Universitat Politècnica de Catalunya
Universitat de Barcelona
Universitat Rovira i Virgili

Master in Artificial Intelligence

Introduction to Multi-Agent Systems

Analysis of Cooperation Mechanisms

Deliverable 2

by:
Sebastian Berns
Helen Byrne
Johannes Heidecke
Sara Hoeksma Palazuelos
Maritza Prieto

December 2016

Contents

1 Introduction **2**

1.1 Goal Definition . . . 2

1.2 Analysis of necessary cooperation . . . 3

2 Analysis of Cooperation Mechanisms **3**

2.1 PGP . . . 3

2.1.1 General Description . . . 3

2.1.2 Application to Practical Work . . . 3

2.1.3 Advantages and Disadvantages . . . 4

2.2 GPGP . . . 4

2.2.1 General Description . . . 4

2.2.2 Application to Practical Work . . . 5

2.3 Coalitions . . . 5

2.3.1 General Description . . . 5

2.3.2 Application to Practical Work . . . 5

2.3.3 Advantages and Disadvantages . . . 6

2.4 Contract Net . . . 6

2.4.1 General Description . . . 6

2.4.2 Application to Practical Work . . . 7

2.4.3 Advantages and Disadvantages . . . 8

2.5 Auctions . . . 8

2.5.1 General Description . . . 8

2.5.2 Application to Practical Work . . . 8

2.5.3 Advantages and Disadvantages . . . 8

2.6 Voting . . . 9

2.6.1 General Description . . . 9

2.6.2 Application to Practical Work . . . 9

2.6.3 Advantages and Disadvantages . . . 9

3 Chosen Cooperation Mechanisms **10**

3.1 Scouting Coordination . . . 10

3.2 Harvesting Coordination . . . 11

3.3 Vehicle Coordination 11

1 Introduction

In this report we analyze the cooperation mechanisms of the multi agent system (MAS) described in the previous report with the task of recycling the garbage in a virtual city. Section 1.1 defines the common goal and performance measure of the MAS on which the choice of different possible actions of the agents will be based. Section 1.2 gives an overview of the different coordination tasks in this specific problem.

Section 2 analyzes the different possible cooperation mechanisms that could be chosen to solve the problem: PGP/GPGP, Coalitions, Contract Net, Auctions and Voting. For all mechanisms a general description is given, followed by possible applications to the specific problem of garbage collection, and advantages and disadvantages.

In section 3, the chosen mechanisms are presented together with a justification of why they were chosen.

1.1 Goal Definition

All the agents in our MAS are contributing to the goal of recycling the garbage in the city in an efficient manner (except for the SystemAgent, which is solely taking care of the state of the virtual city and not further interacting with it). In order for several agents to contribute towards solving a distributed problem, they need to be coherent (working towards the same goal) and competent (knowledge how to work together). While we define the competence of the agents in the following sections by specifying their coordination mechanisms, we start with defining the common goal of the MAS, that ensures coherent behavior.

All agents should aim to maximize a common global performance measure. According to the project description, the MAS ought to maximize the benefits (points received by recycling centers) while minimizing the time to recycle garbage. The agents' decisions will be based on evaluating different possible plans for the next t time steps using these goal measures and often there will be a trade off between achieving maximum points or minimal waiting time.

For all possible plans of actions $p \in \mathbb{P}$ we want to maximize the benefits of the plan:

$$\max_{p \in \mathbb{P}} b(p)$$

with $b(p)$ being the sum of benefit points achieved with the plan p, while, at the same time, we want to minimize some measure of waiting time:

$$\min_{p \in \mathbb{P}} t(p)$$

with $t(p)$ being an evaluation of the waiting time corresponding to p. Looking at the specific domain of garbage recycling, it makes sense to punish long waiting times overproportionately. Hence we choose $t(p)$ to be the sum of squared waiting times of all garbages between being put on the map and being picked up by a HarvesterAgent.

Combining our two subgoals, we get the combined performance measure:

$$\max_{p \in \mathbb{P}} \theta_1 \cdot b(p) - \theta_2 \cdot t(p)$$

with θ_1 and θ_2 weighting the relative importance of benefits and weighting time. Since there are no further indications as to which of the subgoals should be considered more important, we set $\theta_1 = \theta_2$.

1.2 Analysis of necessary cooperation

The MAS is cooperative-benevolent, i.e. all agents work towards the same goal, which we defined in the previous subsection. The following coordination tasks are necessary to achieve the common goal:

- **Scouting Coordination**: The task of coordinating the Scout Agents (SA) in a way that maximizes the probability of detecting new garbage. SAs should obviously be well distributed over the map and primarily scout areas with high probabilities of detecting garbage (i.e. areas that haven't been scouted for a long time).
- **Harvesting Coordination**: The task of coordinating the Harvester Agents (HA) in a way to most efficiently recycle the detected garbage with respect to the performance measure defined in section 1.1.
- **Vehicle Coordination**: The task of combining the plans of scouting and harvesting, detecting collisions of vehicles, and resolving these collisions in a way that leads to the smallest loss of performance measure.

2 Analysis of Cooperation Mechanisms

2.1 PGP

2.1.1 General Description

1. Each agent creates a local plan: its own goals and most important actions to solve problem (abstract plan), and steps to achieve next step in abstract plan. This is updated as plan is executed.
2. Agents exchange local plans and generate PGP by combining local partial plans. Each agent knows organisation structure and knows which agents to send plans to.
3. Optimise PGP: analyse received information e.g. detect if several agents working on same activity. Alter local plans to better coordinate activities.

2.1.2 Application to Practical Work

Scouting Coordination: The ScoutAgents could be responsible for deciding their paths to discover garbage. They will put their proposed route information into their local plans and then send this onto the ScoutCoordinator. The ScoutCoordinator then builds a PGP with the local plans from all agents. The ScoutCoordinator detects if ScoutAgents plans are conflicting e.g. they plan to search the same area of map, they modify the PGP to avoid this and send this back to the scouts.

Harvesting Coordination: Similarly, when HarvesterAgents find out about garbage location they can generate their own plan (route) to harvest this located garbage. They add their suggested route to the garbage and onto appropriate recycling centre to their local plan, and send this onto the HarvesterCoordinator. The HarvesterCoordinator builds a PGP with the local plans from all HarvesterAgents, and decides which HarvesterAgent route is the most beneficial for collecting which garbage. The HarvesterCoordinator modifies the PGP with its decision for each garbage collection and sends this back to the HAs so they have their updated individual plans.

Vehicle Coordination: The ScoutCoordinator decides the routes that the ScoutAgents will take and this information goes into its local plan (it knows the current location and planned future location of all agents). Similarly, the HarvesterCoordinator is aware at all times of the location of all HarvesterAgents, which are idle and which are en route to harvest garbage. This information is put into their local plan. The ScoutCoordinator and HarvesterCoordinator send their local plans to the CoordinatorAgent. The CoordinatorAgent builds a PGP and is able to see the locations and planned routes of all SAs and HarvesterAgents. Using this information it can resolve conflicts e.g.two agents on one cell at one time, by deciding (based on a predetermined hierarchy) which agent must allow for another to pass by before continuing on its path. It will update the PGP with these decisions and send back to the ScoutCoordinator and HarvesterCoordinator to update the ScoutAgents and HarvesterAgents of any path modifications.

2.1.3 Advantages and Disadvantages

Advantages:

Flexible - so beneficial for changing environment i.e. new garbage added, plan can be updated to make optimal.

Efficiency - no two agents will work on same subproblem e.g. collecting same garbage

Disadvantages: Complexity

2.2 GPGP

2.2.1 General Description

GPGP is a domain-independent extension of PGP. Furthermore, GPGP mechanisms communicate scheduling for “particular tasks” [DL95] rather than a complete schedule as in PGP. In this way agents can detect and coordinate task interactions.

(i) Agents update non-local viewpoints by sending information about a particular task to all other agents that can also solve it.

(ii) Deals with ’Hard’ coordination relationships e.g. by committing prerequisites of certain tasks to be completed by a certain deadline and rescheduling agent actions.

(iii) Deals with 'Soft' coordination relationships e.g. where possible, committing for certain tasks to be completed as soon as possible if they benefit other tasks.

(iv) Avoiding redundancy: when it appears that more than one agent will be completing the same task, one agent is chosen either by random choice or by other means e.g. calculating the best solution. All other agents are updated with this information denoting another agent is completing the task.

2.2.2 Application to Practical Work

If two HarvesterAgents commit to collect the same garbage, this information will be shared in their non-local viewpoints. The HarvesterCoordinator would detect the redundancy in this task and will choose the agent that offers the best solution e.g. closest to garbage. This agent will be notified to collect the garbage and the other will be updated with instructions that another agent is completing the task.

2.3 Coalitions

2.3.1 General Description

Coalition formation involves several agents creating a possibly temporary group to help achieve tasks that could either not be achieved individually, or in order to complete the task more efficiently by working together.

2.3.2 Application to Practical Work

Coalitions are definitely applicable to our garbage collection problem as the MAS has agents that are cooperative, communicate, and deliberative.

The different agents in our garbage collection problem have different and complementary abilities to achieve the common goal of collecting garbage as efficiently as possible. We have five different agents, as well as clones of scouts and harvesters. A coalition would bring together different agents (not clones) to help achieve the common goal more efficiently.

Joint action is possible regardless of whether the agents are cooperative or selfish, as long as a coalition is beneficial for the agents involved [Rah+]. For instance, the scouting coordination and harvesting coordination are selfish in the sense that the agents involved act selfishly with regard to their tasks; acting in their own best interest towards their own goal. In our case, because of how the agents are structured, they contribute to the overall goal of the system (to collect and recycle garbage as efficiently as possible), despite their own 'selfish' goals.

"When the task is completed, the payoff is distributed, the coalition is disbanded and agents continue to pursue their own agendas" [SK]

Scouting Coordination: The ScoutAgents are told where to go by the ScoutCoordinator. They are purely reactive, as all they do is respond to find garbage and communicate this information. A coalition between the ScoutAgents only would not be useful as they do not have a common goal other than walking around as efficiently as possible. This efficiency has to be determined by the ScoutCoordinator.

Harvesting Coordination: As suggested in the slides, the HarvesterAgents could form a coalition given a list of detected garbage. This would include their current position, the position and quantity of garbage to be collected, the agents' remaining capacity and their state (whether idle or in the process of harvesting or recycling garbage). A coalition is dismantled once the task is completed.

Vehicle Coordination: The vehicle coordination could work using a coalition between ScoutAgents and HarvesterAgents. If these agents are working on fulfilling their goals and there is due to be a collision this conflict must be resolved before continuing. Forming a coalition could help avoid these conflicts, in order for each to complete their tasks more efficiently. This could work by agents forming a coalition when they are within a certain distance of one another. If an agent notes other agents within its personal radius, it can form a coalition to ensure that they efficiently get past each other without colliding and/or waiting longer than necessary. Additionally, the ScoutAgents and HarvesterAgents do not communicate with each other. However, it could be possible to allow these agents to communicate. This could be beneficial if a ScoutAgent and HarvesterAgent are nearby, and the ScoutAgent detects garbage, that this information is immediately passed to the nearby HarvesterAgent.

2.3.3 Advantages and Disadvantages

The evident advantage of coalitions is that they allow agents to work together to increase efficiency; theoretically they are only formed if a better outcome can be achieved by working together.

A core issue of working with coalitions is to identify which coalitions should be formed in order to achieve a goal. "This usually requires calculating a value for every possible coalition, known as the coalition value, which indicates how beneficial that coalition would be if it was formed." [Rah] It always requires a lot of communication between the agents who have to decide which coalitions to form, and they have to decide which mechanisms to use for the formation.

2.4 Contract Net

2.4.1 General Description

The Contract Net protocol was developed by R.G.Smith in 1980 [Smi80] and is based on the way in which companies put contracts out to tender. There are five phases the protocol runs through to resolve a certain task:

- **Recognition**: An agent (called the "manager") recognizes a task that has to be done and that it cannot solve or does not want to solve.
- **Announcement**: The manager announces the task to all agents (general broadcast), a group of agents (limited broadcast), or specific agents (point-to-point announce-

ment). The specification of the task includes a description of the task, constraints, and meta-information.

- **Bidding**: When an agent receives a task, they evaluate the task with respect to their specific properties and, if they are able and willing to solve the task, they submit a tender to the manager, indicating the relevant capabilities of the agent to solve the task.
- **Awarding**: A manager may receive several tenders for a task. Based on the information provided in the tenders, it chooses the best fitting agent or agents to execute the task. Chosen agents are called contractors for this task.
- **Expediting**: The chosen contractor or contractors then execute the task. If there is more than one contractor, further cooperation mechanisms might be necessary.

The Contract Net is based on a mutual selection process: Managers can select the best fitting tender and contractors can decide, for which announced tasks they want to create a tender.

2.4.2 Application to Practical Work

Harvesting Coordination: The coordination of harvesting garbage can be solved by using a Contract Net protocol. Newly detected occurrences of garbage are recognised by the HarvesterCoordinator and then announced to HarvesterAgents. This announcement can either be a limited broadcast to all HarvesterAgents or even a point-to-point announcement to specific HarvesterAgents of the right type that are currently idle. The HarvesterAgents can individually evaluate which of the announced tasks fit well enough to their current situation (based on garbage types, current location, free capacity, planned route, etc.) and then submit tenders for these tasks. The HarvesterCoordinator can then choose the most suitable candidate(s) to take care of the garbage. If several HarvesterAgents are selected to solve a single task (e.g. because their individual capacity does not allow one HarvesterAgent to take the entire garbage), they can use further coordination mechanisms to split up the task among them, or the HarvesterCoordinator assigns the exact distribution of splitting up the task. Since the bidding process is based on several attributes (distance, capacity, targeted recycling center, etc.), we are facing a multi-attribute negotiation.

Scouting Coordination: The ScoutCoordinator could use a Contract Net to announce the tasks of visiting cells that have been idle for a long time (and thus have a high probability of neighbouring a building with garbage). ScoutAgents can then submit tenders, e.g. containing how many steps they need to reach the idle cell and a numeric evaluation of the "idleness" of the path to this cell. This would be a multi-attribute-negotiation.

Vehicle Coordination: Potentially, when an upcoming collision of two vehicles is detected, the CoordinatorAgent can announce a task to the involved agents (or their coordinators) to resolve this collision in the best way they can come up with. The CoordinatorAgent then chooses the best plan of avoiding the collision, based on a metric such as the number of extra steps required.

2.4.3 Advantages and Disadvantages

The Contract Net is well applicable for this problem, because:

- The goal of collecting all garbage is dividable into several subgoals: one for each occurrence of garbage in the city
- The subtasks of collecting individual garbage are sufficiently complex and it is worthwhile to spend effort into distributing them efficiently.

The disadvantages are the computational effort due to the exchange of messages, and the time delay for deliberation and waiting for responses

2.5 Auctions

2.5.1 General Description

A class of negotiation protocols, which provide us with methods for allocating goods/resources, based upon competition among self-interested parties. There are different basic types of auctions: English, Dutch, FPSB-First price sealed bid and Vickrey. Auctions can furthermore be multi-unit, multi-attribute or combinatorial.

2.5.2 Application to Practical Work

Scouting Coordination: ScoutAgents could compete to get different or specialised routes. They would bid for their different routes and the ScoutCoordinator would choose who to assign which specific routes. In this way we would choose a FPSB auction in which all of the ScoutAgents bid for their route and the one with the least cost/best performance will be the winner for that route. The ScoutAgents bid without knowing the others' bids.

Harvesting Coordination: The HarvesterCoordinator could hold an auction for each garbage unit found to determine which HarvesterAgent(s) would harvest them. The HarvesterAgents' bids could depend on their current position, the position of the garbage and current state i.e. idle, moving to garbage for harvesting, moving to a recycling centre etc. Similarly, this could be a FPSB auction, where each HarvesterAgent would bid with the cost they would incur to reach the garbage without knowing others' bids and the HarvesterCoordinator would choose the best bid.

The auction could be a multi-unit auction, if several garbage units are located close to one another and they are auctioned together.
There could be a combinatorial auction of all garbage at a certain point in time.

2.5.3 Advantages and Disadvantages

The main advantage of an auction is that it creates competition; it enhances the seller's bargaining power, which means that the harvester competes for getting to collect garbage and the coordinator (seller) is able to choose the best bid.

Flexibility, as protocols can be tailor-made.

Advantages of multi-attribute auctions: They allow more degrees of freedom for bidders: price may not be the only attribute of interest and more efficient information exchange among the market participants.

Disadvantages: The disadvantages of auctions are mostly related to bidding problems such as lying, shills etc. These do not apply to our problem.

2.6 Voting

2.6.1 General Description

Voting is a distributed deliberation process. Decisions are taken collectively. There are two different basic voting protocols.

Simple voting is a group term for voting protocols, where each voter either chooses only their preferred option (Plurality), their least favourite option (Anti-plurality), both (Best-worst) or only the options they would accept (Approval).

In total order voting, each voter returns complete list of options sorted by their preference. There are different ways to count results and determine a winning option (Binary, Borda, Condorcet).

2.6.2 Application to Practical Work

Vehicle Coordination: Each vehicle (HarvesterAgent, ScoutAgent) puts a list of possible routes to their destination to a public vote. All other vehicles act as voters. Using the approval voting method the path with least conflicts can be determined. Each voter has to return a subset of paths that do not interfere with its own route. The option with the most votes wins.

2.6.3 Advantages and Disadvantages

In voting the decision process is decentralized. Applied to our problem, this means the coordinating agents (HarvesterCoordinator, ScoutCoordinator) merely act as administrative entities. If they have a vote at all, it only counts as much as any other agent's vote.

This would work well in a scenario, where agents have individual objectives and therefore need more freedom. The problem of harvesting and scouting coordination, though, require a centralized, hierarchic organisational structure. The use of a voting mechanism for both of these tasks, misses the actual purpose of that technique. Voting is used to collectively make a decision that affects all of the involved parties. As we want to split up tasks, though, there is no point in deciding on a plan for everyone together. There are more appropriate methods we can use.

Voting techniques, however, could well be applicable to the vehicle coordination task. Here the premise of individual objectives is given: each vehicle has its own destination. And the decision about which route to take potentially affects other agents.

Two problems, however, remain: Many vehicles have to determine their route at the same time. It is therefore not clear if a proposed path by a vehicle might be conflictive with that of another one, which has not yet determined its route. We need to define an order in which the vehicles determine their paths. This solution also has an unnecessary high computational cost, as we need to calculate alternative routes that will never be used.

3 Chosen Cooperation Mechanisms

3.1 Scouting Coordination

As part of the global goal definition of the MAS (see section 1.1), one important subtask is to efficiently detect new garbage on the map. New garbage should be quickly detected in order to allow its harvesting. As rational agents, ScoutAgents and the ScoutCoordinator want to maximize the expected performance of this task.

We have chosen to employ a combination of cooperation mechanisms in order to best fit the scouting coordination task. This task is a special case of the multi agent patrolling problem, a problem that has been thoroughly investigated by the research community. [Alm+04] have investigated a variety of different approaches for this problem and conclude that the best approach on most classes of map topologies is a TSP[1]-based Single Cycle Approach. A cyclical path based on TSP-algorithms is computed for all cells of the map and ScoutAgents patrol on this path in the same direction with more or less equal distances between them on the path.

A near optimal TSP path through the map will be calculated based on some common heuristic-based algorithm (calculating optimal solutions is unfeasible since the TSP problem belongs to the class of NP-complete problems).

In order to uphold this equidistance, e.g. after collisions, we combine this approach with GPGP. The ScoutAgents keep their local viewpoints updated with their current goals and actions - this will always be to follow their given path and to find garbage. If a situation arises in which a ScoutAgent has to change its path to avoid a collision, (this coordination is described in Vehicle Coordination Mechanism), then this change of path will be updated to the ScoutAgent's local viewpoint and then passed to the ScoutCoordinator who will keep a partial plan of all ScoutAgent local plans. The ScoutCoordinator will calculate how best to keep the ScoutAgents equidistant following this change of path, e.g. by keeping all ScoutAgents stationary until the collision-avoiding ScoutAgent is back to its original given path, and then it will pass these instructions back to the individual ScoutAgents. The ScoutAgents will now have updated local plans and will follow these new actions. In this way the ScoutAgents will continue, ensuring they are at equidistant points on their path to maintain our optimal map search.

[1] Traveling Salesman Problem

3.2 Harvesting Coordination

For the harvesting coordination we have chosen to implement three different cooperation mechanisms: voting for the garbage ordering, contract nets to assign the harvester to the garbage, and coalition for idle harvesters.

For the ordering of pending garbage (not yet picked up and unassigned), a voting mechanism will be implemented. The HarvestCoordinator will announce the pending garbage and each HarvesterAgent will respond with an ordered list of this garbage; ordered by garbage that they can pick up most efficiently. The HarvesterCoordinator chooses the garbage with the most top votes as the first garbage to be assigned (coordination described next) and once this is done, moves to the garbage with the second most votes, and so on. Every time new garbage is detected a new round of votes is cast.

In order to assign a HarvesterAgent to a specific garbage collection task, we have chosen to use contract nets. In the order determined by the voting, the HarvesterAgents bid if they are willing and able to pick-up the garbage in question. This bidding is independent from the previous voting for the garbage ordering. The HarvesterCoordinator then chooses the best fitting agent for this task and assigns it to the garbage. 'Best fitting' would be determined by the efficiency of the garbage collection, including distance to travel and type of garbage.

In order for the idle HarvesterAgents to be more purposeful, they will form a coalition with a ScoutAgent. As the idle HarvesterAgent has no task to complete in the current state, it will follow a ScoutAgent around in order that it is closer to potential newly-discovered garbage.

3.3 Vehicle Coordination

We will use GPGP for vehicle coordination. Each ScoutAgent and HarvesterAgent includes their current goals and actions in their local plan. These are passed to their respective coordinators and from there passed to the CoordinatorAgent who has a partial plan of all ScoutAgent and HarvesterAgent current paths/actions. If a situation arises in which the paths of 2 vehicles are going to collide in n steps, the CoordinatorAgent detects this and sets out to resolve it using the following solution:

We define a hierarchy of vehicle priorities, P, as follows:

1. Vehicle already moving to avoid collision
2. HarvesterAgent moving to recycling centre (to recycle garbage)
3. HarvesterAgent moving to garbage location (to harvest garbage)
4. ScoutAgent
5. HarvesterAgent idle

Avoiding collision solution:

`Max P (Vehicle 1, Vehicle 2) =>` Continue moving on current path.

`not Max P (Vehicle 1, Vehicle 2) =>` Go back on previous path taken until no longer in current path of prioritised vehicle. Remain stationary until prioritised vehicle is out of path. Move again on original path. (Note: whilst moving back on path and waiting stationary to avoid collision, this vehicle has higher priority than any other vehicle).

Using the above solution the CoordinatorAgent sends these updated paths for the vehicles involved to the ScoutCoordinator and HarvestCoordinator, who in turn pass these to each vehicle. Their paths are updated in their local plans to avoid the collision.

References

[Smi80] R. G. Smith. “The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver”. In: *IEEE Trans. Comput.* 29.12 (Dec. 1980), pp. 1104–1113. ISSN: 0018-9340. DOI: `10.1109/TC.1980.1675516`. URL: `http://dx.doi.org/10.1109/TC.1980.1675516`.

[DL95] Keith S. Decker and Victor R. Lesser. “Designing a Family of Coordination Algorithms”. In: *Proceedings of the First International Conference on Multi-Agent Systems.* CA: AAAI Press, 1995, pp. 73–80.

[BPR01] Fabio Bellifemine, Agostino Poggi, and Giovanni Rimassa. “Developing Multi-agent Systems with JADE”. In: *Intelligent Agents VII Agent Theories Architectures and Languages: 7th International Workshop, ATAL 2000 Boston, 2000 Proceedings.* Ed. by Cristiano Castelfranchi and Yves Lespérance. Springer Berlin Heidelberg, 2001, pp. 89–103.

[Alm+04] Alessandro Almeida et al. “Recent Advances on Multi-agent Patrolling”. In: *Advances in Artificial Intelligence – SBIA 2004: 17th Brazilian Symposium on Artificial Intelligence, Sao Luis, Maranhao, Brazil, September 29-Ocotber 1, 2004. Proceedings.* Ed. by Ana L. C. Bazzan and Sofiane Labidi. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 474–483.

[Woo09] Michael Wooldridge. *An Introduction to MultiAgent Systems.* Wiley, 2009.

[RN10] Stuart Russell and Peter Norvig. *Artificial Intelligence: International Version: A Modern Approach.* Pearson, 2010.

[Rah] T. Rahwan. “Algorithms for coalition formation in multi-agent systems”. In: ().

[Rah+] T. Rahwan et al. “Coalition structure generation: a survey”. In: ().

[SK] O. Shehory and S. Kraus. “Methods for task allocation via agent coalition formation”. In: ().