributed heuristic forward search algorithm for privacy-preserving classical multi-agent planning. In [9], multiagent planning for cooperative agents in deterministic environments intertwines synthesis and coordination of the local plans of involved agents. Both of these processes require an underlying structure to describe synchronization of the plans. A distributed planning graph can act as such a structure, and the authors propose a general negotiation scheme for multiagent planning based on planning graphs.

5 CONCLUSION & PERSPECTIVES

Cooperation is a central issue in multi-agent systems and the research effort has focused mainly on trying to understand with models and experiments which are their desirable features. In this paper, we have emphasized some possible shortcomings of cooperation. We have formally introduced the concepts of dependability and conspiracy in cooperative encounters. We have shown that achieving dependable encounters is a hard problem.

We are investigating the search algorithms and the heuristics to find and maximize dependable encounters. The idea is to build solutions with the highest vulnerability at first and then to reduce it by making conspiracies as large as possible (anytime approach).

References

1. Botea, A., Braghin, S.: Contingent versus deterministic plans in multi-modal journey planning. In: Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling, ICAPS 2015, Jerusalem, Israel, June 7-11, 2015. pp. 268–272 (2015)
2. Brafman, R.I., Shani, G.: Online belief tracking using regression for contingent planning. Artif. Intell. 241, 131–152 (2016)
3. Cardoso, R.C., Bordini, R.H.: A distributed online multi-agent planning system. In: Proceedings of the Workshop on Distributed and Mutiltiagent Planning (ICAPS). pp. 15–23 (2016)
4. Cox, J.S., Durfee, E.H.: An efficient algorithm for multiagent plan coordination. In: AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems. pp. 828–835. ACM, New York, NY, USA (2005)
5. Daum, J., Torralba, Á., Hoffmann, J., Haslum, P., Weber, I.: Practical undoability checking via contingent planning. In: Proceedings of the Twenty-Sixth International Conference on Automated Planning and Scheduling, ICAPS 2016, London, UK, June 12-17, 2016. pp. 106–114 (2016)
6. d'Inverno, M., Luck, M., Wooldridge, M.: Cooperation structure. In: Proceedings of the Fifteenth Intenational Joint Conference on Artificial Intelligence. pp. 600 – 605. Nagoya, Japan (1997)
7. Grosz, B., Grosz, B.J., Kraus, S., Kraus, S.: Collaborative plans for complex group action. Artificial Intelligence 86, 269–357 (1996)
8. Hoffmann, J., Brafman, R.I.: Contingent planning via heuristic forward search with implicit belief states. In: Proceedings of the Fifteenth International Conference on Automated Planning and Scheduling (ICAPS 2005), June 5-10 2005, Monterey, California, USA. pp. 71–80 (2005)
9. J. Tozicka, J. Jakubuv, K.D., Komenda, A.: Multiagent planning by iterative negotiation over distributed planning graphs. In: Proceedings of the Workshop on Distributed and Mutiltiagent Planning (ICAPS). pp. 7–15 (2014)
10. Kamar, E., Gal, Y., Grosz, B.J.: Incorporating helpful behavior into collaborative planning. In: AAMAS '09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems. pp. 875–882. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2009)
11. Lesser, V., Decker, K., Wagner, T., Carver, N., Garvey, A., Horling, B., Neiman, D., Podorozhny, R., Prasad, M.N., Raja, A., Vincent, R., Xuan, P., Zhang, X.Q.: Evolution of the GPGP/TÆMS domain-independent coordination framework. Autonomous Agents and Multi-Agent Systems 9(1-2), 87–143 (2004)
12. Luis, N., Borrajo, D.: Plan merging by reuse for multi-agent planning. In: Proceedings of the Workshop on Distributed and Mutiltiagent Planning (ICAPS). pp. 38–46 (2014)
13. Maliah, S., Brafman, R.I., Karpas, E., Shani, G.: Partially observable online contingent planning using landmark heuristics. In: Proceedings of the Twenty-Fourth International Conference on Automated Planning and Scheduling, ICAPS 2014, Portsmouth, New Hampshire, USA, June 21-26, 2014 (2014)
14. Nissim, R., Brafman, R.I.: Distributed heuristic forward search for multi-agent planning. Journal of Artificial Intelligence Research 51, 293–332 (2014)
15. Shehory, O., Kraus, S.: Methods for task allocation via agent coalition formation. Artif. Intell. 101(1-2), 165–200 (1998)
16. Sipser, M.: Introduction to the theory of computation. Thomson Course Technology (2006)
17. Tambe, M.: Towards flexible teamwork. Journal of Artificial Intelligence Research 7, 83–124 (1997)