Heterogeneous Teams for Homogeneous Performance

Ewa Andrejczuk[1,2], Filippo Bistaffa[1], Christian Blum[1], Juan A.Rodriguez-Aguilar[1], and Carles Sierra[1]

[1] Artificial Intelligence Research Institute (IIIA-CSIC), Bellaterra, Spain
[2] Change Management Tool S.L, Barcelona, Spain
{ewa, filippo.bistaffa, christian.blum,jar,sierra}@iiia.csic.es

Abstract. Co-operative learning is used to refer to learning procedures for heterogeneous teams in which individuals and teamwork are organised to complete academic tasks. Key factors of team performance are competencies, personality and gender of team members. Here, we present a computational model that incorporates these key factors to form heterogeneous teams. 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 when grouping students in a classroom into teams.

1 Introduction

Students learn best when they are actively engaged in the processing of information [24]. One way to involve students in active learning is to have them learn from one another within teams. Research shows that students working in teams tend to learn more and retain the knowledge longer than when the same content is presented by means of other instructional formats; they also appear more satisfied with their classes [6]. However, not just any team promotes learning. In order for learning to be productive, all teams in the classroom should be heterogeneous, that is, representative of the diversity of the whole class and balanced in size. Also, effective education must balance performance across teams, that is, performance should be as homogeneous as possible in the classroom: *No one should be left behind*.

Considerable work in fields such as organisational psychology, and industrial psychology has focused on various factors that influence team performance [5, 15, 25, 26]. [5, 26] underline the importance of personality traits or *types* for team composition. Other studies have focused on how team members should differ or converge in their characteristics, such as personality, competencies, or gender, among others [15, 25], in order to increase performance.

Also in the area of multiagent systems, team composition has attracted much research. MAS research has widely acknowledged competencies as important to perform tasks of different nature [9, 17, 21]. However, the majority of approaches represent capabilities of agents in a Boolean way (i.e., an agent either has a required skill or not). This is a simplistic way to model an agent's set of capabilities since it ignores any skill

degree. In real life, capabilities are not binary since every individual shows different performances for each competence. Additionally, the MAS literature has typically disregarded significant organizational psychology findings (with the exception of several recent, preliminary attempts like [11] or [3]). To the best of our knowledge, the current organizational psychology and MAS literature have not tackled how to compose teams taking into account the personality, gender and competencies of individuals.

Given this background, in this paper we address the following team composition problem commonly faced by educators. 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 competencies. We have a pool of students with varying genders, personalities, and competencies' levels. The problem is how to partition students into teams that are balanced in size, competencies, personalities, and gender. We refer to these teams as *synergistic teams*.

This paper makes the following contributions. First, we identify and formalise a new type of real-world problem: the synergistic team composition problem (STCP), requiring *balanced* solutions in terms of team size and team value. Second, we propose two algorithms to solve STCP: an algorithm to optimally solve it that is very efficient for small instances, and an approximate algorithm that is effective for larger instances. And third, a computational comparison of both algorithms over realistic settings in an education context.

Outline. The remainder of this paper is structured as follows. Section 2 introduces basic definitions required by our team composition problem. Section 3 introduces the synergistic team composition problem. Section 4 details how to compute a team's synergistic value. Sections 5 and 6 describe how to optimally and approximately solve the synergistic team composition problem respectively. Then, Section 7 reports on our empirical analysis of both algorithms over artificially-generated instances of the synergistic team composition problem. Finally, Section 8 draws some conclusions and sets paths to future research.

2 Basic definitions

We consider that each student has a gender, personality, and competencies.

First, to measure personality, we explore a novel method: the Post-Jungian Personality Theory [28], a modified version of the Myers-Briggs Type Indicator (MBTI) [8].[3] This questionnaire is short, contains only 20 quick questions (compared to the 93 MBTI questions). This is very convenient for both experts designing teams and individuals doing the test since completing the test takes just a few minutes (for details of the questionnaire, see [28, p.21]). In contrast to the MBTI measure, which consists of four binary dimensions, the Post-Jungian Personality Theory uses the *numerical* data collected using the questionnaire [27]. The results of this method seem promising, since within a decade this novel approach has tripled the fraction of Stanford teams awarded US prizes by the Lincoln Foundation [27]. The test is based on the pioneering psychiatrist C. G. Jung's personality model [14]. It has two sets of variable pairs called

[3] MBTI numerical values can be used with the same purpose.

psychological functions: (1) Sensing / Intuition (SN), and (2) Thinking / Feeling (TF) and two sets of attitudes: (3) Extroversion / Introversion (EI), and (4) Perception / Judgment (PJ).

Psychological functions and attitudes compose together a personality. The numerical values for each dimension of a personality (SN, TF, EI, PJ) are measured through a five multiple choice true/false questions. Thus,

Definition 1 *A* personality profile *is a tuple* $\langle sn, tf, ei, pj\rangle \in [-1, 1]^4$*, where each of these four components represents one personality trait.*

Second, a competence integrates the knowledge, skills and attitudes that enable a student to act correctly in a job, task or situation [22]. Each student is assumed to possess a set of competencies with associated competence levels. Let $C = \{c_1, \ldots, c_k\}$ be the whole set of competencies, where each element $c_i \in C$ stands for a competence.

Definition 2 *A* student *is represented as a tuple* $\langle id, g, \mathbf{p}, l\rangle$ *such that:* id *is an identifier;* $g \in \{man, woman\}$ *is a gender;* $\mathbf{p} = \langle sn, tf, ei, pj\rangle$ *is a personality profile;* $l : C \to [0, 1]$ *is a function that assigns the quality level of the outcome with respect to competence* c*.*[4]

Henceforth, we will note the set of students as $A = \{a_1, \ldots, a_n\}$. Moreover, we will use super-indexes to refer to students' attributes. For instance, given a student $a \in A$, id^a will refer to the id attribute of student a.

Definition 3 (Team) *A* team *is any subset of* A *with at least two students.*

We denote by $\mathcal{K}_A = (2^A \setminus \{\emptyset\}) \setminus \{\{a_i\} | a_i \in A\}$ the set of all possible teams in A.

Finally, a *team* is any subset of A with at least two students. We denote by $\mathcal{K}_A = (2^A \setminus \{\emptyset\}) \setminus \{\{a_i\} | a_i \in A\}$ the set of all possible teams in A.

3 The synergistic team composition problem

We can regard our team composition problem as a particular type of set partitioning. We will refer to any partition of A as a *team partition*. Since all teams should have an even size, we only consider team partitions whose teams are constrained by a given size.

Definition 4 *Given a set of students* A*, we say that a team partition* P_m *of* A *is constrained by size* m*,* $|A| \geq m \geq 2$*, iff for every team* $K \in P_m$*,* $m \leq |K| \leq m + 1$*.*

As $|K|/m$ is not necessarily a natural number, we may need to allow for some flexibility in team size within a partition. This is why we introduced above the condition $m \leq |K| \leq m + 1$. In practical terms, in a partition we want to have teams of sizes differing by at most one student. This is a common constraint when partitioning a classroom: we want teams to be balanced in size. We note by $\mathcal{P}_m(A)$ the set of all team partitions of A constrained by size m.

[4] We assume that the competence level is zero when a student does not have a competence (or we do not know its value).

The question is: which partition to choose? We want to have teams that show a homogeneous behaviour so that there are no big differences in performance (i.e., we do not want partitions for which some teams perform well and some poorly; Remember, no one is to be left behind!). To do that, we first define the synergistic value of a team K, noted as $s(K)$, as an expectation of its performance. We present the formal definition of such a function in section 4. Second, we define the overall performance of a partition as the Bernoulli-Nash product of individual teams' synergistic values, since this function evaluates better homogeneous ("fair") solutions [16] than other functions (e.g. the sum).

Definition 5 *Given a team partition P_m, the synergistic value of P_m is*

$$S(P_m) = \prod_{K \in P_m} s(K). \tag{1}$$

Thus, the STCP is solved by finding the partition with the highest synergistic value.

Definition 6 *Given a set of students A the* synergistic team composition problem (STCP) *is the problem of finding a team partition constrained by size m, $P_m^* \in \mathcal{P}_m(A)$, that maximises $S(P_m)$, namely:*

$$P_m^* = \underset{P_m \in \mathcal{P}_m(A)}{\arg\max}\ S(P_m)$$

3.1 Relation with the coalition formation literature

The STCP is a particular case of a coalition generation problem [20]. Unfortunately, we cannot benefit from the algorithms in the literature. In particular, following [19], given a STCP we can identify a constrained coalition formation (CCF) game $\mathcal{G} = \langle A, \mathcal{P}_m(A), s \rangle$, where A is the set of students, $\mathcal{P}_m(A)$ is the set of feasible coalition structures (i.e. team partitions constrained by size m as per definition 4), and s is the characteristic function (synergistic value function) that assigns a real value to every coalition (team) that appears in some feasible coalition structure (team partition). Given the former CCF game, solving the STCP amounts to finding a coalition structure (team partition) with the highest total value. More precisely, the STCP poses a particular type of CCF game, a so-called *basic* CCF game [20]. Intuitively, the constraints in a basic CCF game are expressed in the form of: (1) sizes of coalitions that are allowed to form; and (2) subsets of students whose presence in any coalition is viewed as desirable/prohibited. On the one hand, a STCP naturally defines constraints on the size of coalitions. On the other hand, expressing a STCP as a CCF problem requires one positive constraint per feasible team (i.e., q positive constrains), while the set of negative constraints is empty. The number of positive constraints is so large for the problems we want to solve (i.e. > 3000) that these problems are prohibitive for the algorithm in [19].

4 Computing team synergistic values

A team K is effective solving a task when it is both *proficient* (covers the required competencies) and *congenial* (balances gender and psychological traits so that students work well together) [28]. We linearly combine these two aspects ($u_{prof}(K)$ and $u_{con}(K)$, respectively) into the synergistic value of K as follows:

Definition 7 *Given a team K, the synergistic value of team K is defined as:*

$$s(K) = \lambda \cdot u_{prof}(K) + (1 - \lambda) \cdot u_{con}(K) \tag{2}$$

$\lambda \in [0, 1]$ *is the relative importance of K being proficient.*

In general, the higher the value of λ, the higher the importance for the proficiency of a team. The setting of the value of λ depends on the task type. For instance, task types that are difficult and performed for the first time (no experts on that matter) require a high level of creativity and exchange of ideas, and hence, personality and gender balance (congeniality) should be more important than proficiency ($\lambda < 0.5$). However, for tasks where team members need to act fast (such as sport competitions or rescue teams) it is crucial for a team to be proficient ($\lambda > 0.5$). For creative task types that require certain levels of both proficiency and congeniality (such as creating a webpage) the value of λ should be set to 0.5 (so that congeniality and proficiency are equally important). The next subsections detail how to measure team proficiency and congeniality.

4.1 Evaluating team proficiency

Given a team and a task, we want to calculate the *degree of proficiency* of the team as a whole, noted u_{prof}. In other words, our aim is to match each competence with the student(s) whose personal competence level is closer to the task competence level requirement. With this we aim at avoiding both *under-proficient* and *over-proficient* allocations as both of those scenarios are ominous for team performance. In the first case, under-proficient students may get frustrated because they do not have enough knowledge to cope with the assigned competence requirements. In the second case, *over-proficient* students may get distracted and unmotivated because of the easiness of a job they are asked to do [7]).

In other words, given a team and a task, we want to measure how apt is the team to solve the task. We understand a task as a particular instance of a *task type* that specifies the competencies and competence levels required to solve it.

Definition 8 *A task type τ is defined as a tuple $\langle \lambda, \{(c_i, l_i, w_i)\}_{i \in I_\tau} \rangle$, where I_τ is the index set of the required competencies; $\lambda \in [0, 1]$ is the importance given to proficiency; $c_i \in C$ is a required competence; $l_i \in [0, 1]$ is the required competence level for c_i; $w_i \in [0, 1]$ is the importance of competence c_i; and $\sum_{i \in I_\tau} w_i = 1$.*

A task is an instance of a task type defined as:

Definition 9 *A* task *t is a tuple $\langle \tau, m \rangle$ such that τ is a task type and m is the required number of students, where $m \geq 2$.*

Henceforth, we denote by T the set of tasks and by $\mathcal{T}$ the set of task types. Moreover, we will note as $C_\tau = \{c_i | i \in I_\tau\}$ the set of competencies required by task type τ.

Students must feel both accountable and useful when working in a team [23]. Hence, each team member must be responsible for at least one competence; this is expressed as a *competence assignment* between competencies and students:

Definition 10 *Given task type τ and a team $K \in \mathcal{K}_{\mathcal{A}}$, a competence assignment is a function $\eta : K \rightarrow 2^{C_\tau}$ satisfying that $C_\tau = \bigcup_{a \in K} \eta(a)$. We note by Θ_τ^K the set of competence assignments for task type τ and team K.*

The degree of proficiency of a team will obviously depend on the particular student(s) assigned to each competence.

Definition 11 *Given task type τ, team K, and competence assignment η, the set $\delta(c_i) = \{a \in K | c_i \in \eta(a)\}$ stands for those students responsible of competence c_i.*

Informally, our aim is to match each competence c_i with the student(s) $\delta(c_i)$ whose personal competence level is closer to the task competence level requirement. With this we aim at avoiding both *under-proficient* (frustrated students because they cannot cope) and *over-proficient* (frustrated students because they get bored [7]) allocations.

Definition 12 (Degree of under-proficiency)
Given a task type τ, a team K, and an assignment η, we define the team's degree of under-proficiency for the task as:

$$u(\eta) = \sum_{i \in I_\tau} w_i \cdot \frac{\sum_{a \in \delta(c_i)} |\min(l^a(c_i) - l_i, 0)|}{|\delta(c_i)| + 1}$$

Definition 13 (Degree of over-proficiency)
Given a task type τ, a team K, and an assignment η, we define the team's degree of over-proficiency for the task as:

$$o(\eta) = \sum_{i \in I_\tau} w_i \cdot \frac{\sum_{a \in \delta(c_i)} \max(l^a(c_i) - l_i, 0)}{|\delta(c_i)| + 1}$$

Finally, we can calculate the team's proficiency degree to perform a task by combining its over-proficiency and under-proficiency as follows:

Definition 14 *Given a team K and a task of type τ, the proficiency degree of the team to perform an instance of τ is:*

$$u_{prof}(K) = \max_{\eta \in \Theta_\tau^K} (1 - (\upsilon \cdot u(\eta) + (1 - \upsilon) \cdot o(\eta)) \tag{3}$$

where $\upsilon \in [0, 1]$ is the penalty given to the under-proficiency of team K.

If we want to penalise teams that cannot cope with the competence requirements (i.e. they are under-competent) we need to choose a large value for υ. And similarly a small υ to penalise teams with members clearly over-competent. Although the exact value to choose will depend on the particular task type and student context, if the objective is to favour *effective* teams we should penalize more their under-proficiency and thus select a significantly large value for υ. Given these definitions, $u_{prof}(K)$ is correctly defined for any team, task type and competence assignment:

Proposition 1. *For any task type τ, team K, and $\eta \in \Theta_\tau^K$, $u(\eta) + o(\eta) \in [0, 1)$ and $0 \leq u_{prof}(K) < 1$.*

Proof. Soundness is straightforward as a student cannot be over- and under-proficient at the same time.

Computing $u_{prof}(K)$ is an optimisation problem: to have each competence assigned to at least one student and each student assigned to at least one competence so that the total *cost* of the assignment is minimal (in terms of under- and over-proficiency). Such optimisation problem can be cast and efficiently solved as a minimum cost flow problem [2]. The network model would contain $v = |K| + |C_\tau| + 2$ nodes and $e = |K| \cdot |C_\tau| + |K| + |C_\tau|$ edges. As discussed in [18], the minimum cost flow problem can be solved in $O(e \cdot log(v) \cdot (e + v \cdot log(v)))$ on a network with v nodes and e arcs.

4.2 Evaluating team congeniality

Given a team and a task, we also need to measure the *degree of congeniality* of the team, u_{con}, that is, how well do students work together in a creative and co-operative atmosphere. According to [10], the only truthful collaboration is the one containing tension, and disagreement as these improve the value of the ideas, expose the risks inherent in plan, and lead to enhanced trust among the team members. This conflict is generated by people having different views of the world (associated with opposing personality and gender), whereas harmony comes from agreement between people with similar personalities [28]. Based on these findings Douglas J. Wilde [27] compiled heuristics to successfully compose teams. According to Wilde's findings the most successful teams are: (i) teams whose SN and TF personality dimensions are as diverse as possible; (ii) teams with at least one student with positive EI and TF dimensions and negative PJ dimension, namely an extrovert, thinking and judging student (called ETJ personality); (iii) teams with at least one introvert student; and (iv) teams with gender balance. Hence, to define the *degree of congeniality* we get inspiration from [27] where D. J. Wilde uses psychological traits (see Section 2) to form successful teams. Formally, this can be captured by function:

$$u_{con}(K) = u_{SNTF}(K) + u_{ETJ}(K) + u_I(K) + u_{gender}(K),$$

with:

1. $u_{SNTF}(K) = \sigma(K, SN) \cdot \sigma(K, TF)$ measures the diversity in a team, where $\sigma(K, SN)$ and $\sigma(K, TF)$ stand for the standard deviations over the SN and TF personality traits of the members of team K. The larger the values of $\sigma(K, SN)$ and $\sigma(K, TF)$, the larger their product, and hence the larger the personality diversity along the SN and TF dimensions within a team.
2. $u_{ETJ}(K) = \max_{a \in K^{ETJ}}[\max(\boldsymbol{\alpha} \cdot \mathbf{p}, 0), 0]$ measures the utility of counting on ETJ personalities, being $K^{ETJ} = \{a \in K | tf^a > 0, ei^a > 0, pj^a > 0\}$ the set of students exhibiting ETJ personality, $\boldsymbol{\alpha} = (0, \alpha, \alpha, \alpha)$ is a vector, and α is the importance of counting on an extrovert, thinking, and judging student (ETJ personality).

3. $u_I(K) = \max_{a \in K}[\max(\boldsymbol{\beta} \cdot \mathbf{p}, 0), 0]$ is the utility of counting on an introvert student, $\boldsymbol{\beta} = (0, 0, -\beta, 0)$ is a vector and β is the importance of introvert students.
4. $u_{gender}(K) = \gamma \cdot \sin(\pi \cdot g(K))$ measures the importance of gender balance, where γ is a parameter to weigh the importance of gender balance, and $g(K) = \frac{w(K)}{w(K)+m(K)}$ calculates the ratio of women in a team ($w(K)$ and $m(K)$ are functions counting the number of women and men, respectively). A team K is perfectly gender-balanced iff $w(K) = m(K)$, and hence $\sin(\pi \cdot g(K)) = 1$.

5 Solving the STCP optimally

Next we study how to optimally solve the STCP. We start by linearising the problem in section 5.1. This allows us to solve the STCP with the aid of off-the-shelf solvers. Thereafter, in section 5.2 we detail an optimal algorithm for the STCP.

5.1 Linearising the STCP

We denote by $n = |A|$ the number of students in A, by t a task of type $\langle \tau, m \rangle$, and by b the total number of teams, $b = \lfloor n/m \rfloor$. Note that depending on the cardinality of A and the desired team size, the number of students in each team may vary in size. For instance, if there are $n = 7$ students in A and we want to compose duets, we split students into two duets and one triplet. In general, whenever $n \geq m$: if $n \bmod m = 0$, each partition must contain b teams of size m; and if $n \bmod m \leq b$, each partition must contain $b - (n \bmod m)$ teams of size m and $n \bmod m$ teams of size $m + 1$.[5] Let $Q(n, m)$ be the quantity distribution of students in teams of sizes m and $m + 1$; these are called *feasible* teams.

Notice that the total number of feasible teams is $q = \binom{n}{m} + \min(n \bmod m, 1) \cdot \binom{n}{m+1}$. Therefore, let $K_1, \ldots, K_q$ denote the feasible teams in A, and $s(K_1), \ldots, s(K_q)$ their synergistic values concerning task t. Moreover, let b be the number of teams required to form a team partition. Finally, let C be a matrix of size $n \times q$ such that c_{ij} takes on value 1 if student a_i is part of team K_j, and 0 otherwise.

We shall consider the set of binary decision variables x_j, $1 \leq j \leq q$, to indicate whether team K_j is selected or not as part of the optimal solution of the STCP. Then, solving the STCP amounts to solving the following non-linear problem:

$$\max \prod_{j=1}^{q} s(K_j)^{x_j} \tag{4}$$

subject to:

$$\sum_{j=1}^{q} x_j = b \tag{5}$$

[5] Beyond these cases, there is no way to compute a partition constrained by m (see def. 4). If so, $m' \leq m$, $m' = \lfloor n/(b+1) \rfloor$ is the largest value smaller than m that can be used to compute partitions.

$$\sum_{j=1}^{b} c_{ij} \cdot x_j = 1 \quad \forall 1 \leq i \leq n \tag{6}$$

$$x_j \in \{0,1\} \quad 1 \leq j \leq q \tag{7}$$

Notice that constraint 5 enforces that the number of teams in the team partition is b, whereas constraint 6 enforces that the selected teams form a partition by imposing that no student can belong to two selected teams at the same time. Now observe that equation 4 —the objective function— is non-linear. Nevertheless, it can be readily linearised if we consider the logarithm of $\prod_{j=1}^{q} s(K_j)^{x_j}$ as our objective function to maximise. Thus, solving the non-linear problem above is equivalent to solving the following binary linear program:

$$\max \sum_{j=1}^{q} x_j \cdot log(s(K_j)) \tag{8}$$

subject to: equations 5, 6, and 7.

5.2 An algorithm to optimally solve the STCP

Algorithm 1 shows the pseudocode of an optimal solver for the STCP. The algorithm starts by generating the input for an integer linear programming solver (lines 2 to 5). Line 2 generates all the possible teams of size m as dictated by the quantity distribution $Q(|A|, m)$. Thereafter, lines 3 and 4 compute the best synergistic value per team. That is, these lines compute (1) the competence assignment with the highest proficiency value. This amounts to solving an optimisation problem, as discussed at the end of subsection 4.1, and (2) the team's congenial value from the personalities and genders of the team members. Once all synergistic values are computed, we can generate an integer linear programming encoding of the problem like in equation 8 (line 5). The generated integer linear program (ILP) can be solved with the aid of an ILP solver (line 6) such as, for instance, CPLEX, Gurobi, or GLPK. Finally, the algorithm returns the team partition together with the competence assignments (line 7).

Algorithm 1 STCPSolver

Require: A ▷ The set of students
Require: $t = \langle \tau, m \rangle$ ▷ Task
Ensure: $(P, \boldsymbol{\eta^*})$ ▷ Best partition found and best assignments
1: $P \leftarrow \emptyset$
2: $[K_1, \ldots, K_q] \leftarrow GenerateTeams(A, Q(|A|, m))$
3: **for** $i \in [1..q]$ **do**
4: $\quad (s(K_i), \eta_i^*) \leftarrow getBestSynergisticValue(K_i, t)$
5: $ILP \leftarrow generateILP([K_1, \ldots, K_q], [s(K_1), \ldots, s(K_q)], b)$
6: $P \leftarrow solve(ILP)$
7: **return** $(P, \{\eta_i^*\}_{K_i \in P})$

The cost of optimally solving an STCP can be split into: the cost of generating the ILP model, and the cost of solving it. As to the first cost, this comes from: (i) generating all

the teams of sizes given by $Q(n, m)$ (line 2); (ii) computing the synergistic values of all teams (lines 3 and 4); (iii) generating a linear programming encoding (line 5). The cost of generating all teams is linear with the total number of teams, and hence $O(q)$. Note that the number of teams grows rapidly with increasing m and n. Moreover, the cost of computing the synergistic value for each team requires finding the optimal competence assignment. As discussed in Sec. 4.1, this can be cast as a minimum cost flow problem and solved in $O(m \cdot log(e) \cdot (m + e \cdot log(e)))$ time, where $e = m \cdot |C_\tau|$, being $|C_\tau|$ the number of competencies required by task type τ. Thus, generating the input to an ILP solver becomes increasingly costly as the number of students per team grows.

6 An approximate algorithm for the STCP

In this section we present an approximate algorithm — *SynTeam* (see Algorithm 2). *SynTeam* quickly finds an initial partition, to subsequently improve it by performing student swaps between teams. First, it randomly orders the list of students and assigns students to teams one by one from that list following $Q(|A|, m)$ (see Sec. 5.1) to generate an initial solution $(P, S(P), \boldsymbol{\eta})$ (line 1). The assignment of students to competencies is solved as described in subsection 4.1.

Second, at each iteration, SynTeam generates a random neighbour of the current solution as follows (line 4). First, 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 (and corresponding competence assignments. see Subsection 4.1) formed by reordering the students of the randomly selected teams in all possible ways. It stores the best option in $(P', S(P'), \boldsymbol{\eta}')$. In addition, if the current iteration is the n_l-th—not necessarily consecutive—non-improving iteration,[6] the following more fine-grained procedure is applied to $(P, \boldsymbol{\eta})$ (line 6). 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 stored in $(P', \boldsymbol{\eta}')$ 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.

7 Experimental Results

In this section we compare our two STCP solvers: optimal (STCPSolver), and approximate (SynTeam). Our empirical evaluation compares: (1) their runtimes as team sizes and number of students increase; (2) the quality of SynTeam's approximate solutions; (3) the anytime performance of SynTeam with respect to STCPSolver.

7.1 Empirical settings

Our empirical evaluation employs the following settings:

[6] If the current solution is improved at an iteration, we refer to it as an improving iteration, a non-improving iteration otherwise.

Algorithm 2 SynTeam

Require: A ▷ The list of students
Require: n_r ▷ Max. # of consecutive non-impr. iterations
Require: n_l ▷ # of non-impr. iterations before student-swap
Ensure: $(P, \boldsymbol{\eta})$ ▷ Best partition found and best assignments

$(P, S(P), \boldsymbol{\eta}) \leftarrow GenerateRandomSolution(A, Q(|A|, m))$
$c_r \leftarrow 1, c_l \leftarrow 1$
while $c_r \leq n_r$ **do**
  $(P', S(P'), \boldsymbol{\eta}') \leftarrow GenerateRandomNeighbor(P, \boldsymbol{\eta})$
  if $S(P') \leq S(P)$ **and** $c_l = n_l$ **then**
    $(P', S(P'), \boldsymbol{\eta}') \leftarrow ApplyImprovingSwap(P, \boldsymbol{\eta})$
    $c_l \leftarrow 1$
  if $S(P') > S(P)$ **then**
    $(P, S(P), \boldsymbol{\eta}) \leftarrow (P', S(P'), \boldsymbol{\eta}')$
    $c_r \leftarrow 1, c_l \leftarrow 1$
  else
    $c_r \leftarrow c_r + 1, c_l \leftarrow c_l + 1$
return $(P, \boldsymbol{\eta})$

- **LP Solver.** We used CPLEX Optimization Studio v12.7.1 [13] for STCPSolver.
- **Students.** We used actual-world data from 102 students, each one with an id, a gender, a personality profile, and seven competencies with varying competence levels.
- **Task type.** The task type used in our experiments here $\{(c_i, l_i, w_i)\}_{i \in [1,7]}$ was the same as the one used in our study involving real students [4]. It had seven equally important competencies, $w_i = 1/7$, with a maximally competence level requirement, $l_i = 1$, and the importance of proficiency set larger than congeniality, $\lambda = 0.8$. In an educational context, task types requiring more than seven competencies are rare and thus the task type used here is complex enough for our purposes [12].
- **Task.** Team size m ranged from 3 to 6. Larger team sizes were not considered because the generated STCPs were too costly for STCPSolver and rare in an education context.
- **Team proficiency.** As in this paper we are just interested in the computational properties of the algorithms, the concrete value for υ is irrelevant. We used $\upsilon = 1$.
- **Team Congeniality.** We analytically assessed that to make each component of the personality requirements equally relevant, we must set importance values as follows: (1) $\alpha = 0.11$, (2) $\beta = 3 \cdot \alpha$, (3) $\gamma = 0.33$.
- **Number of iterations without improvement (n_r).** To give SynTeam a chance to visit all teams at least once without revisiting the same teams too many times, we decided to set n_r based on the value of b (number of teams in a partition). We experimentally observed how the quality of SynTeam solutions improved over time. Thus, setting n_r to $1.5 \cdot b$ offered a good compromise.
- **Frequency of local search (n_l).** We empirically observed that, after performing approximately $\frac{n_r}{6}$ random team re-compositions without improvement, the probability of finding an improvement was very low. Hence, we set n_l to $\frac{n_r}{6}$.

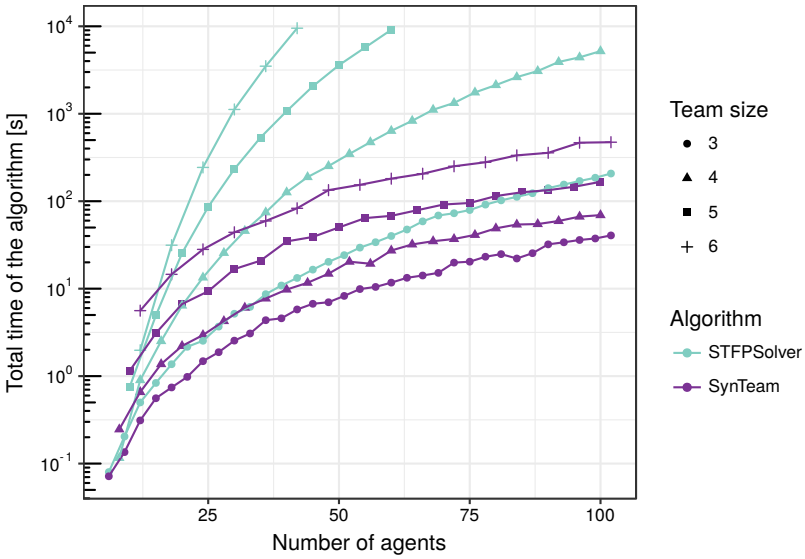

Fig. 1: SynTeam vs STCPSolver runtimes.

7.2 Computational Results

The experimental evaluation was performed on a cluster of PCs with Intel(R) Xeon(R) CPU 5670 CPUs of 12 nuclei of 2933 MHz and at least 40 Gigabytes of RAM. Moreover, we used IBM ILOG CPLEX v12.7.1 within both STCPSolver and SynTeam. Note that CPLEX is used within SynTeam in order to calculate, given a team, the optimal assignment of students to tasks. Moreover, note that CPLEX was run in one-threaded mode, in order to be able to perform a fair comparison.

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.[7] 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 is more than two orders of magnitude faster than STCPSolver.

To better understand this result, we compared STCPSolver solving time (only CPLEX time) with SynTeam. That is, we disregard the time required by STCPSolver to generate the problem (lines 1-5 in alg. 1). Figure 2 shows this comparison. We observe that —

[7] For instance, CPLEX must consider 12.271.512 binary variables for $n = 48$ and $m = 6$.

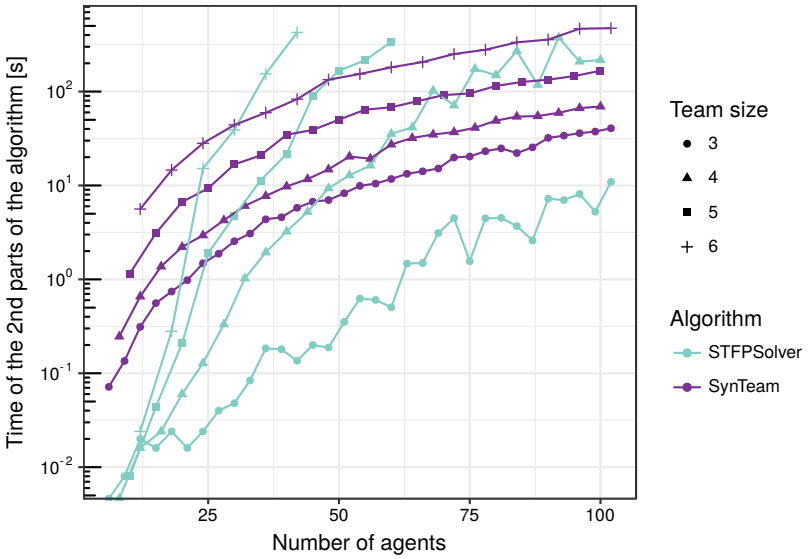

Fig. 2: SynTeam vs. STCPSolver solving times (disregarding problem generation time).

even in this case — SynTeam is more efficient for larger instances (team sizes $m > 3$ and a growing number of students).

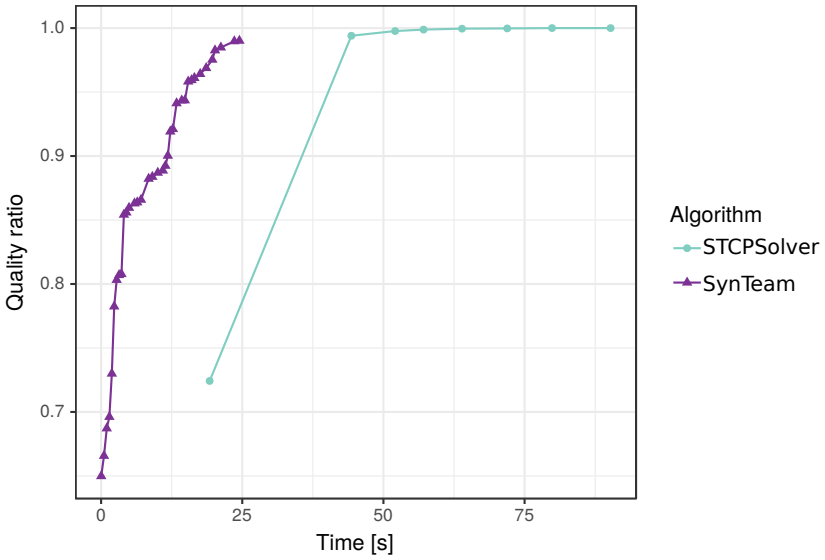

Fig. 3: Anytime performance (in quality ratio) of SynTeam vs. STCPSolver ($n = 45$,$m = 5$).

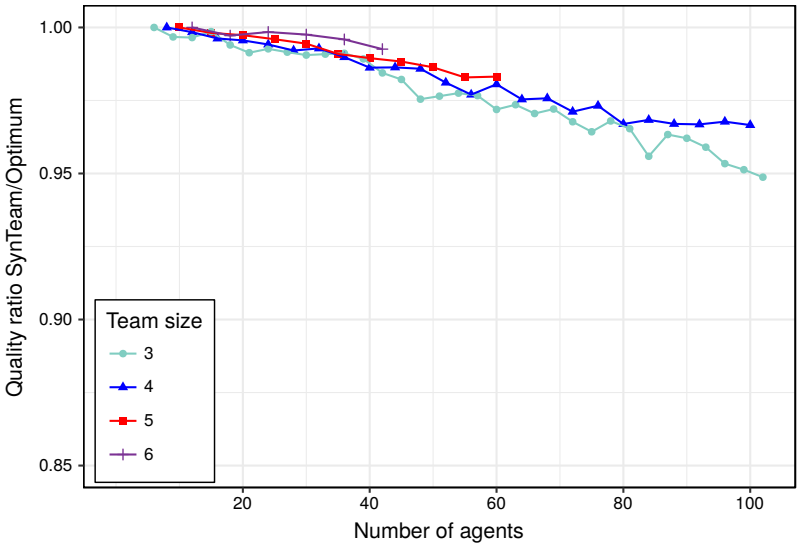

Fig. 4: SynTeam quality ratio.

Quality Analysis. For each case we calculated the optimality ratio. Specifically, we divided the solution obtained by SynTeam by the optimal solution calculated by STCP-Solver. Figure 4 illustrates this quality ratio with respect to the number of students 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 approx. 95%.

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 3 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 approx. 15 seconds, while STCPSolver needs approximately 20 seconds (in addition to more than 1000 seconds of preprocessing time) to come up with a first, low-quality solution. To conclude, to reach optimality, STCPSolver requires nearly two orders of magnitude more time than the one required by SynTeam to obtain solutions very close to optimality.

8 Conclusions

In this paper, we considered the Synergistic Team Composition Problem (STCP) in the context of student team composition and proposed both an optimal and an approximate solution to this problem. First, we discussed an algorithm to optimally solve the STCP called STCPSolver. When we noticed that the algorithm is only effective for small instances of the problem, we developed SynTeam, a greedy algorithm for partitioning groups of humans into proficient, gender, psychologically and size balanced

teams, which yields a good, but not necessarily optimum solution. Our computational evaluation shows that the larger the number of students and team sizes, the larger the benefits of SynTeam with respect to STCPSolver. Furthermore, SynTeam provides good quality approximate solutions (beyond 95% with respect to the optimal).

This paper identified a real-world instance of an interesting new type of constrained coalition formation problem requiring a *balanced* coalition structure in terms of coalition sizes and coalitional values. The computational analysis of our proposed algorithms gives the guidelines for their use by any organisation that faces the need to form problem solving teams (e.g. in a classroom, in a company, in a research unit). The algorithm composes teams in a purely automatic way without consulting experts, which is a huge advantage for environments where there is a lack of experts.

Finally, we have implemented a freely available web-based application to solve the STCP that automatically selects which algorithm to use depending on the size of the problem. It is available here: *https://eduteams.iiia.csic.es*.

This new problem, STCP, has potential to spur future research. In particular, we aim at considering richer and more sophisticated models to capture the various factors that influence the coalition composition process in the real world. For instance, we want to be able to add constraints and preferences coming from experts that cannot be established by any algorithm, e.g. Ana cannot be in the same team with José as they used to have a romantic relationship.

Acknowledgements

This work was supported by the CIMBVAL project (funded by MINECO, project number TIN2017-89758-R), 2017 SGR 172, the AppPhil project (funded by RecerCaixa 2017) and Collectiveware (TIN2015-66863-C2-1-R MINECO/ FEDER). Bistaffa was supported by the H2020-MSCA-IF-2016 HPA4CF project. Andrejczuk thanks an Industrial PhD scholarship from the Generalitat de Catalunya (DI-060).

References

1. S. T. Acuña, M. Gómez, and N. Juristo. How do personality, team processes and task characteristics relate to job satisfaction and software quality? *Information and Software Technology*, 51(3):627–639, 2009.
2. R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. Network flows: theory, algorithms, and applications. 1993.
3. J. M. Alberola, E. Del Val, V. Sanchez-Anguix, A. Palomares, and M. D. Teruel. An artificial intelligence tool for heterogeneous team formation in the classroom. *Knowledge-Based Systems*, 101:1–14, 2016.
4. E. Andrejczuk, J. A. Rodríguez-Aguilar, C. Roig, and C. Sierra. Synergistic team composition. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pages 1463–1465. International Foundation for Autonomous Agents and Multiagent Systems, 2017.
5. J. Arnold and R. Randall. *Work psychology.* Pearson Education Limited., Harlow, England, 2010.

6. E. F. Barkley, K. P. Cross, and C. H. Major. *Collaborative learning techniques: A handbook for college faculty*. John Wiley & Sons, 2014.
7. RM Bashur, A Hernandez, and JM Peiro. The impact of underemployment on individual and organizational performance. *Underemployment: Psychological, Economic, and Social Challenges içinde (ss. 187-213) New York: Springer*, 2011.
8. I. Briggs and P.B. Myers. *Gifts Differing: Understanding Personality Type*. Mountain View, CA: Davies-Black Publishing, 1995.
9. B. Chen, X. Chen, A. Timsina, and L. Soh. Considering agent and task openness in ad hoc team formation. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, Istanbul, Turkey, May 4-8, 2015*, pages 1861–1862, 2015.
10. L. Davey. If your team agrees on everything, working together is pointless. *Harvard Business Review*, 2017.
11. M. Farhangian, M. K. Purvis, M. Purvis, and B. T. R. Savarimuthu. Modeling the effects of personality on team formation in self-assembly teams. In *PRIMA 2015: Principles and Practice of Multi-Agent Systems - 18th International Conference, Bertinoro, Italy, October 26-30, 2015, Proceedings*, pages 538–546, 2015.
12. H. Gardner. The theory of multiple intelligences. *Annals of Dyslexia*, 37(1):19–35, 1987.
13. IBM. Ibm ilog cplex optimization studio, 2017.
14. C.G. Jung. *Psychological types*. Princeton University Press, Princeton, 1921.
15. M.K. Mount, M.R. Barrick, and G.L. Stewart. Five-factor model of personality and performance in jobs involving interpersonal interactions. *Human performance*, 11(2-3):145–165, 1998.
16. J. Nash. The Bargaining Problem. *Econometrica*, 18(2):155–162, April 1950.
17. T. Okimoto, N. Schwind, M. Clement, T. Ribeiro, K. Inoue, and P. Marquis. How to form a task-oriented robust team. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '15, pages 395–403. International Foundation for Autonomous Agents and Multiagent Systems, 2015.
18. J. B. Orlin. A faster strongly polynomial minimum cost flow algorithm. *Operations research*, 41(2):338–350, 1993.
19. T. Rahwan, T. P. Michalak, E. Elkind, P. Faliszewski, J. Sroka, M. Wooldridge, and N. R. Jennings. Constrained coalition formation. In Wolfram Burgard and Dan Roth, editors, *AAAI*. AAAI Press, 2011.
20. T. Rahwan, T. P. Michalak, M. Wooldridge, and N. R. Jennings. Coalition structure generation: A survey. *Artificial Intelligence*, 229:139–174, 2015.
21. S. S. Rangapuram, T. Bühler, and M. Hein. Towards realistic team formation in social networks based on densest subgraphs. *CoRR*, abs/1505.06661, 2015.
22. RA Roe. Competences-a key towards the integration of theory and practice in work psychology. *Gedrag en Organisatie*, 15(4):203–224, 2002.
23. Robert E Slavin. Synthesis of research of cooperative learning. *Educational leadership*, 48(5):71–82, 1991.
24. S. Vosniadou. How children learn. *Successful schooling*, 16, 2003.
25. M. A. West. *Effective Teamwork: Practical Lessons Learned from Organizational Research*. Wiley-Blackwell, West Sussex, 2012.
26. K.B. White. Mis project teams: An investigation of cognitive style implications. *MIS Quarterly*, 8(2):95–101, 1984.
27. D. J. Wilde. *Teamology: The Construction and Organization of Effective Teams*. Springer-Verlag, London, 2009.
28. D.J. Wilde. *Post-Jungian Personality Theory for Individuals and Teams*. SYDROSE LP, 2013.