6

A Decentralised Swarm Approach for Mobile Robot-Systems

Oliver Kosak

Institute for Software and Systems Engineering,
University of Augsburg,
Augsburg, Germany,
`oliver.kosak@informatik.uni-augsburg.de`

Abstract. Mobile Swarm Robotics and its many challenges are highly worked on in many scientific research projects. The drawback of most existing approaches lies in the need of a central coordination unit for planning and controlling the swarms members. To create a real autonomous swarm, adaptive agents have to be introduced that reason about their current situation and constructively cooperate with each other. To port the needed algorithms to robotic systems, agents have to be empowered with appropriate sensors and actuators. Complications increase when focusing on aerial robots like multicopters. An overview of challenges and possible solution strategies is presented whose implementation leads to an autonomous swarm of multicopters.

6.1 Introduction

Swarm behaviour is one of nature's phenomena, everyone observing its occurrence can never forget about. Fish swarms with a large number of autonomous members attracted the attention of many researchers before. Especially *Swarm Intelligence*, e.g., the behaviour patterns each member of the swarm has to pursue so that the global visible effect can emerge, has been deeply investigated. At least since Reynolds introduced the core patterns necessary for swarming behaviour to the world of computer science and artificial intelligence [26], scientists are reasoning about the benefits swarming behaviour provides for autonomous computer systems. Many simulated approaches arose that port swarm behaviour patterns from nature to technical systems, but only few of them actually were brought to reality in terms of robotic scenarios - even less in the area of flying vehicles. This development experienced a twist with the latest improvement of the techniques necessary for controlling micro aerial robots like quad-, hexa- or octarotors (called *multicopter* generally in the following). Since controlling strategies that enable stable flight of multicopters achieved a level were no more human interaction is needed for stabilising the device when flying, the hardware got applicable for the next step: autonomous flights and coordinated flight in groups. The drawback of current approaches for swarm scenarios locates in the lack of their autonomy, where members of the swarm *do not* exclusively decide based on local knowledge (like sensor signals) or communication but are controlled in a *centralised* way. In the wide field of mobile robotics, real autonomous robot swarms where already deployed for ground-based tasks, like cooperative adoptions of formations [29] and more general scenarios like disaster rescue scenarios [23] using non-flying robots.

The task now is to adapt the knowledge originating from these swarm applications and then apply them to the field of flying robots. Autonomous swarms of multicopters can be used in many scenarios where they overcome the limitations of ground-based robot swarms. But not only benefits result from the uplift of a swarm to the air, lots of challenges have to be (re-)mastered when entering the new environment. These challenges range from well known problems like robust communication strategies to new ones, that originate from the fact of flying: necessary sensors might have to be distributed among multiple flying swarm members due to restricted payloads and limited energy supply. Especially, these tasks lead to the conclusion, that not only homogenous but heterogeneous swarms have to be focused on. Heterogeneity can also be given by merging ground and aerial based swarms into one, where the abilities of the members differ drastically. In the following, the goals to be achieved as well as the case study are presented in Section 6.2. An overview of the multitude of challenges that have to be mastered is presented in Section 6.3. Related projects and approaches are presented in Section 6.4.

6.2 Goals and Possible Case Studies

The utmost goal is to establish an autonomous heterogeneous swarm of mobile robots, that can be assigned with a diversity of tasks that are executed without any human interaction or guidance. Such a swarm should be adaptive and self-organising to provide a robust platform that can be trusted in by the user.

6.2.1 Properties of an Autonomous Swarm

One of the core properties of an autonomous swarm for avoiding bottlenecks and single points of failure is a decentralised organisation structure. Whereas centralised systems may possibly fail in situations where e.g., the communication connection to the central coordinator is lost, a decentralised system would still be able to autonomously work on its task. This leads to systems that are more resistant to internal and external failures, a very important fact in unknown environments. Adaptation as a property of the swarm is necessary to enable it to deal with changing environmental conditions, especially if the swarm is deployed in a previously unknown or only partially known area. The swarm has to be able to detect environmental changes and adapt to new situations in an adequate way. Additionally, the swarm should be able to distribute the given task among the swarm's members to guarantee its execution and fulfillment. This should also be ensured in case of the loss of a single member of the swarm (in the simplest case, the energy supply of a mulitcopter exceeds its limit and has to be recharged), e.g. up to a not yet specified degree redundancy is needed as property of the swarm. The swarm as whole has to guarantee the continuing execution of the task, maybe consuming more time than before but compensating for the loss. Additionally, a heterogeneous swarm can provide self-healing capabilities, like autonomous recharging of energy supplies re-enabling the functionality of a previously omitted device. With appropriate coordination algorithms the swarm has to configure itself according to its current situation, meaning that the swarm's members have to form relevant cooperation structures that are efficient in that situation (e.g., hierarchical structures, teams etc.).

6.2.2 Case Study

Given an autonomous swarm supporting all properties, as defined in Section 6.2.1, a multitude of tasks can be formulated for it in a wide variety of case studies.

One is the result of a cooperative partnership with geographical research departments. As multicopters are predestined for aerial measurements of a multitude of parameters (like temperature or humidity [8]) the potential of a swarm as autonomous instrument for measuring at high frequency and fine granularity in changing places is an opportunity for other sciences, especially geography. With the aerial mixture of different gases and particles being of high interest for determining and potentially improving the living conditions in urban places, appropriate measurement of indicating parameters is necessary [24]. At the current state of the art, techniques for aerial measurements either rely on stationary measurement (such as measurement towers [16]) or single flying vehicles. With an autonomous swarm for measuring, improvements of currently only approximated climate models are possible [8]. Additionally, an autonomous swarm can search for points of interest according to locally measured values via local sensors (such as Particle Swarm Optimisation in a real settings). Having found such spots, the swarm cooperatively decides to focus on further measurements by requesting swarm members with special abilities or sensors for investigation or by initiating formations relevant for appropriate measuring (e.g., by building up a virtual measurement tower). Autonomy is needed here because of the potential size and uncertain knowledge of the concerned area. As the execution of the given task is time critical, the swarm has to consist of many cooperating members. With increasing its size, the swarm's decentralisation gets more and more relevant. Only with distributing the computational load, the calculation of solutions can be handled in an appropriate time. With increasing complexity caused by the scale of the swarm and the multitude of retrieved sensor data, a centralised planner (that allocates tasks to the systems components) can not handle calculations anymore. Moreover, an autonomous swarm is needed to overcome probable technical difficulties like connection losses. As a swarm of multicopters is highly mobile, the only realistic communication infrastructure is via wireless technology. In wireless networks, packet losses and communication drops can happen as soon as receiving entities travel out of the sender's range. Only autonomous swarm members are able to choose the optimal solution by either reacting with a *return into range* strategy, or deciding to first complete their current task and rebuild the swarm's connection infrastructure afterward. Parts of the swarm can also form autonomous groups or teams that deliberately travel out of communication range to fulfill their tasks. A centralised planning communication entity is not able to handle all of these scenarios and only represents a bottleneck for efficient execution.

Furthermore, there are lots of other case studies (e.g., disaster scenarios, infrastructure examination, support in large scaled agricultural scenarios etc.) that all lead to the same or similar challenges that have to be overcome for establishing an autonomous swarm.

6.3 Challenges

To bring up an autonomous swarm to reality, supporting all properties defined in Section 6.2.1, that is able to be used in scenarios similar to the ones described in Section 6.2.2, many challenges have to be overcome. Few, but central ones, are shortly described in the following with ideas of and references to possible solutions.

6.3.1 Agent Architectures and Coordination Structures

Of great importance is the software architectures structure which is the base to support all features requested for an autonomous swarm. While the original definition of a swarm, where members only have very limited capabilities and solely act according to a set of very few rules [26], most likely does not cover all aspects necessary for complex scenarios, it brings up the idea of individually acting entities. This also is a central aspect of the highly worked on approaches of *Multi Agent Systems* (MAS), whose base is the decentralised interaction of autonomous agents that are adaptive and cooperative and rational [33]. To choose an appropriate agent architecture that is able to support all requested properties for ideally all possible case studies will be one of the primary tasks. This difficulty occurs at least because of the limitations introduced with porting the architectures to real hardware[1]. Many frameworks exist that describe agent architectures mostly based on or extending forms of BDI (*Belief-Desire-Intention*) [25] and support their implementation like Jadex [5] or SARL [28]. Additionally, architecture approaches featuring the MAPE-Cycle (*Monitor-Analyse-Plan-Execute*) [17] or using Observer/Controller patterns [27] have to be taken into account. Which of these can perform appropriately on multicopter hardware remains to be evaluated. In the following autonomous swarm and a MAS are of *equivalent meaning*, a swarm member is also called *agent*.

As one of the main features of a MAS is cooperation through communication, there is a need for appropriate structures. Which coalition types are needed is highly connected with the definition and decomposition of tasks as well as their suitability to be implemented on limited multicopter hardware. According to the structure of tasks, coalitions can be formed to execute tasks or to decompose tasks internally by creating

---

[1] We use the mobile performance of ODROID U3 (`http://www.hardkernel.com/main/main.php`) powered by a quad core processing unit with 2 GB of RAM available

new sub-tasks the coalitions members execute separately afterward. Formation can happen in various different ways, mainly divided up in two classes: on the one hand in *homogenous* groups of specialised agents that all support the same capabilities. Therefore, they can concentrate on a specific task, or, on the other hand, in *as heterogeneous as possible* groups of agents that are able to execute all tasks. Moreover, possible approaches can be distinguished regarding their assumption if one agent is allowed only to be member of one coalition or if overlapping coalitions are acceptable [30]. Coalitions can be formed dynamically as they are needed in the current situation, which requires algorithms that support online formation and deformation processes. In MAS, multiple approaches exist that enable its agents to form coalitions via voting mechanisms [15] or solving set partitioning problems [2]. The challenge here will be to determine appropriate algorithms, that also can be run on the single multicopter's limited hardware. An appropriate system structure might be hierarchical, where the main task can be introduced at the root of the hierarchy and then be decomposed for the next layer, ideally resulting in sub-tasks that can be executed by the subsystems (following the approach of system of systems [14]).

6.3.2 Task Decomposition and Task Allocation

In most use cases, a user wants a specific task to be executed by the autonomous swarm without micromanaging each agent - the swarm as a whole should be told what the total goal or task is (like: *get an overview map of area x and indicate points of interest*). To achieve this goal, many sub-goals and sub-tasks have to be formulated and accomplished by the MAS automatically. The user-specified goal has to be decomposed in smaller tasks that can either be worked on by single agents or by subsets of the swarm, e.g., by groups or teams. While some tasks can be decomposed straight forward (for example video scanning a huge area might be easily split up in two smaller areas to be scanned in two tasks by two agents or teams of agents), other tasks will be more complicated to be split up. With increasing the diversity of the swarm's agents and a higher abstraction level of task formulation task decomposition gets more complicated. With agents not supporting the same functionality, tasks and resulting sub-tasks only can be processed by specialised agents that are equipped with appropriate sensors or actors. Finding algorithms for automatic or at least semi-automatic decomposition (via predefined procedures) will be one of the most challenging steps to be mastered. Some approaches that lead in the same direction are presented e.g., as Hierarchical Task Networks (HTN) [12] where a formalism for setting up a HTN is given. In addition to the concrete scenario, specific tasks that will have to be modified each time the scenario changes. More common tasks (like strategies to enable at least *n* agents being in the air at the same time - not charging - or exploring a given area with a given set of sensors) can be decomposed in a similar way each time it is necessary, so there will algorithms be found to automatically achieve the task decomposition. Again, a challenge is located in finding suitably performing algorithms for the limited multicopter hardware.

When tasks are decomposed and algorithms are implemented that support appropriate coalition formation, the tasks have to be distributed on the available coalitions.

Achieving such task allocations is an often discussed subject in the area of technical systems, formulated as *Task Allocation Problems*. To allocate tasks to agents or groups of agents, many approaches exist for MAS as described in [30]. Solutions divide in centralised and decentralised, often market based [32], approaches following accurately defined protocols such as the *Contract Net Protocol* [31]. For an autonomous swarm, an appropriate algorithm for *decentralised* solving of the task allocation problem with adequate performance on the used hardware has to be found.

6.3.3 Hardware Challenges

In addition to the software specific challenges some significant hardware challenges have to be taken, too. Because of the overall complexity, the few aspects formulated in the following can only be seen as an excerpt for the hurdles that will occur during the realisation.

The most critical aspect here is the need of total guarantee that obstacles can be detected and collisions are avoided by each agent. On the one hand this is relevant for the swarm internally, as agents should never bump each other (which is highly relevant when considering flying vehicles as a crash may result in a direct loss of the concerned agents) and on the other hand collision avoidance with external elements, e.g., objects placed in the environment. This is relevant for the swarm's operator as well as all objects and persons in the swarm's insertion area to guarantee an adequate level of safety. As mentioned, all agents of the swarm have to be aware of the relative position of other agents and objects near their current location to avoid collisions. This can only be handled autonomously by integrating sensors that have the ability to scan this local environment with appropriate frequencies. Respecting the fact that not every agent might be able to carry every sensor needed, this might be one good reason to form coalitions where sensor carrying agents can communicate their measured values to others that lack this functionality. Suitable sensors can be cameras whose data streams may be analysed with object recognition algorithms, infrared cameras that sense temperature gradients, ultra sonic sensors, or laser scanners. Essential for mostly every action of an autonomous swarm is a robust communication infrastructure. Due to its mobility the swarm can only rely on wireless communication techniques. This leads to multiple specific obstacles as communication over a wireless channel is exposed to many kinds of disturbance. One obstacle to overcome is given by the restricted range of wireless transmitters. Reliable communication can only be kept up to a given (sometimes according to environmental conditions *dynamic*) distance. Here techniques like ad-hoc *Peer-to-Peer Networks* [7] have to be used to route messages correctly and robustly.

6.4 Related Work and Related Projects

Mobile Robotics and Swarm Robotics are research areas, lots of other scientists are interested in. Many approaches already exist that try to find solutions for coordinating multicopters as single units or as swarming units. The approach described in

[19] shows the design of micro quadrotors whose hardware makes them extremely flexible and its implemented software enables stable swarm-like flight consisting of central coordinated formations. In [21] the authors also show how multicopters can cooperate to grasp and carry payloads with a centralised control of the group of involved multicopters. The approach to coordinate a swarm of autonomous agents into a predefined density, only relying on independent probabilistic decisions without communication presented in [1], only lacks in not being hardware ported, but only exists as a simulated algorithmic approach. Nonetheless, ideas from this approach could be used for coordination algorithms with physical agents, like multicopters. Contrary to this simulated approach, in [13] Gutiérrez et al. present an approach with physical ground-based agents, whose swarm members can align only by evaluating communication streams with neighbor agents through analysing the direction of incoming infrared signals. Another approach of coordinating groups of robot agents is presented in [6] where different decentralised strategies for cooperatively moving obstacles are compared and evaluated. The approach in [10] gives a concrete implementation of a visual-SLAM (*Simultaneous Localisation and Mapping*) algorithm for a flying vehicle only relying on onboard vision to build up a map of a GPS-denied area with the goal of an optimal distribution of a team of multicopters to monitor this area. Another implementation of SLAM is presented in [18] where ground robots are enabled to create ad-hoc maps with initial aerial images to improve the original algorithm. Another approach of cooperating agents is given in [22] where a team consisting of a flying and a ground-based vehicle is able to determine the shortest route to a destination and successfully deliver an object to this position. Approaches presented in [9], [4] and [3] describe programming languages for the coordination of a large group of autonomous entities like they occur in a swarm (sometimes called *Amorphous Medium*). The drawback here is that none of them seem to be realised in a physical scenario and all of these have only been evaluated in simulation. Nonetheless the central ideas of the approaches should be investigated when building an autonomous swarm of multicopters. In [11] an approach of a heterogeneous swarm consisting of ground-based and aerial robots is presented whose members are able to cooperate for achieving goals that are only reachable by combining the heterogeneous properties of the swarm's members. Many interesting ideas can be found in this case study and the underlying techniques, for example ad-hoc communication infrastructures that can be set up to introduce stable communication in an autonomous swarm.

6.5 Working base for future approaches

At the current state, the swarm consists of eleven multicopters (ten quadcopters *Saphira* and one octocopter *Tamara*[2]). As an excerpt of a wider bunch of available multicopters, these models offer the most flexible interface for modifications. All multicopters have implemented the same *open source* flight controller (AutoQuad[3]),

---

[2] `http://www.rosewhite.de/`

[3] `http://autoquad.org/`

whose operating firmware can be modified in every necessary way. In the currently delivered stable firmware, GPS sensors are integrated to control outdoor positioning in combination with barometric sensors and accelerometers. For stabilising attitude of the multicopters a combination of PID controllers is used that guarantee a very stable flight. With an integrated extension board it is possible to extend the hardware with more sensors and other infrastructure, which we used to integrate another powerful processing unit (ODROID U3[4]). With its quad core arm A9 processor and 2 GB of RAM it is the base of all further algorithmic implementations to enable agent autonomy. Via a serial connection link to the flight controller it is possible to send steering instructions on the one hand and retrieve sensor information from the flight controller on the other hand, using the supported MAVLINK protocol[5]. Steering commands can be computed in each beneficial programming language as the processing unit supports a wide variety with its Ubuntu operating system. Especially the addition of more sensors gets a lot more practical as they can be integrated via CAN, SPI, UART or USB. For testing the stages of the swarm's autonomy we built up a flying arena inspired [20] that guarantees a safe environment for testing autonomous flights. The arena has a length of 9 meters, a height of 4 meters and a width of 5 meters, the sides are secured by nets, the floor by crash safe material. For indoor position control we use the motion capturing system by VICON[6] enabling high speed position tracing. This testing ground enables us to safely evaluate future autonomous flying approaches without heavy environmental disturbances. One current goal is the integration of an adaptive mechanism into the flight controllers firmware, that is able to change from a GPS position controller to a VICON tracking based position controller online according to the precision the two possible systems can offer (adaptive switching to the more precise one).

6.6 Conclusion

The research field of mobile robots combined with autonomous swarm techniques includes many challenges in a diversity of scientific areas. These range from algorithms for local orientation up to coordination and planing algorithms for autonomous agents. This paper shows an excerpt of some of the most important aspects that have to be taken into account when planing to establish an heterogeneous and autonomous swarm of multicopters that can be used in a variety of different scenarios for supporting a human user. The main difficulty is given in the process of decomposing the user's goal into single tasks for swarm members. Additionally, for realising and using an autonomous swarm, many more hurdles have to be taken. The route to an autonomous swarm may be challenging, but with many related approaches and already evaluated strategies for similar problems, that can be found in existing literature, the single steps seem to be manageable.

---

[4] `http://www.hardkernel.com/main/main.php`

[5] `http://qgroundcontrol.org/mavlink/start`

[6] `http://www.vicon.com/`

References

[1] Acikmese, B., Bayard, D.S.: A markov chain approach to probabilistic swarm guidance (2012)

[2] Anders, G., Siefert, F., Steghöfer, J.P., Reif, W.: A decentralized multi-agent algorithm for the set partitioning problem. In: PRIMA 2012: Principles and Practice of Multi-Agent Systems, pp. 107–121. Springer (2012)

[3] Ashley-Rollman, M.P., Goldstein, S.C., Lee, P., Mowry, T.C., Pillai, P.: Meld: A declarative approach to programming ensembles. In: Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ Int. Conf. on. pp. 2794–2800. IEEE (2007)

[4] Bachrach, J., McLurkin, J., Grue, A.: Protoswarm: a language for programming multi-robot systems using the amorphous medium abstraction. In: Proc. of the 7th Int. joint Conf. on Autonomous agents and MAS-Volume 3. pp. 1175–1178. Int. Found. for Autonomous Agents and MAS (2008)

[5] Braubach, L., Lamersdorf, W., Pokahr, A.: Jadex: Implementing a bdi-infrastructure for jade agents (2003)

[6] Campo, A., Nouyan, S., Birattari, M., Groß, R., Dorigo, M.: Negotiation of goal direction for cooperative transport. In: Ant Colony Optimization and Swarm Intelligence, pp. 191–202. Springer (2006)

[7] Chen, W., Cai, S.: Ad hoc peer-to-peer network architecture for vehicle safety communications. Communications Magazine, IEEE 43(4), 100–107 (2005)

[8] Corrigan, C., Roberts, G., Ramana, M., Kim, D., Ramanathan, V.: Capturing vertical profiles of aerosols and black carbon over the indian ocean using autonomous unmanned aerial vehicles. Atmospheric Chemistry and Physics 8(3), 737–747 (2008)

[9] De Nicola, R., Loreti, M., Pugliese, R., Tiezzi, F.: Scel: a language for autonomic computing. Univ. Firenze, Tech. Rep (2013)

[10] Doitsidis, L., Weiss, S., Renzaglia, A., Achtelik, M.W., Kosmatopoulos, E., Siegwart, R., Scaramuzza, D.: Optimal surveillance coverage for teams of micro aerial vehicles in gps-denied environments using onboard vision. Autonomous Robots 33(1-2), 173–188 (2012)

[11] Dorigo, M., Floreano, D., Gambardella, L.M., Mondada, F., Nolfi, S., Baaboura, T., Birattari, M., Bonani, M., Brambilla, M., Brutschy, A., et al.: Swarmanoid: a novel concept for the study of heterogeneous robotic swarms. Robotics & Automation Magazine, IEEE 20(4), 60–71 (2013)

[12] Erol, K., Hendler, J.A., Nau, D.S.: Umcp: A sound and complete procedure for hierarchical task-network planning. In: AIPS. vol. 94, pp. 249–254 (1994)

[13] Gutiérrez, Á., Campo, A., Dorigo, M., Donate, J., Monasterio-Huelin, F., Magdalena, L.: Open e-puck range & bearing miniaturized board for local communication in swarm robotics (2009)

[14] Horling, B., Lesser, V.: A survey of multi-agent organizational paradigms. The Knowledge Engineering Review 19(04), 281–316 (2004)

[15] Jackson, M.O., Moselle, B.: Coalition and party formation in a legislative voting game. Journal of Economic Theory 103(1), 49–87 (2002)

[16] Kaimal, J.C., Finnigan, J.J.: Atmospheric boundary layer flows: their structure and measurement (1994)

[17] Kephart, J.O., Chess, D.M.: The vision of autonomic computing. Computer 36(1), 41–50 (2003)

[18] Kümmerle, R., Steder, B., Dornhege, C., Kleiner, A., Grisetti, G., Burgard, W.: Large scale graph-based slam using aerial images as prior information. Autonomous Robots 30(1), 25–39 (2011)

[19] Kushleyev, A., Mellinger, D., Powers, C., Kumar, V.: Towards a swarm of agile micro quadrotors. Autonomous Robots 35(4), 287–300 (2013)

[20] Lupashin, S., Hehn, M., Mueller, M.W., Schoellig, A.P., Sherback, M., D'Andrea, R.: A platform for aerial robotics research and demonstration: The flying machine arena. Mechatronics 24(1), 41–54 (2014)

[21] Mellinger, D., Shomin, M., Michael, N., Kumar, V.: Cooperative grasping and transport using multiple quadrotors. In: Distributed autonomous robotic systems, pp. 545–558. Springer (2013)

[22] Mueggler, E., Faessler, M., Fontana, F., Scaramuzza, D.: Aerial-guided navigation of a ground robot among movable obstacles. In: Safety, Security, and Rescue Robotics (SSRR), 2014 IEEE Int. Symposium on. pp. 1–8. IEEE (2014)

[23] O'Grady, R., Pinciroli, C., Groß, R., Christensen, A.L., Mondada, F., Bonani, M., Dorigo, M.: Swarm-bots to the rescue. In: Advances in Artificial Life. Darwin Meets von Neumann, pp. 165–172. Springer (2011)

[24] Pöschl, U.: Atmospheric aerosols: composition, transformation, climate and health effects. Angewandte Chemie Int. Edition 44(46), 7520–7540 (2005)

[25] Rao, A.S., Georgeff, M.P., et al.: Bdi agents: From theory to practice. In: ICMAS. vol. 95, pp. 312–319 (1995)

[26] Reynolds, C.W.: Flocks, herds and schools: A distributed behavioral model. ACM Siggraph Computer Graphics 21(4), 25–34 (1987)

[27] Richter, U., Mnif, M., Branke, J., Müller-Schloer, C., Schmeck, H.: Towards a generic observer/controller architecture for organic computing. GI Jahrestagung (1) 93, 112–119 (2006)

[28] Rodriguez, S., Gaud, N., Galland, S.: Sarl: a general-purpose agent-oriented programming language. In: Web Intelligence (WI) and Intelligent Agent Tech. (IAT), 2014 IEEE/WIC/ACM Int. Joint Conf. on. vol. 3, pp. 103–110. IEEE (2014)

[29] Rubenstein, M., Cornejo, A., Nagpal, R.: Programmable self-assembly in a thousand-robot swarm. Science 345(6198), 795–799 (2014)

[30] Shehory, O., Kraus, S.: Methods for task allocation via agent coalition formation. Artificial Intelligence 101(1), 165–200 (1998)

[31] Smith, R.: The contract net protocol: Highlevel communication and control in a distributed problem solver, 1980. IEEE Trans. on Computers, C 29, 12

[32] Walsh, W.E., Wellman, M.P.: A market protocol for decentralized task allocation. In: Multi Agent Systems, 1998. Proc. Int. Conf. on. pp. 325–332. IEEE (1998)

[33] Wooldridge, M., Jennings, N.R.: Intelligent agents: Theory and practice. The knowledge engineering review 10(02), 115–152 (1995)