Some challenges in Game Theory solvable via CP

Vincent Armant, Kenneth N. Brown, Barry O'Sullivan,
Luis Quesada, Mohamed Siala, and Mohamed Wahbi

firstname.surname@insight-centre.org
Insight Centre for Data Analytics
Department of Computer Science, University College Cork, Ireland

Abstract. Game theory is concerned with reasoning about strategic interaction between self-interested entities. This theory was not originally meant for computational settings, but researchers have observed that the set of concepts developed in this field can be adapted and applied to multi-agent systems in particular [1]. In this paper we elaborate on some challenges encountered when looking at the intersection between game theory and the theory of computer science, and discus the role that *constraint programming* (CP) can play in solving them.

1 Introduction

We first elaborate on *Constraint Programming* and *Game Theory* since the challenges discussed in the paper are in the intersection of these two areas of research. After this elaboration we present the structure of the paper.

1.1 Constraint Programming

The basic idea of *constraint programming*(CP) is that the user states the constraints and a general purpose constraint solver is used to solve them. Constraints are just relations. A constraint satisfaction problem (CSP) states which relations should hold among the given decision variables [2]. A solution is an assignment to the decision variables that satisfies the constraints of the CSP. In a constraint optimisation problem (COP), the solutions are ordered under a given preference relation. The goal, when solving a COP, is not only to find a solution that respects all the constraints but also that satisfies the preferences given to the greatest degree possible.

The *Quantified Constraint Satisfaction Problem* (QCSP) [3] is an extension of the classic CSP, where parameters under the control of the decision maker are modelled as existentially quantified variables since a value (a decision) must be assigned (made) to these variables. All uncertain variables are universally quantified so that decision makers must consider how to preempt every possible assignment to those variables. Of course, such a formulation means that it will be seldom possible for a decision maker to satisfy the constraints of the QCSP since it is likely that some values of the universal (uncertain) variables cannot be preempted. Therefore, in [4], the authors assist the decision maker by abstracting their decision problem so the specific reasons for infeasibility can be focused upon. Their approach is a non-intrusive approach that assumes nothing about the capabilities of the solver but a way of testing the consistency of a QCSP. Such non-intrusive explanation algorithms are the most commonly used in practice, e.g. in ILOG Configurator. The non-intrusive approach is advocated by Junker [5].

Distributed constraint reasoning (DCR) is a powerful framework for representing and solving distributed combinatorial problem applications in multi-agent coordination. These problems are becoming ubiquitous as a result of the unprecedented expansion of the internet and advances in smart devices technologies which, in turn, have an increasingly important impact in industrial and other real-world applications. DCR has recently gained momentum due to its ability to handle many combinatorial problems that are naturally distributed over a set of agents such as distributed scheduling, distributed planning, distributed resource allocation, target tracking in sensor networks, distributed vehicle routing, optimal dispatch in smart grid, etc.

These applications can be solved by a centralised approach once the knowledge about the problem is delivered to a centralised authority. However, in such applications, it may be undesirable or even impossible to gather the whole problem into a single centralised authority. In general, this restriction is mainly due to privacy and/or security requirements: constraints may represent strategic information that should not be revealed to other agents that might be seen as competitors, or even to a central authority. The cost associated with translating all information to a single format may be another reason: in some settings constraints arise from complex decision processes that are internal to an agent and cannot be articulated to a central authority. In addition, sending the whole problem description to a centralised location will create a bottleneck on the communication towards that location.

In DCR a problem is expressed as a distributed constraint network. A distributed constraint network is composed of a group of autonomous agents in which each agent has control of some elements of information about the problem, that is, variables and constraints. Each agent owns its local constraint network; variables owned by different agents are connected by constraints.

Traditionally, there are two large classes of distributed constraint network. The first class considers problems where all constraints are described by boolean relations (*hard* constraints) over the possible assignments of variables they involve. They are called *distributed constraint satisfaction problems* (DisCSP). The second class of problems are *distributed constraint optimisation problems* (DCOP) where constraints are described by a set of *cost* functions for combinations of values assigned to the variables they connect. In DisCSP the goal is to find assignments of values to variables such that all (hard) constraints are satisfied while in DCOP the goal is to find assignments that minimise the objective function defined by the sum of all constraint costs. The *asymmetric distributed constraint satisfaction problem* (ADisCSP) is an extension of the general DisCSP formalism that models constraints that produce different gains (or costs) for the participating agents while allowing agents to keep their utilities private.

An ADisCSP ([6]) is composed of a group of autonomous agents, where each agent has control of some elements of information about the whole problem, that is, a set of variables and constraints. Each agent holds its local constraint satisfaction problem where its variables and constraints are private. Variables in different agents are connected by constraints. Agents must assign values to their variables so that all constraints are satisfied. To achieve this goal, agents check the value assignments to their variables for local consistency and exchange messages to check consistency of their proposed assignments against constraints that contain variables that belong to other agents.

1.2 Game Theory

Game theory studies situations in which multiple agents with conflicting objectives must reach a collective decision. In a strategic game, each player is given a set of actions and has to choose one to perform. A reward is given to a player by a utility function which depends on the actions taken by other players. In many multi-agent problems, agents must interact with each other to achieve a global goal while optimising their own individual preferences. The challenge is to design local control algorithms for the individual agents that ensures the desirable global behaviour while achieving the given system objective. The success of game theory in modelling interaction of selfish agents in multi-agent systems has been proven.

Normal-form games are probably the simplest representation of strategic games [7]. In a normal-form game, each agent chooses an action from its action sets. Each agent has a utility function that specifies its own payoff under each possible combinations of strategies of other players. n-players normal-form games are represented by a n-dimensional matrix whose size is exponential in the number of players/agents. The intractability of this representation is a severe limitation and the question of a compact representation language for agents utility function is of crucial importance.

Motivated by the fact that in many scenarios an agent's utility is directly dependent on only a subset of the total number of agents, *graphical games* have been introduced in [8]. *Graphical games* are a compact representation of normal-form games that use a graphical representation to capture the payoff independence structure of the game. Precisely, the utility of each agent depends on some subset of the other agents decisions rather than on all other agents decisions.

The *hypergraphical games* model [9] provides a compact representation of the problem, in which agent interactions are represented as normal-form strategic games, and the relationship topology between the agents is represented as a hypergraph. Each agent has a set of strategies, and a utility function specifying the agent's payoff under each possible combination of the strategies of its neighbouring agents in the hypergraph. Hypergraphical games are a generalization of *graphical games* [8] where each agent is involved in exactly one subgame.

Boolean games are an expressive and compact class of games for which the goal of the players is modelled by a logical propositional formula. Each player has a set of possible actions by which he can manoeuvre toward the goal. The set of actions is represented by a valuation of the variables controlled of player. The strategy of each player is then represented by the set of valuations over the controlled variables. The goals and the actions of other players may go in the same direction or against each other. Thus, the formula modelling the goal of each player describes outcomes of the game where both the variables controlled by the player and the variables controlled by other players may appear. Boolean games are a more compact description than the normal form game description which represents the game by way of a matrix. The advantage at the description level comes with a computational cost that is paid at the reasoning phase when a solution concept has to be found. A solution concept is a formal rule describing which strategies will be adopted by the players and the result of the game.

Given a set of agents, a coalition structure is a partition of the set of agents into subsets. In hedonic games, the appreciation of an agent over a coalition structure depends only on the coalition he is member of and not how the others are grouped. In that sense, *hedonic games* are more general than matching under preferences since any coalition structure is feasible. Hedonic games have been successfully used in diverse applications such as scheduling group activities, clustering in social networks, and task allocation for wireless agents [10]. Similar to stable matching problems, stability is the most important criterion used to evaluate the coalition structures. Here, stability holds when every agent has no incentive to leave his current coalition to different one.

Constraint games, which use *constraint satisfaction/optimisation problems* (CSP/COP) as the underlying tool for expressing players preferences, have been recently proposed [11]. In a constraint game each player controls a set of finite domain variables and their Cartesian product defines a possible action space for the player. Each player owns a CSP on all players variables which defines satisfaction. Besides the use of hard (global) constraints that provide a crucial expressivity in modelling, constraint games allow the compact expression of most classical games such as congestion games, network games, strategic scheduling, etc.

In a purely selfish environment, rational agents might be tempted to always react to the current context with playing their best strategy, i.e. the strategy that maximises their local utility function. This behaviour might cause instability in the system. The process of stabilising the system corresponds to searching for a Nash Equilibrium (NE), one of the most studied concepts in game theory. A NE corresponds to the situation where the best (dominant) strategy is selected by each agent such that all global goals are achieved, and no agent can improve its payoff by changing its strategy. To model more realistic problems, ϵ-approximate Nash Equilibria (ϵ-NE) are considered, in which no agent can improve its payoff by more than some minimum threshold ϵ.

1.3 Structure of the paper

In this paper we have identified four areas in the field of *Game Theory* where we believe CP can contribute. First we elaborate on the potential contributions in the area of stable matching and hedonic games. Then, we discus why we think CP can play a crucial role in the efficient implementation of Boolean games. We continue with our reasons to believe why distributed constraint solving can be of great help too. And finally, we consider those cases where we need to (automatically) explain the user why it is not possible to achieve a Nash equilibrium.

2 From Matching Under Preferences to Hedonic Games: A Challenging Research Area for Constraint Programming

Many real world problems involve matching preferences between agents while respecting some stability criteria [12]. This family of problems has gained considerable attention as it has a wide range of applications such as assigning doctors to hospitals, students to college, and in kidney exchange programmes. For instance, in *College Admissions* one needs to assign students to colleges while respecting the students' preferences over colleges, the colleges' preferences over students, as well as college quotas. Gale and Shapley introduced the first polynomial time algorithm for solving this problem in their seminal paper [13]. However, when facing real world situations the problem often considers additional optimality criteria. In many cases, the problem becomes intractable and specialised algorithms for solving the standard version are usually hard to adapt. The use of a modular approach such as constraint programming is very beneficial to tackle such cases. Expressing problems involving preferences in CP is extremely beneficial for tackling variants that involve side constraints. We first give a summary of our contributions in [14,15,16,17,18], then we discuss potential research directions for the general case of hedonic games.

2.1 Constraint Programming Approaches for Stable Matching

We consider in [14] the notion of two-sided stability as a global constraint. We first make the observation that the previous CP propositions on two-sided stability problems do not enforce a complete filtering, i.e., Arc Consistency (AC), however they do maintain a weaker form of consistency called Bound Consistency (BC). Next, we propose an incremental algorithm that achieves BC with $O(L)$

time complexity where L is the length of the preference lists, thereby improving the previously best known complexity of $O(c \times L)$ (where c is the maximum quota). We also present, for the first time, an adaptation of the filtering to achieve AC on this global constraint with an additional cost of $n \times L$ (where n is the number of residents).

Based on the BC propagator, we show that the hospital/resident problem with forced and forbidden pairs can be solved in polynomial time. Furthermore, we show that the particular case of this problem for stable marriage can be solved in $O(n^2)$ which improves the previously best complexity by a factor of n^2. Finally, we present a set of experiments to evaluate the filtering efficiency on randomly generated instances. The experimental results show compelling evidence that AC does further prune the search space as compared with BC, however, it considerably slows down the exploration of the search space.

In [15], we tackled the general case of many-to-many stable matching. Our propositions are based on a powerful structure called *rotation*. Informally, given a stable matching M, a rotation is an ordered list of pairs of M such that shifting each agent's match with the next one's partner gives a new stable matching. We leverage some known properties related to rotations in order to propose a novel SAT formulation of the general case of many-to-many stable matching. We show that unit propagation [19] on this formula ensures the existence of a particular solution. Next, we use this property to give an algorithm that maintains arc consistency if one considers many-to-many stable matching as a (global) constraint. The overall complexity for arc consistency is $O(L^2)$ time where L is total input size of all preference lists. Our experimental study on hard instances of sex-equal and balanced stable matching show that our approach outperforms the state-of-the-art constraint programming approach.

2.2 Finding Robust Solutions to Stable Marriage

We introduce the notion of (a, b)-supermatches as a measure of robustness for SM. Informally, a stable matching M is called an (a, b)-supermatch if any a agents decide to break their matches in M, thereby breaking a pairs, it is possible to "repair" M (i.e., find another stable matching) by changing the assignments of those a agents and the assignments of at most b others. This concept is inspired by the notion of (a, b)-supermodels in boolean satisfiability [20] and super solutions in constraint programming [21].

Given a stable matching M, we show that, for every pair $\langle m, w\rangle \in M$ (where m is a man and w is a woman) to remove, the closest stable matching to M that does not contain $\langle m, w\rangle$ can be computed in polynomial time using a powerful structure called graph poset. This observation has led to a polynomial time procedure to verify whether a given stable matching is a $(1, b)$-supermatch. Next, based on this procedure, we design a constraint programming (CP) model, as well as local search (LS) and genetic algorithm (GA) to find the most robust stable matching. Our empirical evaluation on randomly generated instances shows that the local search algorithm is by far the most efficient approach to tackle this problem.

A preliminary version of this work appeared in [16] and a more complete version is accepted for publication in [17].

2.3 Popular Matching and its Variants

We consider here the problem of matching a set of applicants to a set of posts, where each applicant has a preference list, ranking a nonempty subset of posts in order of preference, possibly involving ties [22]. The popular matching problem is to determine if a given instance admits a popular matching and to find such a matching, if one exists. We say that a matching M' is *more popular* than M if the number of applicants that prefer M' to M exceeds the number of applicants that prefer M to M'. A matching M is *popular if and only if* there is no matching M' that is more popular than M.

The popular matching problem has never been studied in the context of CP. We study this problem and propose the first CP formulation of it. We consider two cases of the problem of popular matching - instances with and without ties in the preference lists - and show that one can elegantly encode these problems using the Global Cardinality Constraint *gcc* [23].

A preliminary version of this work appeared in [18].

2.4 From Matching Under Preferences to Hedonic Games

Many research work have been done in the literature to understand under which conditions stable coalitions exist in hedonic games. These restrictions are related to the way the preferences are expressed. Moreover, some weaker forms of stability are often considered when no stable coalition exist [10,24].

Challenge 1 *How CP can help in determining the existence of stable coalitions in hedonic games?*

While CP has been successfully applied to a variety of matching under preferences problems, the extension of these proposition to hedonic games have never been considered. We think that the flexibility of CP makes it a very promising candidate to address classic questions in hedonic games. Example of such question concern the existence of stable coalition and finding one if it exists.

3 Extending CP framework for solving efficiently Boolean games

Challenge 2 *Can we come up with new CP techniques to solve Boolean games efficiently?*

In the literature, most attention has focused on designing new preference schemes [25,26,27] and representations [28] extending the Boolean games framework. Because of the computational hardness associated with solving Boolean games, little attention has been paid to solving Boolean games efficiently. Recent work has proposed an answer-set programming (ASP) approach [29]. The declarative paradigm of ASP facilitates the formulation of the game into an ASP program that naturally expresses the goal of agents as a set of disjunctive rules. However, this framework does not scale to more than 100 agents. This presents a challenge for future study, namely addressing this gap in the literature by proposing new CP techniques for solving a wide range of solution concepts and preferences envisaged for Boolean games.

On the other hand, a Boolean game can be seen as a multi-agent system description where the solutions returned by the agents entail the satisfaction of each agent preferences as well as the optimisation of a global objective function. In this context, the Distributed Constraint Optimisation Problem (DCOP) [30] and the distributed Constraint Satisfaction Problem (DisCSP) [31] have been used as a vehicle for using the CP framework to solve graphical games. The message-passing protocol at the core of these approaches enforces the privacy of local knowledge [32]. The protocol does not preserve the privacy of the players' strategies in a coalition of agents. As a consequence, an agent can make a decision based on the action of other agents that is not part of its coalition. This raises an additional research challenge, namely the development of extended privacy schemes for DCOP and the DisCSP in order to solve more realistic boolean games.

4 Distributed Constraint Reasoning for Game Theory

In a distributed setting, such as that in multi-agent systems, agents are usually not willing to reveal their private information (including beliefs, preferences, and utilities) to other agents that can be seen as competitors. Therefore, a centralised solving process that requires that all relevant information to be gathered into a single agent (or computer) is not desirable. Additionally, centralising all information in a distributed setting throughout the network into a single agent is not always feasible. Thus, given the multi-agent setting, algorithms to compute solutions should be distributed, to avoid the need for agents to reveal potentially private information.

Early work on solving games (finding an equilibrium) focused on identifying graph topologies that allowed polynomial-time solutions, based on algorithms for Bayesian Network inference [8]. More recent approaches focus on arbitrary graphs with cyclic dependencies, and represent the problem as ADisCSP [6], maintaining the individual utility functions as extensional table constraints [33]. For dense graphical games with large strategy sets, this encoding of the table constraints becomes very expensive in the number or size of messages that need to be exchanged.

In [31], we developed a new model of hypergraphical game as an ADisCSP based on a new global constraint that encodes the requirement of finding an approximate best strategy given the decisions of other agents in its neighbourhood. We also introduced asymmetric asynchronous backtracking, AABT, a new algorithm for solving ADisCSP with global constraints using intelligent backtracking to avoid thrashing. AABT is then used to solve the problem of finding an ϵ-NE in hypergraphical games formulated as an ADisCSP.

Challenge 3 *How to solve Game Theory using Distributed Constraint Reasoning?*

The challenges posed by solving distributed optimisation problems include dealing with resource restrictions (such as limits on time, space and communication), privacy requirements, exploiting opportunities for cooperation, and designing conflict resolution strategies. Thus, compact representations for games that facilitate the modelling of agents with limited resources while providing solutions that keep their utilities private need to be designed. The usage of methods developed in the field of constraint programming for modelling and solving strategic games needs to be studied. Mainly, the usage of rich constraints language to model different games and the incorporation of elaborated search strategies for constraints games needs to be investigated. DCR is the natural interface between constraint programming and game theory research areas. Actually, DCR represents an opportunity for both research communities to collaborate to tackle challenges in solving the arising distributed applications. We believe that DCR has the potential to contribute actively in this area of research.

5 Computing preferred explanations in the Rational Verification framework

In [1] the authors are concerned with the question of how we should think about the issues of correctness and verification in multi-agent systems. They argue that the classical view of correctness is not appropriate for multi-agent systems. In a multi-agent setting it is more appropriate to ask what behaviours the system will exhibit under the assumption that agents act rationally in pursuit of their preferences. They advance the paradigm of rational verification for multi-agent systems, as a counterpart to classical verification. Rational verification is concerned with establishing whether a given temporal logic formula ϕ is satisfied in some or all game-theoretic equilibria of a multi-agent system, i.e. whether the system will exhibit the behaviour ϕ under the assumption that agents within the system act rationally in pursuit of their preferences/goals.

As explained in [1], the basic idea of equilibrium checking is that, instead of asking whether a given temporal formula ϕ is satisfied on some possible run of the system, we instead ask whether it is satisfied on some run corresponding to a Nash equilibrium of the system. Informally, one can understand this as asking whether ϕ could be made true as the result of rational choices by agents within the system. One could even consider the question of verifying whether a given profile represents a Nash equilibrium.

Challenge 4 *How to explain the non existence of (a run corresponding to) a Nash equilibrium?*

We argue for a focus on those scenarios where we are confronted with unsatisfiable queries. That is, cases where ϕ cannot be made true as the result of rational choices by agents within the system. In those scenarios it might be desirable to explain the reason of the inconsistency. In order to achieve this goal we propose to take advantage of the extra level of expressivity that we have in the Quantified Constraint Satisfaction framework [3] and cast the computation of explanation for scenarios with no run corresponding to a Nash equilibrium as the computation of explanations for an unsatisfiable quantified constraint satisfaction problem.

Challenge 5 *How to rank the explanations to the non existence of (a run corresponding to) a Nash equilibrium?*

The framework presented in [4] also covers the generation of preferred explanations in a QCSP setting, where both total and partial orders amongst the requirements of a QCSP and efficient algorithms for each case are proposed. As there can be exponentially many ways to explain why there is no run corresponding to a Nash equilibrium, we believe that this handling of preference can come in handy to give priority to more important explanations.

6 Conclusion

In many societal challenges we face today, including cybersecurity, electronic commerce, and game theory on social networks, we must make more intelligent decisions. In particular, cybersecurity has become an increasingly important problem domain due to the ubiquity of the internet. Many challenges in this area can be modelled and reasoned with using game theory techniques. However, this is a multidisciplinary domain with connections to many research areas including security, optimisation, distributed systems, and machine learning, and it should be addressed from different perspectives by a collaborative approach.

In this paper, we have elaborated on the intersection between game theory and the theory of computer science. We have discussed some challenges encountered in this intersection and elaborated on the role that CP can play in solving them.

Acknowledgements

This publication has emanated from research supported in part by a research grant from Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289.

References

1. Wooldridge, M., Gutierrez, J., Harrenstein, P., Marchioni, E., Perelli, G., Toumi, A.: Rational verification: From model checking to equilibrium checking. In Schuurmans, D., Wellman, M.P., eds.: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA., AAAI Press (2016) 4184–4191
2. Rossi, F., Beek, P.v., Walsh, T.: Handbook of Constraint Programming (Foundations of Artificial Intelligence). Elsevier Science Inc., New York, NY, USA (2006)
3. Chen, H.: The Computational Complexity of Quantified Constraint Satisfaction. PhD thesis, Cornell (August 2004)
4. Mehta, D., O'Sullivan, B., Quesada, L.: Extending the notion of preferred explanations for quantified constraint satisfaction problems. In Leucker, M., Rueda, C., Valencia, F.D., eds.: Theoretical Aspects of Computing - ICTAC 2015 - 12th International Colloquium Cali, Colombia, October 29-31, 2015, Proceedings. Volume 9399 of Lecture Notes in Computer Science., Springer (2015) 309–327
5. Junker, U.: Quickxplain: Preferred explanations and relaxations for over-constrained problems. In: Procs of AAAI 2004. (2004) 167–172
6. Brito, I., Meisels, A., Meseguer, P., Zivan, R.: Distributed Constraint Satisfaction with Partially Known Constraints. Constraints **14** (2009) 199–234
7. Shoham, Y., Leyton-Brown, K.: Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations. Cambridge University Press, New York, NY, USA (2008)
8. Kearns, M.J., Littman, M.L., Singh, S.P.: Graphical models for game theory. In: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence. UAI'01, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (2001) 253–260
9. Papadimitriou, C.H., Roughgarden, T.: Computing correlated equilibria in multi-player games. Journal of the ACM (JACM) **55**(3) (August 2008) 14:1–14:29
10. Brandt, F., Conitzer, V., Endriss, U., Lang, J., Procaccia, A.D., eds.: Handbook of Computational Social Choice. Cambridge University Press (2016)
11. Nguyen, T.V.A., Lallouet, A., Bordeaux, L.: Constraint games: Framework and local search solver. In: Proceedings of the 2013 IEEE 25th International Conference on Tools with Artificial Intelligence. ICTAI'13, Washington, DC, USA, IEEE Computer Society (2013) 963–970
12. Manlove, D.F.: Algorithmics of Matching Under Preferences. Volume 2 of Series on Theoretical Computer Science. WorldScientific (2013)
13. Gale, D., Shapley, L.S.: College admissions and the stability of marriage. American mathematical monthly (1962) 9–15
14. Siala, M., O'Sullivan, B.: Revisiting two-sided stability constraints. In: Integration of AI and OR Techniques in Constraint Programming - 13th International Conference, CPAIOR 2016, Banff, AB, Canada, May 29 - June 1, 2016, Proceedings. (2016) 342–357
15. Siala, M., O'Sullivan, B.: Rotation-based formulation for stable matching. In: Principles and Practice of Constraint Programming - 23rd International Conference, CP 2017, Melbourne, Australia, August 28 - September 1, 2017, Proceedings. (2017)
16. Genc, B., Siala, M., O'Sullivan, B., Simonin, G.: Robust stable marriage. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA. (2017) 4925–4926
17. Genc, B., Siala, M., O'Sullivan, B., Simonin, G.: Finding robust solutions to stable marriage. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, August 2017. (2017)
18. Chisca, D.S., Siala, M., Simonin, G., O'Sullivan, B.: A CP-based approach for popular matching. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA. (2016) 4202–4203
19. Moskewicz, M.W., Madigan, C.F., Zhao, Y., Zhang, L., Malik, S.: Chaff: Engineering an efficient SAT solver. In: Proceedings of the 38th Design Automation Conference, DAC 2001, Las Vegas, NV, USA, June 18-22, 2001. (2001) 530–535

20. Ginsberg, M.L., Parkes, A.J., Roy, A.: Supermodels and robustness. In: In AAAI/IAAI. (1998) 334–339
21. Hebrard, E.: Robust solutions for constraint satisfaction and optimisation under uncertainty. PhD thesis, University of New South Wales (2007)
22. Abraham, D.J., Irving, R.W., Kavitha, T., Mehlhorn, K.: Popular matchings. SIAM Journal on Computing **37**(4) (2007) 1030–1045
23. Régin, J.C.: Generalized arc consistency for global cardinality constraint. In: AAAI. (1996) 209–215
24. Aziz, H., Savani, R.: Hedonic games. In: Handbook of Computational Social Choice. Cambridge University Press (2016) 356–376
25. Dunne, P.E., van der Hoek, W., Kraus, S., Wooldridge, M.: Cooperative boolean games. In: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2, International Foundation for Autonomous Agents and Multiagent Systems (2008) 1015–1022
26. Wooldridge, M., Endriss, U., Kraus, S., Lang, J.: Incentive engineering for boolean games. Artificial Intelligence **195** (2013) 418–439
27. Aziz, H., Harrenstein, P., Lang, J., Wooldridge, M.: Boolean hedonic games. In: KR. (2016) 166–175
28. Gao, T., Gutierrez, J., Wooldridge, M.: Iterated boolean games for rational verification. In: Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems. AAMAS'17, Richland, SC, International Foundation for Autonomous Agents and Multiagent Systems (2017) 705–713
29. De Clercq, S., Bauters, K., Schockaert, S., Mihaylov, M., Nowé, A., De Cock, M.: Exact and heuristic methods for solving boolean games. Autonomous Agents and Multi-Agent Systems **31**(1) (Jan 2017) 66–106
30. Pearce, J.P., Maheswaran, R.T., Tambe, M.: Solution sets for dcops and graphical games. In: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems, ACM (2006) 577–584
31. Wahbi, M., Brown, K.N.: A distributed asynchronous solver for nash equilibria in hypergraphical games. In: ECAI 2016: 22nd European Conference on Artificial Intelligence, IOS Press (2016)
32. Armant, V., Simon, L., Dague, P.: Distributed tree decomposition with privacy. In: Principles and Practice of Constraint Programming, Springer (2012) 102–117
33. Grubshtein, A., Meisels, A.: Finding a nash equilibrium by asynchronous backtracking. In: Proceedings of the 18th International Conference on Principles and Practice of Constraint Programming. CP'12, Berlin, Heidelberg, Springer-Verlag (2012) 925–940