Solving the Synergistic Team Composition Problem

Extended Abstract

Ewa Andrejczuk
Artificial Intelligence Research Institute (IIIA-CSIC),
Change Management Tool S.L
Barcelona, Spain
ewa@iiia.csic.es

Filippo Bistaffa, Christian Blum
Juan A. Rodríguez-Aguilar, Carles Sierra
Artificial Intelligence Research Institute (IIIA-CSIC)
Barcelona, Spain
{filippo.bistaffa,jar,christian.blum,sierra}@iiia.csic.es

ABSTRACT

Co-operative learning is used to refer to learning procedures for heterogeneous teams in which individual and teamwork are organised to complete academic tasks. Key factors for team performance are competences, personality and team members' genders. Here, we present a computational model to form heterogeneous teams that incorporates those key factors. In addition, we propose efficient algorithms to partition a classroom into teams of even size and homogeneous performance. The first algorithm is based on an ILP formulation. For small problem instances this approach is appropriate. However, this is not the case for large problems, for which we propose a heuristic algorithm. We study the computational properties of both algorithms in the context of student teams.

KEYWORDS

Coalition Formation; Approximation Algorithms; Team Composition

ACM Reference Format:
Ewa Andrejczuk and Filippo Bistaffa, Christian Blum
Juan A. Rodríguez-Aguilar, Carles Sierra. 2018. Solving the Synergistic Team Composition Problem. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 3 pages.

1 INTRODUCTION

In this paper we want to address the following common education situation: there is a *complex task that has to be solved by different teams of students of the same size* [1]. The task requires that each team has at least one student that shows a minimum level of competence for a given set of competences. We have a pool of students with varying genders, personalities, and competences' levels. The problem is how to partition students into teams that are balanced in size, competences, personality, and gender. In other words, we want a team partition whose teams exhibit homogeneous performance. We refer to these teams as *synergistic teams*.

2 TEAM COMPOSITION MODEL

Our model considers that each student has a gender, personality, and competences with associated competence levels. A *team* is composed of at least two agents. A *task* has a type and a required number of agents to complete it. A task type determines the competence levels required for the task as well as their importance. Finally, given a team and a task type, a task assignment is a function that assigns all competences in a task type to some team member(s) so that each member is assigned to at least one competence.

Team proficiency. Given a task assignment for a team, a proficiency value measures the degree to which the task assignment covers the task requirements. That is, it measures the distances between the competence levels required by the task and those offered by the assignment.

Team congeniality. We measure personality using the Post-Jungian Personality Theory [3], which considers four numerical dimensions: Sensing–Intuition (SN), Thinking–Feeling (TF), Extroversion–Introversion (EI), and Perception–Judgment (PJ). Inspired by Wilde's experiments [4], we define team congeniality as an additive function that: (1) values more teams whose SN and TF dimensions are as diverse as possible; (2) prefers teams with at least one agent with positive EI and TF dimensions and negative PJ dimension, namely an extrovert, thinking and judging agent; (3) values more teams with at least one introvert agent; and (4) values gender balance.

Team synergy. Given a team, we obtain its *synergistic value* as a weighted combination of its proficiency and congeniality values. The setting of weights depends on each task type, since each task requires different levels of congeniality and proficiency. For instance, while creative tasks require intense communication and exchange of ideas (and hence much congeniality), difficult tasks may require higher levels of proficiency.

Partition synergy. We want to have teams that show a homogeneous behaviour so that there are no big differences in performance. To do that, we define the overall performance of a partition as the Bernoulli-Nash product of the individual team synergistic values, as this function gives larger values to homogeneous, i.e., "fair", solutions [2], than other functions like e.g. the sum.

The synergistic team composition problem (STCP). The STCP is the problem of partitioning a set of students into teams so that each team: (1) has even size, (2) is proficient given a task (i.e. competences requirements are covered as much as possible), and (3) is congenial (i.e. balanced in terms of personality and gender).

3 SOLVING THE STCP

Optimal algorithm. We observe that our problem can be linearised by using a logarithmic transformation of the partition synergy function. The algorithm starts by generating the input for an integer linear programming solver. That is, it generates all the possible teams of a given size and computes the best synergistic value per team. Next, it generates an integer linear programming encoding of the problem. The generated integer linear program (ILP) can be solved with the aid of an ILP like CPLEX or Gurobi.

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

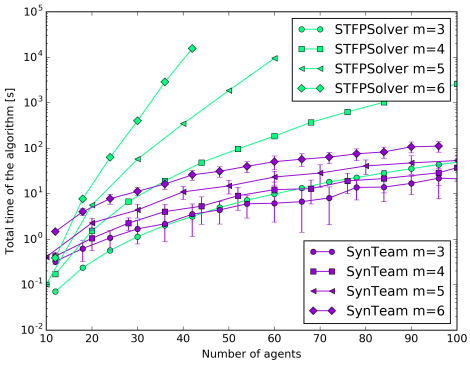

Figure 1: Total runtimes of SynTeam and STCPSolver.

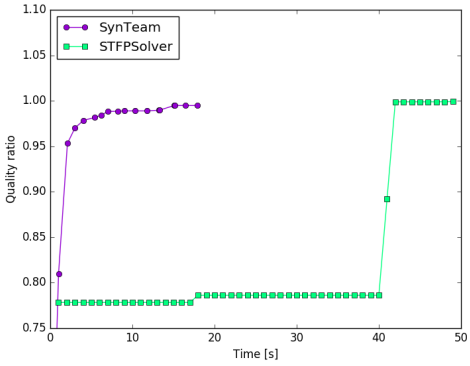

Figure 2: Anytime performance (in quality ratio) of SynTeam vs. STCPSolver ($n = 45$,$m = 5$).

Approximate algorithm. Our any-time algorithm, *SynTeam*, quickly finds an initial partition to subsequently improve it by performing student swaps between teams. First, it randomly composes teams given team sizes to generate an initial solution. Next, at each iteration, it randomly selects two teams from the current solution. Then, it computes the synergistic value of all partitions resulting from substituting the randomly selected teams by two new teams obtained by shuffling their members. In addition, if the current iteration is the n_l-th—not necessarily consecutive—non-improving iteration, the following more fine-grained procedure is applied: in the ascending order determined by team and student indexes it tries to swap two students from two different teams. The first improving solution found this way (if any) is chosen and the c_l counter, for non-consecutive non-improving iterations, is re-initialized. Finally, the algorithm stops after n_r consecutive non-improving iterations.

4 EXPERIMENTAL RESULTS

We ran our experiments on a 4-core machine with an Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz and 15 GB memory. STCPSolver used the 8 threads allowed by CPLEX, while SynTeam used one.

Runtime Analysis. Figure 1 shows the performance, in terms of total running time, of SynTeam and STCPSolver for different teams as the number of students increases. We performed 20 runs for each configuration, and recorded the total run time average and standard deviation. As team size (m) increases, generating the input for STCPSolver becomes prohibitively costly. Therefore, for STCPSolver we were only able to do calculations for: 102 students for $m \in \{3, 4\}$, 60 students for $m = 5$, and 42 students for $m = 6$. For larger values of n and m, reading the problem was beyond CPLEX capabilities (e.g. for $n = 48$ and $m = 6$ CPLEX handles over 12 million binary variables). We observe that the runtime of STCPSolver dramatically increases with the number of students (n) and team size (m). Note that for team size $m = 6$ and $n = 42$ students, SynTeam (using 1 thread) is at least 3 orders of magnitude faster than STCPSolver (using 8 threads).

Quality Analysis. For each case we calculated the optimality ratio. Specifically, we divided the solution obtained by SynTeam by the optimal solution calculated by STCPSolver. Figure 3 illustrates SynTeam's quality ratio with respect to the number of students

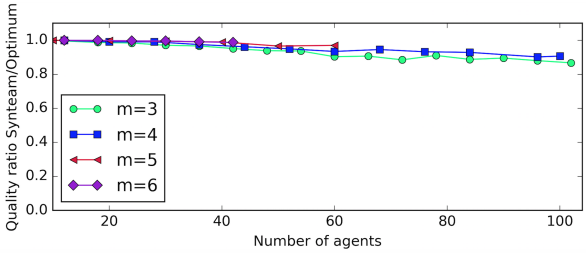

Figure 3: The quality ratio of the SynTeam algorithm.

and team sizes. The results show that the quality of approximate solutions slightly decreases with the number of students and team sizes but it always remains above 87%.

Anytime performance. We chose the configuration with $n = 45$ students and team size $m = 5$, since it is still in the region of problems that STCPSolver could afford. Figure 2 shows the evolution of the best solutions found over time (divided by the optimal solution) for both algorithms. Note that the problem generation time required by STCPSolver is not included, and hence we only plot the CPLEX solving time. Observe that SynTeam provides very good solutions in less than 3 seconds, while STCPSolver needs approximately 700 seconds (preprocessing time plus solving time) to come up with a first, low-quality solution. To conclude, to reach optimality, STCPSolver requires 233 times the time required by SynTeam to obtain solutions very close to optimality.

ACKNOWLEDGEMENTS

This work was supported by the CIMBVAL project (funded by MINECO, project number TIN2017-89758-R) the AppPhil project (funded by RecerCaixa 2017) and Collectiveware (TIN2015-66863-C2-1-R MINECO/ FEDER). Andrejczuk thanks an Industrial PhD scholarship from the Generalitat de Catalunya (DI-060) and Bistaffa the H2020-MSCA-IF-2016 HPA4CF project.

REFERENCES

[1] Silvia T Acuña, Marta Gómez, and Natalia Juristo. 2009. How do personality, team processes and task characteristics relate to job satisfaction and software quality? *Information and Software Technology* 51, 3 (2009), 627–639.
[2] J. Nash. 1950. The Bargaining Problem. *Econometrica* 18, 2 (April 1950), 155–162.
[3] D.J. Wilde. 2013. *Post-Jungian Personality Theory for Individuals and Teams.* SYDROSE LP.
[4] D. J. Wilde. 2011. *Jung's Personality Theory Quantified.* Springer-Verlag London.