Constraint Optimisation Techniques for Real-World Applications

Ph.D. Thesis Summary

Filippo Bistaffa, University of Verona

Constraint optimisation [7] represents a fundamental technique that allows to address a wide variety of optimisation problems in several contexts. Constraint optimisation techniques have been successfully employed in *Multi-Agent Systems* (MAS) to face a number of multi-agent coordination challenges such as planning, scheduling, resource allocation and satellite management. In this thesis we focus on *Coalition Formation* (CF) [9], a particular application of constraint optimisation that represents one of the key approaches for coordination in MAS.

1 Coalition Formation

CF aims at establishing collaborations in MAS with multiple entities provided with common or individual objectives. It involves the coming together of multiple, possibly heterogeneous, agents into groups, called *coalitions*, in order to achieve either their individual or collective goals, whenever they cannot do so on their own. Sandholm et al. [9] identify the key computational tasks involved in the CF process. As a first step, a *characteristic function* associates each possible coalition to a value, which quantifies how well or bad a given group of agents performs together. For example, in the *Collective Energy Purchasing* (CEP) scenario [3, 5] the value associated to a group of agents is the total cost of the energy that they consume as a collective. The second fundamental aspect of CF, denoted as *Coalition Structure Generation* (CSG), involves the computation of the best *partition* of the set of agents, i.e., the one that maximises the sum of the values of the associated coalitions. Notice that the number of possible partitions (namely, *coalition structures*) is equal to the number of ways in which a set of n agents can be partitioned, a quantity that grows exponentially with respect to n. For this reason, *State of the Art* (SoA) algorithms are limited to a few tens of agents, and, to the best of our knowledge, they have never been applied to realistic CF scenarios.

The challenges posed by CF in real-world scenarios represent the main objects of study this thesis, whose main contributions will be discussed in the following sections.

2 Graph-constrained CF

In many real-world applications, there are constraints that may limit the formation of some coalitions. In this thesis we focus on a specific type of constraints that encodes synergies or relationships among the agents and that can be expressed by a graph, where nodes represent agents and edges encode the relationships between the agents. Such constraints are present in several real-world scenarios, such as social or trust constraints (e.g., energy consumers who prefer to group with their acquaintances in forming energy cooperatives [3, 5], or commuters sharing rides with their friends [2, 6]), and physical constraints (e.g., emergency responders may join specific teams in disaster scenarios where only certain routes are available). Hereafter, we refer to CF in these scenarios as *Graph-Constrained Coalition Formation* (GCCF).

In this thesis, we are primarily interested in developing GCCF solutions that are deployable in the real world. Hence, our main objective is to develop an algorithm that can solve problems using real-world data (rather than focusing on purely synthetic environments) in scenarios involving hundreds or thousands of agents. To achieve this objective, one of the fundamental contributions of this thesis is to exploit the structure of such scenarios in order to represent the CF problem in a polynomial amount of space (in contrast to the exponential memory requirements inherent in the standard CF).

In our first domain of interest, i.e., the CEP scenario [3, 5], agents form coalitions to buy energy together at cheaper prices. Specifically, each agent is characterised by an energy consumption profile that represents its energy consumption throughout a day. Each coalition is associated to the total cost that the group would incur if they bought energy as a collective. Such a scenario involves a thoughtful commitment by the agents belonging to a group, due to the purchasing of a significant amount of goods (i.e., energy in this particular case). Henceforth, it is reasonable to assume that agents may desire to participate to such contract along with trusted people, i.e., people engaged in a social connection. On these premises, it is natural to formalise the CEP as a GCCF, in which purchasing groups are formed in order to establish a friend-of-friend relationship.

3 Search-based GCCF

We propose a novel search-based approach to solve GCCF that employs the concept of *edge contraction.* Intuitively, the contraction of an edge of the graph represents the merge of the corresponding coalitions. Edge contraction is the fundamental operation that allows us to represent the entire solution space of the GCCF problem. More precisely, we propose a technique to model such solution space as a rooted tree, in which each feasible coalition structure is represented exactly once. Moreover, we theoretically prove the completeness and the absence of any redundancy in our model. Such search tree can then be explored with any known traversal technique. We adopt a Depth-First Search (DFS) approach, since it is characterised by polynomial memory requirements. Crucially, since we consider closed-form characteristic functions (hence avoiding the exponentiality inherent in the classic formulation of CF), the memory requirements of our approach are polynomially bounded. This property allows us to tackle large-scale problems with thousands of agents, in contrast with previous GCCF approaches. Furthermore,we propose a technique that helps prune significant parts of the search space when the characteristic function belongs to a general class of functions (denoted as $m + a$). Such method is then employed within CFSS [3, 5], our branch and bound algorithm to compute the optimal solution for any GCCF problem based on an $m + a$ function. Our empirical evaluation considers both realistic and synthetic network topologies. Results show that CFSS outperforms DyCE [10], the SoA GCCF algorithm. Finally, our algorithm provides approximate solutions with good quality guarantees (i.e., whose values are at least 88% of the optimal) for systems of unprecedented scale (i.e., more than 2700 agents).

4 Social Ridesharing

In the above discussed scenarios, we have addressed CF problems by considering a type of constraints induced by a graph. On the other hand, when coalitions are mapped to physical objects with limited capacity, it is natural to enforce a constraint on the *cardinality* of such coalitions. A straightforward real-world example is *ridesharing*, in which agents model users that need to commute across a geographical space (usually within a city), and coalitions represent cars that are shared among multiple agents with the objective of reducing travel costs, by sharing rides. In this case, the cardinality of coalitions is limited by the number of seats in each car, which is usually quite low (e.g., 5 seats). In particular, we focus on a ridesharing scenario that involves a set of agents, connected through a social network, which necessitate to commute within a urban environment. In this context, agents arrange one-time rides at short notice, travelling together with friends, in contrast with complete strangers. This assumption is motivated by a clear tendency among ridesharing companies, which favour the formation of groups of users that are connected in such network. In fact, Uber and Lyft incentivise users to share rides with their friends, showing that social relationships play a central role in the ridesharing scenario, which is then referred as *Social Ridesharing* (SR) [2, 6].

Within such scenario, we first address the optimisation problem of minimising the total cost of all the cars formed by the system. As a consequence, we define the value of each coalition as the travel cost of the associated car. Specifically, we present the first model that encodes the above discussed scenario as a GCCF problem, and we formally define the value of each coalition on the basis of the spatial and temporal preferences of the agents. Subsequently, we show how to solve the GCCF problem associated to the SR scenario by means of a modified version of CFSS, i.e., SR-CFSS. SR-CFSS employs a different bounding technique with respect to the original version, since the SR characteristic function is not an $m + a$ function.

We empirically evaluate SR-CFSS on realistic datasets for both spatial and social data, i.e., GeoLife by Microsoft Research and Twitter. Results show that our approach can produce significant cost reductions (up to -36.22%) and it features a good scalability, computing approximate solutions for very large systems (i.e., up to 2000 agents) and good quality guarantees (i.e., at least 71% of the optimal) within minutes.

5 Payments for SR

In addition, we tackle the problem of splitting the travel costs corresponding to each car among its passengers. Payoffs (corresponding to cash payments for sharing trip costs) to the commuters need to be computed given their distinct needs (e.g., shorter/longer trips), roles (e.g., drivers/riders, less/more socially connected) and opportunity costs (e.g., taking a bus, their car, or a taxi). One key aspect of payment distribution in CF is the game-theoretic concept of stability, which measures how agents are keen to maintain the provided payments instead of deviating to a configuration deemed to be more rewarding from their individual point of view.

We achieve this by means of the PK (Payments in the Kernel) algorithm [2], our method to compute a kernel-stable allocation, given a coalition structure that is a solution to the SR problem. Our method computes payments for 2000 agents in less than an hour and it is 84 times faster than the SoA in the best case. Our approach is a practical solution technique for large-scale systems thanks to a speed-up of 10.6 on a 12-core machine with respect to the serial approach. Finally, we develop new insights into the relationship between payments incurred by a user by virtue of its position in its social network and its role (rider or driver).

6 Constraint optimisation for GCCF

The techniques discussed so far perform particularly well under the assumption that the value of each coalition can be expressed by means of a closed-form function (i.e., a general expression that, for each coalition, provides its value on the sole basis of its members), and it is possible to derive a method to compute an upper bound for such function, in order to apply the branch and bound approaches discussed above. However, in some GCCF scenarios it may be difficult (or not possible at all) to meet these premises, hence the application of the CFSS algorithm may be not convenient in certain settings.

In the optimisation literature, Dynamic Programming (DP) historically represents the counterpart approach with respect to search, especially in the context of GCCF [10]. This warrants the study of an approach for GCCF based on DP, with the objective of developing a solution method that overcomes the drawbacks of previously discussed algorithms. Now, our objective is the development of a DP solution framework for constraint optimisation (and, specifically, for GCCF) with a particular focus on the runtime performance. In recent years, Graphics Processing Units (GPUs) have been successfully used to speed-up the computation in different applications [8], achieving performance improvements of several orders of magnitude in fields including computer vision, human-computer interaction, and artificial intelligence.

Along these premises, we propose CUBE (CUda Bucket Elimination), a highly parallel implementation of Bucket Elimination (BE) [7], one of the most important constraint optimisation approaches based on DP. In the design of CUBE, we aim at developing a general, high-performance GPU framework that allows us to deal with the computational effort inherent in several BE-based algorithms.

CUBE achieves this objective with a novel preprocessing algorithm that enables highly-optimised memory management. We avoid unnecessary, expensive memory accesses by means of a technique that allows threads to efficiently locate their input data, and by taking advantage of the fastest memory in the GPU hierarchy. Moreover, CUBE is not limited by the amount of GPU memory, as it can process large tables by splitting them into manageable chunks that meet the memory capabilities of the GPU. Hence, our solution can tackle large-scale real-world problems. Our experiments on a realistic dataset show that CUBE achieves a speed-up up to 696× w.r.t. the CPU version of BE, and it is up to two orders of magnitude faster than other recent GPU approaches.

Finally, CUBE is then employed to solve COP-GCCF [1], the first approach that models GCCF as a Constrained Optimisation Problem (COP) [7]. Results show that our approach outperforms SoA algorithms on sparse graphs, both in terms of runtime and memory. In particular, COP-GCCF allows to compute solutions at least one order of magnitude faster than counterpart approaches using Twitter as a graph topology.

7 Publications

All the main parts of this thesis have been published in very prestigious international AI journals (respectively in [3], [2], [4], and [1]), and top international AI conferences (i.e., AAMAS [5] and AAAI [6]). See Ph.D. thesis for a full list of publications.

References

[1] F. Bistaffa and A. Farinelli. "A COP Model for Graph-Constrained Coalition Formation". In: *JAIR* 62 (2018).

[2] F. Bistaffa, A. Farinelli, G. Chalkiadakis, and S. D. Ramchurn. "A Cooperative Game-Theoretic Approach to the Social Ridesharing Problem". In: *AIJ* 246 (2017).

[3] F. Bistaffa, A. Farinelli, J. Cerquides, J. Rodríguez-Aguilar, and S. Ramchurn. "Algorithms for Graph-Constrained Coalition Formation in the Real World". In: *ACM TIST* 8.4 (2017).

[4] F. Bistaffa, N. Bombieri, and A. Farinelli. "An Efficient Approach for Accelerating Bucket Elimination on GPUs". In: *IEEE CYB* 47.11 (2017).

[5] F. Bistaffa, A. Farinelli, J. Cerquides, J. Rodríguez-Aguilar, and S. Ramchurn. "Anytime Coalition Structure Generation on Synergy Graphs". In: *AAMAS*. 2014.

[6] F. Bistaffa, A. Farinelli, and S. Ramchurn. "Sharing Rides with Friends: a Coalition Formation Algorithm for Ridesharing". In: *AAAI*. 2015.

[7] R. Dechter. *Constraint processing*. Morgan Kaufmann, 2003.

[8] R. Farber. *CUDA Application Design and Development*. Elsevier, 2012.

[9] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohmé. "Coalition structure generation with worst case guarantees". In: *AIJ* 111.1 (1999).

[10] T. Voice, S. Ramchurn, and N. Jennings. "On coalition formation with sparse synergies". In: *AAMAS*. 2012, pp. 223–230.