An Anytime Heuristic Algorithm for Allocating Many Teams to Many Tasks*

Extended Abstract

Athina Georgara
IIIA-CSIC & Enzyme Advising Group
Barcelona, Spain
ageorg@iiia.csic.es

Juan A. Rodríguez-Aguilar
IIIA-CSIC
Barcelona, Spain
jar@iiia.csic.es

Carles Sierra
IIIA-CSIC
Barcelona, Spain
sierra@iiia.csic.es

Ornella Mich
Fondazione Bruno Kessler
Trento, Italy
mich@fbk.eu

Raman Kazhamiakin
Fondazione Bruno Kessler
Trento, Italy
raman@fbk.eu

Alessio Palmero Aprosio
Fondazione Bruno Kessler
Trento, Italy
aprosio@fbk.eu

Jean-Christophe Pazzaglia
SAP
Mougins, France
jean-christophe.pazzaglia@sap.com

ABSTRACT

In many practical applications, we often need to form a team of agents to solve a task, since no agent alone has the full set of required competences or the power to complete the task on time. Here, we address the problem of distributing individuals in teams, with each team being in charge of a specific task. In particular, we formalise the problem and propose a heuristic approach to solve it.

KEYWORDS

Team Formation, Task allocation, Optimisation, Heuristics

ACM Reference Format:
Athina Georgara, Juan A. Rodríguez-Aguilar, Carles Sierra, Ornella Mich, Raman Kazhamiakin, Alessio Palmero Approsio, and Jean-Christophe Pazzaglia. 2022. An Anytime Heuristic Algorithm for Allocating Many Teams to Many Tasks: Extended Abstract. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Online, May 9–13, 2022*, IFAAMAS, 3 pages.

1 INTRODUCTION

Many real-world problems require allocating teams of individuals to tasks. For instance, forming teams of robots for search and rescue missions [4], teams of Unmanned Aerial Vehicles (UAVs) for surveillance [9], or grouping students to undertake school projects [3]. Considering the domain of education and motivated by the hard and time-consuming procedure of allocating student teams to school projects or internship programs, we focus on this specific instance of team formation. Currently, teachers and education authorities obtain such allocations mainly by hand, but given the combinatorial nature of the problem, (good) manual allocation requires a large amount of work. That is, manual allocation will very likely not find a good solution given the size of the problem. In this extended abstract, we study the allocation of many teams to many tasks with size constraints, disallowing overlaps. That is, each agent can be part of at most one team, each team can be allocated to at most one task, and each task must be solved by at most one team. Then, we introduce an anytime heuristic approach to match teams to tasks.

*Research supported by projects AI4EU (H2020-825619), TAILOR (H2020-952215), 2019DI17, Crowd4SDG (H2020-872944), CI-SUSTAIN (PID2019-104156GB-I00).

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Online.

2 RELATED WORK

Team formation has received much attention by the AI and MAS communities. Anagnostopoulos et al. in [1] thoroughly study the problem of forming a single team to resolve a single task, and show the employability of several algorithms in large scale communities. Lappas et al. in [8] tackle the problem of finding a single team of experts for a given task in an attempt to minimise the communication cost among the members of the team. Chad et al. [6] add a new dimension to the problem by considering robustness, and focus on finding a single robust team to perform several tasks. Andrejczuck et al. [3] tackle the many teams to single task problem and present algorithms for partitioning a set of agents into equal-size teams in order to resolve the very same task. Capezzuto et al. in [4] tackle the many teams to many tasks team formation problem considering temporal and spacial constraints, and propose an anytime, efficient algorithm. However, compared to the problem we tackle in this paper, the proposed algorithm in [4] provides solutions with overlapping teams.

Regarding the many teams to many tasks team formation problem with no overlaps—i.e., problems for which different teams share no common agents, each team can be allocated to only one task, and each task can be assigned to only one team—we can find a handful of works in the literature. Specifically, we have singled out two works, namely [7, 10], which can be considered as the one most directly related to ours. Although these papers tackle the general *many teams to many tasks* problem, their version of the problem is essentially different to ours, hence preventing us from conducting meaningful comparisons. In more detail, [10] propose a branch-and-bound technique to determine the optimal team size structure and then they proceed with a brute-force search. On the other hand, Czatnecki and Dutta [7] propose an algorithm for matching non-overlapping teams of robots to tasks. Similarly to [10], [7] sets no constraints on team sizes. However, even if we could 'bypass' the team size misalignment (by allowing [7] to yield a result, and use these team sizes in our version), there is yet another essential difference between [7] and our approach. Our algorithm pursues to *optimise* the competence affinity between all teams with their assigned tasks while targeting at balanced allocations (i.e., all teams are more or less equally competent for their task). Instead, [7] targets at finding stable teams such that the agents have no incentive to unilaterally abandon their current team and task. As such, [7] and our approach differ notably in their objectives.

3 THE MANY TEAMS TO MANY TASKS ALLOCATION PROBLEM

In this section we formally describe the *many teams to many tasks* allocation problem with no overlaps. To begin with, a *competence* corresponds to a specified capability, skill, or knowledge. We assume there is a known, predefined and fixed set of competencies, denoted by C. A *task* is characterised by a set of requirements on agents' competencies and team size constraints; formally, it is a tuple $\langle \text{t\_id}, C, w, \text{s}\rangle$, where t_id is a unique task identifier, $C \subseteq C$ is the set of required competencies, $w : C \longrightarrow (0, 1]$ is a function that weighs the importance of each competence, and $\text{s} \in \mathbb{N}_+$ is the required team size. An agent a is a tuple $\langle \text{a\_id}, C'\rangle$, where a_id is a unique agent identifier, and $C' \subseteq C$ is a set of competencies. We denote with T and A the sets of all tasks and all agents, respectively. Given $\tau \in T$, we denote the set of all size-compliant teams for τ as $\mathcal{K}_\tau = \{K \subseteq A : |K| = \text{s}_\tau\}$[1], where s_τ is the team size required by τ.

3.1 Matching Quality

The quality of a size-compliant team when solving a task depends on the competencies required by the task and those collectively offered by the team. Given a task and a team, we first shall assign 'responsibilities' to all team members, i.e., which required competencies shall each agent cover/fulfil. As soon as each agent is assigned with responsibilities, we can determine the quality of each agent against the task, by comparing their assigned competencies (as responsibilities) with their own ones. Thus, the quality of an agent $a \in K$ for a task τ is given by $\prod_{c \in \eta_{\tau \to K}(a)} \text{ql}(c, C_a)$, where $\eta_{\tau \to K} : K \longrightarrow 2^{C\tau}$ is a competence assignment function, and $\text{ql} : C \times 2^C \longrightarrow [0, 1]$ is an evaluation function that measures a quality degree for covering some competence $c \in C_\tau$ if the agent has competencies C_a. Note that given a task and a size-compliant team, assigning responsibilities to agents in a way that maximises the overall team's quality is an optimisation problem itself. The quality of a team K for tackling task τ is given by aggregating the quality of each $a \in K$ for its given responsibilities, i.e., $\prod_{a \in K} \prod_{c \in \eta_{\tau \to K}(a)} \text{ql}(c, C_a)$.

[1]Note: we use subscript a to refer to the set of competences and the identifier of an agent $a \in A$, and subscript τ to refer to the elements of task $\tau \in T$.

3.2 The Optimisation Problem

Finding a good allocation of agents to a collection of tasks is yet another optimisation problem whose aim is to maximise the *overall* matching quality of all teams for their assigned tasks. Specifically, we need to match each task $\tau \in T$ to *at most one* team, so that the matching quality of all pairs team-task is similarly high, i.e., the optimum allocation should promote balanced teams instead of a few high-quality teams together with poor-quality teams [2, 3]. Therefore, we use the product among the matching qualities of all team-task pairs [5]. Formally, let $g : T \longrightarrow 2^A$ be an allocation function that maps each $\tau \in T$ to a size-compliant team $K \subset \mathcal{K}_\tau$, so that each agent belongs to *at most one* team. That is, $g(\tau) \cap g(\tau') = \emptyset$ for all $\tau \neq \tau'$. Then, the optimal allocation function is:

$$g^* = \arg\max_g \prod_{\tau \in T} \prod_{a \in g(\tau)} \prod_{c \in \eta^*_{\tau \to g(\tau)}(a)} \text{ql}(c, C_a) \qquad (1)$$

Notice that in each allocation g, for each pair $g(\tau) - \tau$ we consider the optimal competence assignment function $\eta^*_{\tau \to g(\tau)}$, Thus, for each team allocation function we need to solve $|T|$ optimisation problems (one per task) in order to determine $\eta^*_{\tau \to g(\tau)}$.[2]

4 A HEURISTIC APPROACH

Here we sketch a heuristic approach that allows us to obtain many team to many tasks allocations in acceptable time. Our approach comprises a two-step greedy search.

First Step: The first step finds an initial allocation considering a competence analysis on the agents and the tasks at hand. That is, for each task we determine whether it is hard to find a high-quality team based on the task's requirements and the agents' competencies. Then, we greedily select a team for each task, considering the hardest tasks first.

Second Step: As soon as our algorithm finds an initial allocation, we improve the allocation by interchanging agents between pairs of teams following a greedy procedure. That is, for two randomly selected tasks we *optimally* re-allocate the agents in their teams in order to obtain the highest quality. This step is performed as many times as time allows, always having a feasible allocation (anytime algorithm).

5 CONCLUSIONS & ONGOING WORK

In this work, we studied a novel type of team formation problem: the *many teams to many tasks* allocation problem with no overlaps. First, we provided the formulation of the problem, and then we introduced the optimisation problem leading to a solution. Finally, we sketched an anytime heuristic approach that solves this optimisation problem. Currently, we are working on a systematic evaluation of our approach to confirm its effectiveness. Specifically, we are conducting an empirical analysis both using our approach in real-world problems and performing an experts' validation. Our initial results are promising regarding the quality of the solutions obtained by the heuristic algorithm.

[2]Note that for a fixed team allocation, finding the optimal competence assignment for one pair team-task is independent from finding the optimal competence assignment for any other pair.

REFERENCES

[1] Aris Anagnostopoulos, Luca Becchetti, Carlos Castillo, Aristides Gionis, and Stefano Leonardi. 2010. Power in Unity: Forming Teams in Large-Scale Community Systems. *International Conference on Information and Knowledge Management, Proceedings*, 599–608. https://doi.org/10.1145/1871437.1871515

[2] Ewa Andrejczuk, Rita Berger, Juan A. Rodríguez-Aguilar, Carles Sierra, and Víctor Marín-Puchades. 2018. The composition and formation of effective teams: computer science meets organizational psychology. *Knowledge Eng. Review* 33 (2018), e17. https://doi.org/10.1017/S026988891800019X

[3] Ewa Andrejczuk, Filippo Bistaffa, Christian Blum, Juan A. Rodríguez-Aguilar, and Carles Sierra. 2019. Synergistic team composition: A computational approach to foster diversity in teams. *Knowledge-Based Systems* 182, 104799 (10/2019 2019). https://doi.org/10.1016/j.knosys.2019.06.007

[4] Luca Capezzuto, Danesh Tarapore, and Sarvapali D. Ramchurn. 2020. Anytime and Efficient Coalition Formation with Spatial and Temporal Constraints. arXiv:2003.13806 [cs.MA]

[5] Yann Chevaleyre, Paul E Dunne, Ulle Endriss, Jerome Lang, Michel Lemaitre, Nicolas Maudet, Julian Padget, Steve Phelps, Juan A Rodrguez-Aguilar, and Paulo Sousa. 2006. Issues in Multiagent Resource Allocation. *Informatica* 30 (2006), 3–31.

[6] Chad Crawford, Zenefa Rahaman, and Sandip Sen. 2016. Evaluating the Efficiency of Robust Team Formation Algorithms. In *Autonomous Agents and Multiagent Systems*, Nardine Osman and Carles Sierra (Eds.). Springer International Publishing, Cham, 14–29.

[7] E. Czatnecki and A. Dutta. 2019. Hedonic Coalition Formation for Task Allocation with Heterogeneous Robots. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. 1024–1029.

[8] Theodoros Lappas, Kun Liu, and Evimaria Terzi. 2009. Finding a Team of Experts in Social Networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Paris, France) *(KDD '09)*. Association for Computing Machinery, New York, NY, USA, 467–476. https://doi.org/10.1145/1557019.1557074

[9] Sameera S. Ponda, Luke B. Johnson, Alborz Geramifard, and Jonathan P. How. 2015. *Cooperative Mission Planning for Multi-UAV Teams.* Springer Netherlands, Dordrecht, 1447–1490. https://doi.org/10.1007/978-90-481-9707-1_16

[10] Fredrik Präntare and Fredrik Heintz. 2018. An Anytime Algorithm for Simultaneous Coalition Structure Generation and Assignment. In *PRIMA 2018: Principles and Practice of Multi-Agent Systems*, Tim Miller, Nir Oren, Yuko Sakurai, Itsuki Noda, Bastin Tony Roy Savarimuthu, and Tran Cao Son (Eds.). Springer International Publishing, Cham, 158–174.