Multi-Agent Simulation Framework for Large-Scale Coalition Formation

Pavel Janovsky
Kansas State University
Department of Computer Science
Manhattan, Kansas 66506, USA
Email: janovsky@ksu.edu

Scott A. DeLoach
Kansas State University
Department of Computer Science
Manhattan, Kansas 66506, USA
Email: sdeloach@ksu.edu

***Abstract*—Coalition formation, a key factor in multi-agent cooperation, can be solved optimally for at most a few dozen agents. This paper proposes a general approach to find sub-optimal solutions for a large-scale coalition formation problem containing thousands of agents using multi-agent simulation. We model coalition formation as an iterative process in which agents join and leave coalitions, and we propose several valuation functions that assign values to the coalitions. We propose several coalition selection strategies that agents may use to decide whether or not to leave their current coalition and which coalition to join. We also show how these valuation functions and coalition selection strategies represent specific coalition formation applications. Finally, we show almost-optimal performance of our algorithms in small-scale scenarios by comparing our solutions with an optimal solution, and we show stable performance in a large-scale setting in which searching for the optimal solution is not feasible.**

***Index Terms*—Large-scale coalition formation, multi-agent simulation.**

1. Introduction

Coalition formation is a process in which multiple autonomous agents create cooperating groups called *coalitions* in order to achieve individual or group goals. Agents can form groups to achieve tasks that require their cooperation, to optimize their expenses, or to share resources. Since so many people are connected to the web via their personal devices, coalition formation can play major role in scenarios such as collective purchasing or resource sharing.

Coalition formation typically requires formation of a *coalition structure* (CS), which is a set of coalitions such that each agent belongs to a coalition in CS. Each coalition is assigned a *value*, and CS with the highest aggregate value of coalitions, i.e. an optimal CS, is determined. For n agents there are $O(n^n)$ possible CSs [1], and determining the optimal CS has been proven to be an NP-complete problem [2]. Algorithms that find the optimal solution can do so in a reasonable time (in a matter of hours) for only approximately a few dozen agents. For applications such as collective purchasing or resource sharing this scale is not sufficient.

In this paper we address *large-scale* coalition formation. We formally define this problem in Section 2. We assume that the number of agents is in range of thousands. For such a problem, state-of-the-art algorithms cannot find the optimal solution in a feasible time, so we propose to find suboptimal solutions using multi-agent simulation. Multi-agent simulation allows us to observe the process of forming coalitions in an iterative manner. While coalition formation is typically approached as a single-step task that finds a CS, it is beneficial to model coalition formation as a dynamic process where coalitions change over time. In such a process the agents can utilize information about previous and current values of coalitions to support their decision to leave the current coalition and join a new coalition. We propose a general framework that models coalition formation as a dynamic, iterative process. Our framework can be used to simulate real world applications of coalition formation.

The approach presented in this paper provides the following contributions:

1) *A framework for evaluating coalition formation strategies that uses multi-agent simulation* as described in Section 3. The framework can be used to simulate real-world scenarios of coalition formation. We show two examples of such scenarios in Section 3.1 along with their representation in the framework in Section 3.2. We discuss the practical meaning of our framework in Section 5.
2) *The capability to evaluate large-scale coalition selection strategies on scenarios consisting of thousands of agents.* Since the solutions reflect the decision making of single agents in the dynamic coalition formation process (see Section 3.3), optimality is not guaranteed. Thus, we show how to evaluate proposed strategies by comparing them against optimal solutions for small numbers of agents (up to 20) and then demonstrating that those strategies are stable in large-scale scenarios with up to 10,000 agents (see Section 4). We also show that in majority of the tested instances our proposed strategies perform similarly or better than C-Link [3], a state-of-the-art coalition formation algorithm.

2. Problem Statement

We studied the problem in which large numbers of agents create coalitions. Specifically we considered the number of agents ranging from 2 to 10,000. We refer to the problem as *large-scale coalition formation*, and we define it as follows.

Let us consider a set of agents $A = \{a_1, a_2, \ldots, a_n\}$ where n is the number of agents. The task is to find a coalition structure $CS = \{C_1, C_2, \ldots, C_m\}$, which is a set of m coalitions C_j, where each agent is contained in a single coalition. This condition is formally defined as

$$\forall i \in \langle 1, n \rangle \ \exists! \ j \in \langle 1, m \rangle : a_i \in C_j. \tag{1}$$

In order to measure quality of CS, we further define the value of CS as

$$v(CS) = \sum_{C \in CS} v(C) \tag{2}$$

where $v(C)$ is a value assigned to the coalition C by a valuation function, which will be defined in Section 3.2. In order to compare solutions of problem instances, we define a gain of CS as

$$g(CS) = \frac{v(CS) - v(CS_0)}{n} \tag{3}$$

where CS_0 denotes a coalition structure containing only coalitions of size one. The gain shows how much, on average, each agent in CS benefits from participating in a coalition. This metric shows the quality of a solution from the perspective of a single agent.

As defined in [3], we also use a gain ratio

$$gr(CS) = \frac{g(CS)}{g_{opt}} \tag{4}$$

where g_{opt} denotes the gain of an optimal solution obtained by a dynamic programming algorithm [4]. Note that g_{opt} and $gr(CS)$ can only be found in the small-scale scenarios due to limitations of optimal algorithms. We use the gain and gain ratio metrics to compare the quality of our solutions.

3. Methodology

We propose a general framework that can model and solve specific applications of the coalition formation problem. We modeled coalition formation as an iterative process in which the agents leave and join coalitions in an iterative fashion. The algorithm for this process is depicted in Algorithm 1 and works as follows.

First the simulator is initialized. Initialization consists of following steps. Agents are created, and each agent initially forms a singleton coalition (line 1). Interest vectors of size k are then assigned to agents (line 2). Interest vectors are essential to the simulation because elements of an interest vector express specific interests of the agent. In various problem applications these interest vectors may represent the amount of resources owned or requested by the agent. Next, a strategy is assigned to each agent (line 3). This strategy is later used to determine whether or not the agent should leave a current coalition and which coalition it should join. A new coalition structure is then created (lines 4 and 5). This structure holds all agents grouped in current coalitions. Finally, the evaluation agent and a valuation function are initialized. The evaluation agent is responsible for evaluating all coalitions and announcing coalition rankings in each iteration based on the specified valuation function. The time complexity of this initialization step (lines 1 to 6) is $O(n \cdot k)$.

Algorithm 1 Multi-agent simulation of coalition formation

Input: number of agents n, number of iterations N, size of interest vectors k.
Output: CS with highest gain.

1: create n agents
2: assign interest vectors of size k to agents
3: assign strategies to agents
4: initialize new coalition structure CS
5: create a coalition for each agent and add it to CS
6: initialize evaluation agent and valuation function
7: **for** iteration in 1:N **do**
8:     **for all** agents in random order **do**
9:         **if** agent.strategy.leaveCoalition() **then**
10:             leave current coalition, update its value
11:             newcoalition $\leftarrow$
12:                 agent.strategy.pickCoalition()
13:             update value of newcoalition
14:         **end if**
15:     **end for**
16:     evaluate all coalitions
17:     announce the ranking of coalitions
18:     store current CS
19: **end for**
20: **return** best CS

After the initialization step, the simulation begins with a first iteration. In every iteration the agents use their strategies to decide whether to leave their current coalition and, if so, which coalition to join (lines 9 to 12). Then all remaining coalitions are evaluated by the evaluation agent, and the ranking of coalitions is announced (lines 16 and 17). Finally, the current CS is stored (line 18). Each agent in each iteration accesses during its decision making at most n other agents' interest vectors, regardless of the agents' grouping in coalitions. Therefore the worst-case time complexity of Algorithm 1 is $O(N \cdot n^2 \cdot k)$. At the end of the simulation, the best CS is selected out of all stored CSs, and returned as the solution (line 20).

3.1. Applications

In this section we discuss two general applications of large-scale coalition formation. We also discuss the approach we took to model these applications in our framework. Other applications of coalition formation can be modeled in the framework following our approach.

- **Resource sharing** - Agents operate with several resources, and they can have either a surplus or a shortage of each resource. Agents with a surplus try to form coalitions with agents with a shortage so that the surplus amount can be transferred to (bought by) agents with shortages. We use interest vectors to store surplus or shortage of each resource. The value of a coalition then depends on the amount of resources shared.
- **Collective energy purchasing** - Electricity can be bought either at a spot or forward market. A spot market provides electricity according to the current amount requested. However, agents can exploit reduced tariffs at forward markets that sell constant amounts of electricity for long periods of time. In order to exploit the forward market, the aggregate energy profile of the buyers (i.e. the hourly energy requirements for a day) must be flat. Agents can form coalitions in order to flatten their aggregate daily energy profile. We model energy purchasing by storing the agents' energy profiles in their interest vectors. A value of a coalition then depends on the flatness of its aggregate interest vector.

3.2. Valuation functions

A valuation function $f : C \rightarrow \mathbb{R}$ assigns a value v to each coalition C. We propose market-based valuation function to represent resource sharing. We also discuss collective energy purchasing [5] and normally distributed coalition structures [6] valuation functions.

- **Market-based valuation function** is used in the resource sharing application. The value of a coalition is determined by the amount of resources shared (bought and sold) within the coalition. More specifically,

$$v(C) = \sum_{l=1}^{k} min(b_C^+[l], b_C^-[l]) + \kappa(C) \quad (5)$$

  where $b_C^+[l]$ is the positive balance for resource l, which is the sum of surpluses of resource l over all agents in coalition C, and $b_C^-[l]$ is an absolute value of the negative balance computed with the shortages, respectively. $\kappa(C) = -|C|^\gamma$ was proposed in [3] to represent the penalty for the coalition size. The penalty prevents the grand coalition (coalition containing all n agents) from forming, and it represents the difficulty associated with a cooperation of a large number of agents.
- **Collective energy-purchasing valuation function** was proposed in [5]. The value of an expected payment (coalition value) for coalition C is given by

$$v(C) = \sum_{t=1}^{T} q_S^t(C) \cdot p_S + T \cdot q_F(C) \cdot p_F + \kappa(C) \quad (6)$$

  where p_S and p_F represent unit prices at the spot and forward markets, respectively, $q_S^t(C)$ represents the amount of energy to be bought at the spot market at time t, and $T \cdot q_F(C)$ represents the total amount of energy to be bought at the forward market for time interval T (in our experiments, $T = 24$ represents a length of a daily energy profile). $\kappa(C) = -|C|^\gamma$ [3] captures the penalty for the coalition size. Unlike the market-based valuation function, the collective energy-purchasing valuation function creates strong interdependence between the elements of interest vectors.

  An algorithm given in [5] computes optimal energy amounts for a coalition given the coalition's aggregate energy profile, which we store in the coalition's aggregate interest vector. Using this algorithm, we obtain energy amounts $q_S^t(C)$ and $q_F(C)$ that we use to compute the coalition value $v(C)$.
- **Normally Distributed Coalition Structures (NDCS)** is a challenging valuation function benchmark proposed by [6]. The value of a coalition is drawn from a normal distribution $N(\mu, \sigma)$ with $\mu = |C|$ and $\sigma = \sqrt{|C|}$. We include NDCS here for the sake of comparison of our approach with the C-Link algorithm, which is a hierarchical clustering approach for coalition formation proposed in [3].

3.3. Coalition selection strategies

Agents use coalition selection strategies to make two decisions: whether to leave a coalition and which (if any) of the existing coalitions to join. We propose *mixed* and *local search* coalition selection strategies. We also discuss *coalition value-based* strategy and *random* strategy. We assume that the agents have complete information about the system including other agents' interest vectors.

- **Coalition value-based strategy** advises the agent to join a coalition that maximally benefits from the addition of the agent. For agent i, the new coalition C_{new} is

$$C_{new} \leftarrow \arg\max_{C \in CS} (v(C \cup i) - v(C)). \quad (7)$$

  An agent leaves a coalition if the coalition is not the agent's current choice. This strategy maximizes marginal contribution of an agent to a coalition. In game theory literature this strategy is often referred to as the best response strategy.
- **Random strategy**, proposed in [7], makes decisions to leave and join coalitions randomly. Despite its trivial reasoning, this strategy can be used for a fast search of the state-space.
- **Mixed strategy** utilizes decision making of at least two strategies. The deciding strategy is chosen using a roulette wheel algorithm, which picks a strategy randomly based on given probabilities of the strategies.
- **Local search strategy** performs local optimization with random jumps when a local optimum is reached. This strategy combines the *coalition value-based* and *random* strategies as follows. The *coalition value-based* strategy is used by all agents as long as the resulting coalition structure continues to change. If an iteration yields the same coalition structure as the previous

iteration, the *random* strategy is used once by all agents in order to escape the local optimum.

4. Experimental Analysis

We evaluated our coalition selection strategies and valuation functions experimentally using the gain and gain ratio metrics (Equations 3 and 4). However, the gain ratio metric can only be applied to instances with small numbers of agents (here up to 20) because an optimal solution is used as a baseline. For the baseline, we used an optimized implementation of a dynamic programming algorithm[1] from [4].

We used the following parameter settings for the experiments. In order to achieve reasonable run-times of our algorithm in instances with various numbers of agents n, we used the following numbers of iterations N. We set the number of iterations to $N = 500$ for small-scale instances with $n \leq 20$, $N = 10$ for instances with $n \in \langle 20, 5000\rangle$ and $N = 3$ for instances with $n > 5000$. Because our algorithms are any-time, a solution can be returned at any point during the simulation.

For the energy purchasing scenario the interest vectors of length $k = 24$ stored real-world daily energy profiles of households in Portugal [8] (one value for each hour, T was therefore set to 24). The hourly values were averaged for each agent over all days in January 2014 into a single average January day.

For the resource sharing scenario we used an international trade dataset provided by the World Trade Organization [9]. The dataset stores import and export amounts in US dollars between 167 countries in 17 commodity types, therefore we set $k = 17$. The amount of each resource of each agent was computed as the difference between export and import amounts of the given country in the year 2014. Positive and negative values of the resulting resource amounts denote surplus and shortage respectively.

The parameter γ representing coalition size penalty was set to 1.1 following [3] for the energy purchasing scenario and to 2 for the resource sharing scenario. The higher value of γ was used in the resource sharing scenario to prevent the grand coalition, which is the trivial solution, from being the optimal solution. As suggested in [5] and [3], we fixed the prices for the energy purchasing scenario at $p_S = -80$ and $p_F = -70$. Negative values are used because the coalition value is maximized. Due to numeric limitations of the baseline dynamic programming implementation[2] we used randomly generated data for the small scale experiments. Specifically, elements of interest vectors were drawn from a uniform distribution $U\{0, 10\}$ for the collective energy purchasing scenario and $U\{-10, 10\}$ for the resource sharing scenario.

1. For the dynamic programming algorithm, evaluations of all possible coalitions were generated using the given valuation functions.

2. The baseline dynamic programming algorithm is implemented in C using integer and long types, which creates a risk of integer overflow for large coalition values.

We ran our Java implementation of the proposed algorithms on 2.7 GHz Intel Xeon E5 CPU with 2 GB of memory[3]. All results were generated by averaging over 10 random runs of our algorithms.

4.1. Small-scale problem instances

For problem instances containing up to 20 agents, we compared the performance of our coalition selection strategies with optimal solutions. Table 1 shows an average gain ratio achieved by the strategies. The table also shows results achieved by a state-of-the-art hierarchical agglomerative clustering algorithm C-Link [3].

The best average gain ratio was achieved by strategies that combine local search and random approaches, as shown in the first and third highest ranking strategies in Table 1. The highest ranking *mixed* strategy utilized random search more often than the *local search* strategy, it was therefore able to search larger portion of the search space and consequently find better solutions.

The locally optimizing *coalition value-based* strategy achieved a worse gain ratio because it cannot escape local optimums and therefore it wastes the remaining iterations after a local optimum is found. The *random* strategy, which performs an uninformed search of the state-space, ranked last.

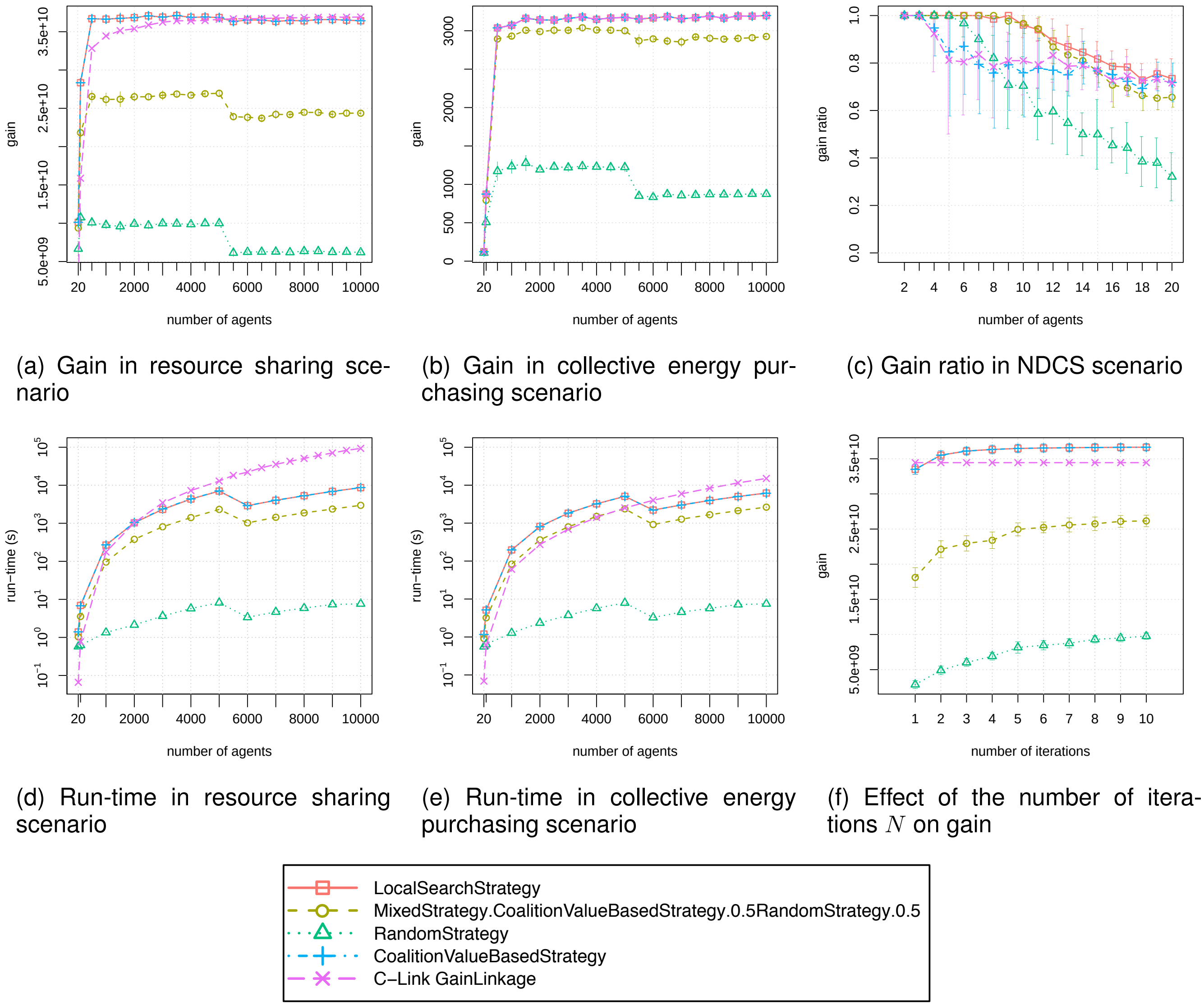
Figure 1c shows gain ratio of our strategies and C-Link in the challenging NDCS scenario. In this scenario the *local search* strategy ranks first, outperforming both the *mixed* strategy and C-Link, thus showing the usability of the *local search* strategy in challenging small-scale problem instances.

The results showed that the multi-agent simulation approach for coalition formation yields solutions with gain on average up to 94% of the gain of optimal solutions in problem instances where optimal solutions can be obtained using a state-of-the-art optimal algorithm. The results also showed that *local search* and *mixed* strategies find solutions of similar or higher quality than the state-of-the-art algorithm C-Link.

TABLE 1. AVERAGE GAIN RATIO IN THE SMALL-SCALE PROBLEM INSTANCES, with number of iterations $N = 500$. Combination of local and random searches yields results closest to the optimum. Results are averaged over all small-scale experiments with resource sharing, collective energy purchasing and NDCS scenarios.

Coalition selection strategy	$\mathbf{avg(gr(CS))}$
Mixed: coalition value-based, random	0.9399
C-link: Gain Linkage	0.9289
Local search	0.9203
Coalition value-based	0.8700
Random	0.8097

3. The 2 GB limit provides sufficient memory for our algorithms used in experiments with up to 10,000 agents.



(a) Gain in resource sharing scenario

(b) Gain in collective energy purchasing scenario

(c) Gain ratio in NDCS scenario

(d) Run-time in resource sharing scenario

(e) Run-time in collective energy purchasing scenario

(f) Effect of the number of iterations N on gain

Figure 1. 1a, 1b) Gain achieved by our algorithm and C-Link for given coalition selection strategies and valuation functions, for number of iterations $N = 10$ for $n \leq 5000$ and $N = 3$ for $n > 5000$. Gain reflects how much on average each agent benefits from coalition formation. We cannot compare gain values across scenarios because each valuation function yields a different magnitude of the gain. 1c) Gain ratio achieved by our algorithm and C-Link in the normally distributed coalition structures scenario with number of iterations $N = 500$: *Local search strategy* dominates the hierarchical clustering approach C-Link. 1d, 1e) Run-time of our algorithm and C-Link in seconds. Our algorithm is faster than C-Link in large instances. 1f) Effect of the number of iterations N on gain in the resource sharing scenario with 1000 agents. *Local search* and *coalition value based* strategies dominated C-Link after two iterations.
The mixed strategy is encoded as "MixedStrategy.StrategyA.probability-of-AStrategyB.probability-of-B." Error bars show standard deviation of aggregated variables.

4.2. Large-scale problem instances

Given the promising results of the small-scale experiments, we experimented with higher numbers of agents (up to 10,000) in order to measure the performance of our algorithms in a large-scale setting in which optimal algorithms cannot be applied. Figures 1a and 1b compare our strategies and C-Link based on the achieved gain. Note that the gain loss between iterations with number of agents $n = 5000$ and $n = 5500$ is caused by the decreased number of iterations N for $n > 5000$. We decreased the number of iterations N in order to achieve reasonable run-time of our algorithm. Further experiments showed that increasing the number of iterations yields results with both gain and run-time comparable to C-Link.

We highlight the following three observations. First, in collective energy purchasing scenario the gain achieved by *local search* and *coalition value-based* strategies is the same as the gain achieved by C-Link, and in instances of resource sharing scenario with $N = 10$ the gain of these two strategies is greater than the gain of C-Link. Decreasing number of iterations to $N = 3$ causes the gain of our strategies in resource sharing scenario to become slightly lower than the gain of C-Link. However, in these instances the run-time of our strategies is over one order of magnitude lower than the run-time of C-Link, as shown in Figure 1d.

Second, the gain achieved by *local search* and *coalition value-based* strategies for $n \geq 20$ is greater than the gain at $n = 20$, demonstrating that our algorithms provide stable average gain for the agents with increasing scale of the problem. Therefore agents in large-scale scenarios benefit from coalition formation more than agents in small-scale scenarios in which the solutions are very close to the optimum.

Third, the *local search* and the *coalition value-based* strategies outperformed other strategies in all scenarios because each agent in each iteration locally optimizes the overall gain. Although in small-scale problem instances this approach is outperformed by quicker strategies that randomly search larger part of the state-space within the given number of iterations, this advantage is no longer as important in large-scale problem instances because the state-space grows exponentially and therefore these quicker strategies can only search its fraction. Therefore slow, locally optimizing strategies yield solutions with higher gain.

Since the input to Algorithm 1 is the number of iterations N, it is important to study the effect of choosing N. Figure 1f shows the gain achieved by our strategies after $N = \langle 1; 10 \rangle$ iterations in the resource sharing scenario. The gain of the highest ranking *local search* and *coalition value based* strategies dominated the gain achieved by C-Link after two iterations, and it became stable around fifth iteration. This result shows that the number of iterations used in the resource sharing scenario can be very low. Similar results were achieved in the collective energy purchasing scenario.

Another factor in choosing the number of iterations is the convergence of our algorithm. Since the *local search*, *random*, and *mixed* strategies are random-based and therefore do not converge, we only study the convergence of the *coalition value-based* strategy. Figure 2 shows the number of iterations until convergence in resource sharing and collective energy purchasing scenarios. The results vary between the two scenarios, indicating that the resource sharing scenario is harder to solve. Interestingly, the number of iterations until convergence does not increase significantly with increasing number of agents in the collective energy purchasing scenario. Overall, the number of iterations needed to reach convergence is relatively small.

Finally, in practice the run-time of the algorithm can be a constraint, it is therefore also a deciding factor in choosing the number of iterations N. Figures 1d and 1e show run-time of our algorithm. Note again that the sudden decrease in run-time between $n = 5000$ and $n = 6000$ is caused by decrease in the number of iterations from $N = 10$ to $N = 3$. *Coalition value-based* and *local search* strategies are slower than the other strategies because they more often perform expensive search for a coalition maximizing the marginal contribution. However, in instances with $n > 5000$, in which $N = 3$, these strategies are faster than C-Link. Particularly in the resource sharing scenario with high numbers of agents our algorithm is faster by one order of magnitude. Together Figures 1f, 1d, and 1e show the trade-off between run-time and solution quality. Unlike C-Link, our approach allows for tuning of this trade-off using the number of iterations N.

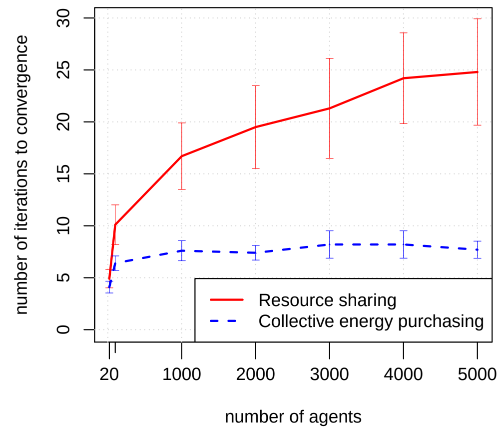


Figure 2. Number of iterations it takes the *coalition value-based* strategy to converge to a local optimum in both scenarios.

5. Discussion

We designed a framework that simulates coalition formation. While the simulator part of the framework is fixed, the valuation functions and agents' strategies are configurable, which gives a user extensive flexibility in modeling various coalition formation scenarios. We showed the usability of the framework by designing and implementing various types of agents' behavior along with specific applications of coalition formation. Our design approach can be followed to model other applications by designing and implementing new valuation functions. New agents' behavior can also be modeled with the use of new coalition selection strategies following our approach.

The framework can be used to simulate scenarios with thousands of agents. In these large-scale scenarios we do not guarantee finding an optimal solution. However, the framework can be used to generate a step-by-step evolution of the multi-agent system based on specific behavior of the agents, which better reflects the real world where global optimum is seldom an achievable goal. We show performance of the framework in simulations of the coalition formation applications. Using a combination of local and random searches, the framework is able to find solutions very close to the optimum in small-scale instances of the problem. With the scale of the problem increasing to 10,000 agents, the single agents achieve higher gain than in small-scale instances. For new user-defined coalition formation applications and agents' strategies it might not be obvious which strategy will yield the best solutions. Our framework can be used to test different strategies of the agents in given applications and determine the performance of the strategies, similarly as we did in Figure 1.

As mentioned in Section 3, the worst-case time complexity of our algorithm is $O(N \cdot n^2 \cdot k)$, which is lower than the $O(n^3)$ complexity of the C-Link algorithm [3]. An element that could affect the time complexity as well as the behavior of our algorithm is a cost of communication between agents. The communication cost would have to be included if the approach was distributed among multiple computational units. However, since Algorithm 1 is centralized, we do not include the communication cost in our calculations for the cause of simplicity and to be able to compare our results with results of C-Link, which also does not take communication cost into account.

In order to reuse our framework, several steps have to be taken and several design decisions must be made. First Algorithm 1 has to be implemented. Then a decision has to be made about strategies that the agents will use. For example, a self-interested agent might utilize a different strategy than a selfless agent. A valuation function has to be then designed to represent the specific problem application that is being examined. Finally, a decision has to be made about which CS should be selected as the solution, since one CS is created in every iteration. In our experiments we selected CS with the highest gain, but other options include selecting the last CS or CS containing coalition with the highest value.

6. Related Work

This section discusses related work in coalition formation, including multi-agent simulation used for coalition formation. Coalition formation is usually solved by one of the following approaches: dynamic programming (DP), graph-based algorithms, heuristic algorithms, or hierarchical clustering. The first two approaches are exact and guaranteed to find optimal solutions. Although the last two approaches do not provide guarantees on solution quality, they are able to solve large problem instances. Our approach of using multi-agent simulation can be classified as a heuristic algorithm because the simulation performs a greedy search in the state-space of CSs.

DP was initially used to solve the coalition formation problem with the first algorithm proposed by Yeh [10]. Rahwan and Jennings [1] proposed an algorithm called Improved Dynamic Programming (IDP), an improvement of standard DP that performs fewer operations and uses less memory. Recently, Cruz-Mencía et al. [4] proposed an optimized implementation of DP and IDP algorithms.

Graph-based algorithms utilize synergy graphs, which are graphs that encode agents' abilities to cooperate peer to peer. The DyCE algorithm [11] improves upon IDP by recognizing and ignoring infeasible coalitions using the synergy graph. Bistaffa et al. [12] proposed a branch and bound algorithm CFSS that searches the state-space by contracting edges of the synergy graph. This state-of-the-art algorithm is able to solve instances of the problem containing up to approximately 50 agents. Also, Sless et al. [13] have recently proposed a centralized graph-based algorithm in which a central organizer suggests new cooperation by adding edges to the graph.

An increasing number of agents causes the search for optimal CS to become infeasible. Suboptimal solutions can then be found using heuristic algorithms. Shehory and Kraus [14] first proposed a greedy algorithm that restricts the allowed size of coalitions to solve task allocation in a multi-agent system. Other approaches were later used to tackle coalition formation in larger scale setting, such as use of a genetic algorithm [15], simulated annealing [16], and greedy adaptive search [17]. Despite promising results, particularly for the greedy adaptive search, these algorithms focus on instances of the coalition formation problem that contain less than hundred agents.

Farinelli et al. recently proposed C-Link [3], which is a hierarchical clustering algorithm that addresses large-scale coalition formation. C-Link finds a suboptimal solution for 2732 agents in 4 minutes. Although C-Link addresses a similar problem to the problem we address, we focus on the simulation aspect by studying how strategies of single agents affect overall behavior of the system. Unlike C-Link, our framework models systems that change and adapt and it computes the evolution of the CS over time.

Multi-agent simulation studies coalition formation from several viewpoints. In [7], agents randomly choose coalitions in a coalition game in order to perform tasks. After each round, depending on their simple strategies, the agents can decide to leave the coalition or to stay. We take this simulation approach further by proposing more complex heuristic strategies and applications. An iterative approach for finding core-stable coalitions was proposed in [18]. While the approach in [18] is similar to ours, it can only be used in small scale scenarios due to its high complexity, as was shown in [19], where the algorithm from [18] was improved and empirically tested. In [20], game theoretical perspective is taken in which agents are defined by attraction for gain, stability, and strength of character. Even though the authors of [20] provide strong game theoretical background, they only experiment with four agents. A physics-motivated algorithm is proposed in [21] to solve the coalition formation problem for large-scale electronic markets. Coalition formation is solved in [21] using a macroscopic model, in which agents encounter coalitions randomly. The decisions about leaving coalitions are also made randomly based on some probability. Our framework does not use the macroscopic point of view and can therefore model behavior of single agents. Our agents also utilize more complex strategies. A recent approach has been proposed in [22] to dynamically assemble teams of workers to perform crowdsourcing tasks.

Coalition formation can also be solved using auctions, [23] presented an auction-based system for buyer coalition formation in large-scale e-markets.

To conclude the summary of related work in the field of coalition formation, we refer the reader to a comprehensive survey on CS generation in [24].

7. Conclusion

In this work we proposed a general framework for finding suboptimal solutions for a large-scale coalition formation problem containing thousands of agents using a multi-agent simulation. We modeled coalition formation as an iterative process in which agents leave and join coalitions based on the information from the current and previous iterations. We presented example applications of coalition formation: resource sharing and collective energy purchasing, along with valuation functions that model them by assigning values to the coalitions. We discussed coalition selection strategies that the agents can use in their decision making to leave and join coalitions. Finally, we analyzed our algorithms experimentally by comparing performances of the strategies in various problem settings using synthetic and real-world data.

We showed that our algorithms perform almost optimally in small-scale problem instances in which our best strategies performed similarly or better than the state-of-the-art algorithm for coalition formation C-Link. We also showed that the performance of our algorithms is stable in large-scale instances in which comparison with an optimal solution is infeasible. In these large-scale instances the quality of solutions found by our algorithm is greater or equal to the quality of solutions found by C-Link in majority of instances, and in remaining instances our algorithm yields run-time lower than run-time of C-Link by one order of magnitude, while still keeping solution quality similar to the quality of solutions found by C-Link.

We found that the best performance is achieved by strategies that combine local search with random jumps. Our strategies found solutions with values, on the average, up to 94% of the optimum in small-scale problem instances and maintained a steady gain per agent in large-scale problem instances.

In future work we plan to investigate the relationship between the local and random search. We will also shift our focus towards adaptive systems in which factors outside and inside the multi-agent system can change coalition values. This problem includes scenarios with self-interested agents in which each agent evaluates coalitions individually based on the agent's preferences. Finally, we will investigate how our algorithm can be designed in a distributed way, thus further increasing the scale of the coalition formation problems that our approach can solve.

References

[1] T. Rahwan and N. Jennings, "An improved dynamic programming algorithm for coalition structure generation," *7th International Conference on Autonomous Agents and Multiagent Systems*, 2008.

[2] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohmé, "Coalition structure generation with worst case guarantees," *Artificial Intelligence*, 1999.

[3] A. Farinelli, M. Bicego, S. Ramchurn, and M. Zucchelli, "C-link: A hierarchical clustering approach to large-scale near-optimal coalition formation," *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

[4] F. Cruz-Mencía, J. Cerquides, and A. Espinosa, "Optimizing performance for coalition structure generation problems' idp algorithm," *2013 International Conference on Parallel and Distributed Processing Techniques and Applications*, 2013.

[5] M. Vinyals, F. Bistaffa, A. Farinelli, and A. Rogers, "Coalitional energy purchasing in the smart grid," *2012 IEEE International Energy Conference and Exhibition, ENERGYCON 2012*, 2012.

[6] T. Rahwan, S. D. Ramchurn, and N. R. Jennings, "An anytime algorithm for optimal coalition structure generation," *Journal of Artificial Intelligence Research*, 2009.

[7] C. Merida-Campos and S. Willmott, "Modelling coalition formation over time for iterative coalition games," *Third International Joint Conference on Autonomous Agents and Multiagent Systems AAMAS 2004*, 2004.

[8] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014

[9] World Trade Organization. (n.d.) Available at http://stat.wto.org/StatisticalProgram/WSDBStatProgramSeries.aspx. Accessed: 2016-03-03.

[10] D. Yun Yeh, "A dynamic programming approach to the complete set partitioning problem," *BIT Numerical Mathematics*, 1986.

[11] T. Voice, S. D. Ramchurn, and N. R. Jennings, "On coalition formation with sparse synergies," *11th International Conference on Autonomous Agents and Multiagent Systems*, 2012.

[12] F. Bistaffa, A. Farinelli, J. Cerquides, J. Rodríguez-Aguilar, and S. D. Ramchurn, "Anytime coalition structure generation on synergy graphs," *13th International Conference on Autonomous Agents and Multiagent Systems*, 2014.

[13] L. Sless, N. Hazon, S. Kraus, and M. Wooldridge, "Forming coalitions and facilitating relationships for completing tasks in social networks," *13th International Conference on Autonomous Agents and Multiagent Systems*, 2014.

[14] O. Shehory and S. Kraus, "Methods for task allocation via agent coalition formation," *Elsevier Science*, 1998.

[15] S. Sen and P. Dutta, "Searching for optimal coalition structures," *Fourth International Conference on Multiagent Systems*, pp. 287–292, 2000.

[16] H. Keinänen, "Simulated annealing for multi-agent coalition formation," *Third KES International Symposium on Agent and Multi- Agent Systems: Technologies and Applications, KES-AMSTA '09, Springer, Berlin/Heidelberg*, pp. 30–39, 2009.

[17] N. Di Mauro, T. M. A. Basile, S. Ferilli, and F. Esposito, "Coalition structure generation with grasp," *14th International Conference on Artificial Intelligence: Methodology, Systems, and Applications, AIMSA'10, Springer, Berlin/Heidelberg*, pp. 111–120, 2010.

[18] T. Arnold and U. Schwalbe, "Dynamic coalition formation and the core," *Journal of Economic Behavior & Organization*, 2002.

[19] F. Bistaffa and A. Farinelli, "A fast approach to form core-stable coalitions based on a dynamic model," *2013 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2013*, 2013.

[20] S. Bonnevay, N. Kabachi, and M. Lamure, "Agent-based simulation of coalition formation in cooperative games," *IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, 2005.

[21] K. Lerman and O. Shehory, "Coalition formation for large-scale electronic markets," *Fourth International Conference on MultiAgent Systems*, 2000.

[22] Z. Pan, H. Yu, C. Miao, and C. Leung, "Efficient collaborative crowdsourcing," *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[23] J. Yamamoto and K. Sycara, "A stable and efficient buyer coalition formation scheme for e-marketplaces," *AGENTS '01 Proceedings of the fifth international conference on Autonomous agents*, 2001.

[24] T. Rahwan, T. P. Michalak, M. Wooldridge, and N. R. Jennings, "Coalition structure generation: A survey," *Artificial Intelligence*, 2015.