Fundamenta Informaticae 174 (2020) 83–101
DOI 10.3233/FI-2020-1933
IOS Press

Solving the Team Composition Problem in a Classroom

Felip Manyà[*†]
Artificial Intelligence Research Institute (IIIA, CSIC), Bellaterra, Spain
felip@iiia.csic.es

Santiago Negrete
Universidad Autónoma Metropolitana (DCCD, Cuajimalpa), CDMX, Mexico
snegrete@correo.cua.uam.mx

Carme Roig
INS Torres i Bages, Hospitalet de Llobregat, Spain
mroig112@xtec.cat

Joan Ramon Soler
Artificial Intelligence Research Institute (IIIA, CSIC), Bellaterra, Spain
jramonsoler@iiia.csic.es

Abstract. Given a classroom containing a fixed number of students and a fixed number of tables that can be of different sizes, as well as a list of preferred classmates to sit with for each student, the team composition problem in a classroom (TCPC) is the problem of finding an assignment of students to tables in such a way that the preferences of students are maximally-satisfied. In this paper, we first formally define the TCPC, prove that it is NP-hard and define two different MaxSAT models of the problem, called maximizing and minimizing encoding. Then, we report on the results of an empirical investigation that show that solving the TCPC with MaxSAT solvers is a promising approach and provide evidence that the minimizing encoding outperforms the maximizing encoding. Finally, we illustrate how the proposed MaxSAT-based modeling approach is also well-suited for modeling other more complex team formation problems.

Keywords: Team creation, MaxSAT, NP-hard, encoding, optimization solver.

[*]Address for correspondence: Campus UAB, Carrer de Can Planas, Zona 2, 08193 Bellaterra, Spain
[†]This work was supported by the MINECO-FEDER project RASO TIN2015-71799-C2-1-P. The second author is supported by grant PSPA-PNB-CGVyDI.324.2016 from Universidad Autónoma Metropolitana (Cuajimalpa), Mexico.

1. Introduction

Given a classroom containing a fixed number of students and a fixed number of tables that can be of different sizes, as well as a list of preferred classmates to sit with for each student, the team composition problem in a classroom (TCPC) is the problem of finding an assignment of students to tables in such a way that the preferences of students are maximally-satisfied. Our motivation behind this work is to solve a problem posed by the director of studies of a secondary school in the area of Barcelona, though this problem may be found in a wide range of situations and institutions.

In this paper, we first formally define the TCPC, prove that it is NP-hard and define two different MaxSAT models of the problem, called maximizing and minimizing encoding. Next, we report on the results of an empirical investigation that show that solving the TCPC with MaxSAT solvers is a promising approach and provide evidence that the minimizing encoding outperforms the maximizing encoding. We then illustrate how the proposed MaxSAT-based modeling approach is also well-suited for modeling other more complex team formation problems. Finally, we discuss some related work.

To tackle the TCPC we use a MaxSAT-based problem solving approach, which is an active area of research in Artificial Intelligence, (see e.g. [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14] and the references therein for previous and related work). MaxSAT-based problem solving is a generic problem solving approach for optimization problems which consists on first defining a MaxSAT model for instances of the problem to be solved, and then derive solutions to the encoded instances of the problem using an off-the-shelf MaxSAT solver. By a MaxSAT model we mean a representation of the problem using the language of Boolean propositional logic. It is a declarative approach: we only need to define a model and from that model an optimal solution is automatically derived. Furthermore, the method is highly efficient because we may take advantage of the extremely efficient MaxSAT solvers which are publicly available.

It is commonly assumed that designing an algorithm to work directly on the original problem encoding should outperform approaches that require a translation via a generic intermediate formalism, such as a CSP, SAT or MaxSAT. However, this line of reasoning ignores the fact that generic solvers can benefit from many years of development by a broad research community. It is not easy to replicate this kind of effort in other domains.

In the present formulation of the problem, we consider the preferences of the students. Nevertheless, our approach could also be easily adapted to take into account other factors that can be relevant to the performance of a team such as personality, expertise, competence, competitiveness and human formation [15, 16]. To illustrate this point, we describe how the Synergistic Team Composition Model (STCM) [17], can be mapped into our framework.

This paper extends the results of [18]. The new contributions are the definition of the minimizing encoding, experiments with maximally-satisfied instances (only fully-satisfied were considered in [18]), and an empirical comparison of the maximizing and minimizing encodings that provides evidence that the minimizing encoding outperforms the maximizing encoding. Furthermore, we describe a MaxSAT encoding of the STCM problem and add a related work section.

The rest of the paper is organized as follows: Section 2 defines the TCPC formally and proves that it is NP-hard. Section 3 gives some background on MaxSAT. Section 4 defines the maximizing and minimizing MaxSAT encodings of the TCPC. Section 5 reports on the empirical investigation conducted. Section 6 illustrates how to solve the STCM problem with the proposed approach. Section 7 discuss some related work. Section 8 gives some conclusions and future work.

2. The team composition problem in a classroom

Depending on the activity to be performed in a classroom at a given moment, the distribution of the students may need to be different. In the general case, we consider there is a fixed number of students and there is a list of preferred classmates to sit with for each student. Then, the goal is to partition students into teams, which may have different sizes, in such a way that the preferences of the students are maximally-satisfied.

The version of the TCPC that we use as a case study in this paper has the following constraints:

- The classroom has n students.
- The classroom has tables of 2 and 3 students with a combined capacity for n students.
- Each student has provided a list of classmates she would prefer to sit with.

The objective is to find an assignment of students to tables such that the preferences of students are maximally-satisfied. Notice that the first two constraints are hard whereas the last one is soft. We will say that a solution is *fully-satisfied* if, and only if, all the students in the same table have the rest of the students of the table in their list of preferences. We will say that a solution is *maximally-satisfied* if, and only if, the number of students who have their preferences satisfied is maximized. Note that a fully-satisfied solution is also a maximally-satisfied solution.

Proposition 2.1. Given n students, a classroom that has tables of 2 and 3 students with a combined capacity for n students, and a list of preferred classmates to sit with for each student, the problem of deciding if there is a fully-satisfied solution is NP-complete.

Proof:
This problem belongs to NP: we can check, in polynomial time, wether or not an assignment of students to tables is a fully-satisfied solution by inspecting the lists of preferences of the students.

We now prove that this problem is NP-hard by reducing the problem of *partitioning a graph into triangles* (PIT problem) to it. Given a graph $G = (V, E)$, where V is the set of vertices and E is the set of edges, that verifies that $|V| = 3q$ for some integer q, the partition of V into triangles consists on finding a partition of V formed by $V_1, \ldots, V_q$, each containing exactly 3 vertices, such that for each $V_i = \{u_i, v_i.w_i\}$, $1 \leq i \leq q$, the edges $\{u_i, v_i\}$, $\{u_i, w_i\}$ and $\{v_i, w_i\}$ belong to E. This problem is NP-complete [19].

That problem can be reduced to an instance of our problem without loss of generality by considering a classroom with $3q$ students, 0 tables of 2 and q tables of 3. For each edge $\{u, v\}$ on graph V, establish a preference of student u for student v and a preference of student v for student u. Note that this reduction takes polynomial time. Then, the problem of partitioning the vertices of a graph into triangles has a solution if, and only if, all the students in the classroom can be sat in such a way that all the preferences of students are fully-satisfied. $\square$

Corollary 2.2. The TCPC is NP-hard.

Proof:
This follows from the fact that every fully-satisfied solution ia also a maximally-satisfied solution. □

We can find a fully-satisfied solution with a decision algorithm but need an optimization algorithm to find a maximally-satisfied solution. Indeed, finding a maximally-satisfied solution is in general harder than finding a fully-satisfied solution. For example, if we assume that there are just tables of 2 students, finding a fully-satisfied solution can be solved in polynomial time but finding a maximally-satisfied solution remains NP-hard.

3. The MaxSAT problem

We assume readers have some familiarity with basic concepts of Boolean propositional logic. The most well-know problem of propositional logic is SAT: given a formula ϕ in Conjunctive Normal Form (CNF), decide whether there is a truth assignment that satisfies ϕ.

Reminder: a literal is a propositional variable or a negated propositional variable, a clause is a disjunction of literals, a CNF formula is a conjunction of clauses, and a truth assignment is a mapping that assigns 0 (false) or 1 (true) to each propositional variable. A CNF is satisfied by an assignment if it is true under the usual truth-functional interpretation of $\vee$ and $\wedge$ and the truth values assigned to the variables.

An optimization variant of SAT is MaxSAT: given a CNF formula ϕ, MaxSAT is to find a truth assignment that maximizes the number of satisfied clauses of ϕ. However, in this paper we use the term MaxSAT in a broad sense: we allow to distinguish between hard and soft clauses, and allow to associate a weight to soft clauses (formally, hard clauses have an infinite weight). This more general formulation of MaxSAT is technically known as weighted partial MaxSAT [10], which is formally defined in the remaining of this section.

We start by defining a more general notion of clause. A weighted clause is a pair (c, w), where c is a clause and w, its weight, is a positive integer or infinity. A clause is hard if its weight is infinity; otherwise it is soft.

A weighted partial MaxSAT instance is a multiset of weighted clauses

$$\phi = \{(h_1, \infty), \ldots, (h_k, \infty), (c_1, w_1), \ldots, (c_m, w_m)\},$$

where the first k clauses are hard and the last m clauses are soft. For simplicity, in what follows, we omit infinity weights, and write $\phi = \{h_1, \ldots, h_k, (c_1, w_1), \ldots, (c_m, w_m)\}$. A soft clause (c, w) is equivalent to having w copies of the clause $(c, 1)$, and $\{(c, w_1), (c, w_2)\}$ is equivalent to $(c, w_1 + w_2)$.

Weighted partial MaxSAT for an instance ϕ is the problem of finding an assignment that satisfies all the hard clauses and minimizes the sum of the weights of the falsified soft clauses; such an assignment is called optimal assignment of ϕ. It can also be defined as the problem of finding an assignment that satisfies all the hard clauses and maximizes the sum of the weights of the satisfied soft clauses. Both definitions are equivalent and relevant for the TCPC because in the next section we define one MaxSAT encoding based on the first option and another based on the second option.

4. MaxSAT encodings for the TCPC

We present two different ways of encoding the TCPC in the weighted partial MaxSAT formalism. In the first approach, the objective is to maximize the quality of the solution and we refer to it as the maximizing encoding. In the second approach, the objective is to minimize the quality loss and we refer to it as the minimizing encoding.

4.1. The maximizing encoding

We first present how the TCPC can be represented as a weighted partial MaxSAT instance using the maximizing encoding. To illustrate how to model the problem, we will consider that the classroom has 28 students and there are 8 tables of 2 students and 4 tables of 3 students. This is a typical classroom distribution in some secondary schools.

First of all, we define the set of Boolean variables of our encoding:

$$\{x_{ij} | 1 \leq i < j \leq 28\} \cup \{x_{ijk} | 1 \leq i < j < k \leq 28\} \cup \{y_i | 1 \leq i \leq 28\}$$

These variables have the following intended meaning: x_{ij} is true iff students i and j sit together in a table of 2; x_{ijk} is true iff students i, j and k sit together in a table of 3; and y_i is true if student i sits in a table of 2 and is false if student i sits in a table of 3.

Using the previous Boolean variables, we create a Weighted Partial MaxSAT instance that encodes the constraints of the problem. The proposed encoding has the following hard clauses:

1. For each student i, where $1 \leq i \leq 28$, the encoding contains a set of hard clauses that encode the following cardinality constraint:

   (a) If $i = 1$, then
   $$\sum_{j=2}^{28} x_{1j} + \sum_{j=2}^{27} \sum_{k=j+1}^{28} x_{1jk} = 1$$

   (b) If $2 \leq i \leq 27$, then
   $$\sum_{j=1}^{i-1} x_{ji} + \sum_{j=i+1}^{28} x_{ij} + \sum_{k=2}^{i-1} \sum_{j=1}^{k-1} x_{jki} + \sum_{j=1}^{i-1} \sum_{k=i+1}^{28} x_{jik} + \sum_{j=i+1}^{27} \sum_{k=j+1}^{28} x_{ijk} = 1$$

   (c) If $i = 28$, then
   $$\sum_{j=1}^{27} x_{j28} + \sum_{j=1}^{26} \sum_{k=j+1}^{27} x_{jk28} = 1$$

   This cardinality constraint states that student i sits exactly in one table, and the table is either of 2 or of 3.

2. For each variable x_{ij}, the encoding contains the hard clauses $\neg x_{ij} \vee y_i$ and $\neg x_{ij} \vee y_j$. Note that $(\neg x_{ij} \vee y_i) \wedge (\neg x_{ij} \vee y_j)$ is equivalent to $x_{ij} \rightarrow y_i \wedge y_j$. This clause states that if x_{ij} is true, then students i and j sit in a table of 2.

3. For each variable x_{ijk}, the encoding contains the hard clauses $\neg x_{ijk} \vee \neg y_i$, $\neg x_{ijk} \vee \neg y_j$ and $\neg x_{ijk} \vee \neg y_k$. Note that $(\neg x_{ijk} \vee \neg y_i) \wedge (\neg x_{ijk} \vee \neg y_j) \wedge (\neg x_{ijk} \vee \neg y_k)$ is equivalent to $x_{ijk} \rightarrow \neg y_i \wedge \neg y_j \wedge \neg y_k$. This clause states that if x_{ijk} is true, then students i, j and k sit in a table of 3.

4. The encoding contains a a set of hard clauses that encode the following cardinality constraints: $\sum_{i=1}^{28} y_i = 16$ and $\sum_{i=1}^{28} \neg y_i = 12$. These cardinality constraints state that there are 16 students sitting in tables of 2 and 12 students sitting in tables of 3.

   In practice, it is sufficient to add either the constraint $\sum_{i=1}^{28} y_i = 16$ or the constraint $\sum_{i=1}^{28} \neg y_i = 12$ because if there are exactly 16 (12) variables y_i, $1 \leq i \leq 28$, that evaluate to true (false), then the remaining 12 (16) variables must evaluate to false.

The encoding of a cardinality constraint of the form $x_1 + \ldots + x_n = k$ has $\mathcal{O}(n)$ clauses if one uses the encoding based on counters and defined in [20]. Other efficient encodings of cardinality constraints are described and analyzed in [21, 22]. In our empirical investigation, we encode the previous cardinality constraints using PBLib[1], which is a C++ tool for efficiently encoding pseudo-Boolean constraints to CNF.

Since we considered two sizes of tables, we just need one variable y_i for each student. If we consider n different sizes, then we need $\lceil \log_2 n \rceil$ variables for each student. For example, for four different sizes, we need two variables (y_i, y_i') and each size is represented by one of the following conjunctions: $y_i \wedge y_i'$, $\neg y_i \wedge y_i'$, $y_i \wedge \neg y_i'$ and $\neg y_i \wedge \neg y_i'$.

The soft clauses of our encoding are the following weighted unit clauses:

1. For each variable x_{ij}, $1 \leq i < j \leq 28$, the encoding contains the weighted unit clause (x_{ij}, w_{ij}).

2. For each variable x_{ijk}, $1 \leq i < j < k \leq 28$, the encoding contains the weighted unit clause (x_{ijk}, w_{ijk}).

A key aspect of our encoding is how weights are assigned to the variables of the form x_{ij} and x_{ijk}. First of all, we build a directed graph $G = (V, E)$, where V contains a vertex i for each student i in the classroom, and E contains an edge (i, j) if student i wants to sit with student j. The weight associated with each student i in G, denoted by $w(i)$, is the out-degree of the vertex i of G.[2] The weight associated with the variable x_{ij}, denoted by w_{ij}, is $2(w(i) \times w(j))$, where $w(i)$ and $w(j)$ are the weights associated with vertices i and j, respectively, in the subgraph of G induced by the set of vertices $\{i, j\}$ (i.e.; the weight of student i and j in $G(\{i, j\})$). The weight associated with the variable x_{ijk}, denoted by w_{ijk}, is $3(w(i) \times w(j) \times w(k)/8)$, where $w(i)$, $w(j)$ and $w(k)$ are the weights associated with vertices i, j and k, respectively, in $G(\{i, j, k\})$. The value of $w(i) \times w(j)$ ranges from 0 to 1 and the value of $w(i) \times w(j) \times w(k)$ ranges from 0 to 8. This explains the fact that $w(i) \times w(j) \times w(k)$ is divided by 8. Moreover, we multiply the weights by 2 in the tables of 2 and by 3 in the tables of 3. In this way, we maximize the number of satisfied students. Note that if the weight assigned to x_{ij} is 2, there are 2 satisfied students if they sit together in a table of 2, whereas if

[1]http://tools.computational-logic.org/content/pblib.php
[2]The out-degree of a vertex is the number of edges going out of a vertex in a directed graph.

the weight assigned to x_{ijk} is 3, there are 3 satisfied students if they sit together in a table of 3. The weight w_{ij} (w_{ijk}) associated with a table of 2 (3) indicates the quality of the assignment of students i and j (i, j and k) to a table of 2 (3): the bigger the weight, the better the assignment of students to tables.[3]

In the previous encoding, if the weight associated with a variable is 0, then the negation of this variable is added as a unit clause in the hard part. Moreover, an optimal solution corresponds to a fully-satisfied solution if, and only if, all the satisfied soft clauses of the form (x_{ij}, w_{ij}) and (x_{ijk}, w_{ijk}) have weight 2 and 3, respectively.

For fully-satisfied instances, if we add to the hard part the negation of x_{ij} (i.e., the unit hard clause $\neg x_{ij}$) for each variable x_{ij} whose associated weight is different from 2 and the negation of x_{ijk} (i.e., the unit hard clause $\neg x_{ijk}$) for each variable x_{ijk} whose associated weight is different from 3, then we do not need to add any soft clause. Moreover, any satisfying assignment of the hard part allows us to derive a fully-satisfied solution. This case can be solved either with a SAT solver or with a MaxSAT solver fed with a MaxSAT instance that only contains hard clauses. Actually, to find a fully-satisfied solution is a decision problem.

If there is no fully-satisfied solution, the problem becomes an optimization problem and the objective is to find a solution that satisfies students as much as possible. Because of that, in the general case, we add the clauses (x_{ij}, w_{ij}) and (x_{ijk}, w_{ijk}) such that $w_{ij} \neq 0$ and $w_{ijk} \neq 0$ in the soft part of the encoding. In this way, we provide a solution that maximizes the number of satisfied students. In this case, we say that we have a maximally-satisfied solution.

An optimal solution to the TCPC is obtained from a MaxSAT optimal interpretation by assigning students i and j to the same table of 2 if, and only if, the literal x_{ij} is satisfied by the optimal interpretation; and by assigning students i, j and k to the same table of 3 if, and only if, the literal x_{ijk} is satisfied by the optimal interpretation.

If an optimal interpretation satisfies the soft clause (x_{ij}, w_{ij}), then this interpretation falsifies all the soft clauses (x_{lm}, w_{lm}) and (x_{lmn}, w_{lmn}) such that l, m or n are equal to i or j because of the cardinality constraint that states that every student sits exactly in one table. A similar situation happens when the satisfied clause is of the form (x_{ijk}, w_{ijk}), corresponding to a table of 3. Thus, the number of falsified soft clauses is usually greater than the number of satisfied soft clauses, and the maximum sum of weights of satisfied clauses indicates the maximum quality that can be reached taking into account the preferences of the students.

4.2. The minimizing encoding

The minimizing encoding focus on minimizing the quality loss instead of maximizing the quality of the solution as in the maximizing encoding. So, the challenge now is to adequately represent the notion of quality loss in the TCPC and derive a more efficient encoding.

The minimizing encoding is defined over the same set of Boolean variables and has the hard constraints of the maximizing encoding. The soft clauses are derived from the soft clauses of the maximizing encoding as follows:

[3] Since most of the MaxSAT solvers deal with weights that are positive integers, in the experiments we multiply the weights by 100 and take the integer part.

1. each soft clause (x_{ij}, w_{ij}) is replaced with the soft clause $(\neg x_{ij}, w_{max} - w_{ij})$, and

2. each soft clause (x_{ijk}, w_{ijk}) is replaced with the soft clause $(\neg x_{ijk}, w'_{max} - w_{ijk})$,

where w_{max} is the maximum weight that can be assigned to a table of 2 and w'_{max} is the maximum weight that can be assigned to a table of 3. In our encoding, $w_{max} = 2$ and $w'_{max} = 3$.

An optimal solution to the TCPC is obtained from a MaxSAT optimal interpretation by assigning students i and j to the same table of 2 if, and only if, the literal $\neg x_{ij}$ is falsified by the optimal interpretation; and by assigning students i, j and k to the same table of 3 if, and only if, the literal $\neg x_{ijk}$ is falsified by the optimal interpretation. Note that $\neg x_{ij}$ and $\neg x_{ijk}$ are falsified if, and only if, x_{ij} and x_{ijk} are satisfied. If an optimal interpretation falsifies the soft clause $(\neg x_{ij}, w'_{ij})$, then it satisfies all the soft clauses $(\neg x_{lm}, w'_{lm})$ and $(\neg x_{lmn}, w'_{lmn})$ such that l, m or n are equal to i or j because of the cardinality constraint that states that every student sits exactly in one table. A similar situation happens when the falsified clause is of the form $(\neg x_{ijk}, w'_{ijk})$.

In contrast to the maximizing encoding, the number of satisfied soft clauses in an optimal solution of the minimizing encoding is usually greater than the number of falsified soft clauses. This implies that the number of conflicts that a MaxSAT solver has to identify for finding an optimal solution is greater in the maximizing encoding than in the minimizing encoding and, as we will see in the experimental results, this has a tremendous impact on the performance of the solver.

The weight of the soft clause $(\neg x_{ij}, w_{max} - w_{ij})$ $((\neg x_{ijk}, w'_{max} - w_{ijk}))$ indicates the quality loss if students i and j (i, j and k) sit together in a table of 2 (3): the smaller the weight, the better the assignment of students to tables. In fact, the weight $w_{max} - w_{ij}$ ($w'_{max} - w_{ijk}$) is the penalty to be paid by students i and j (i, j and k) if they sit in the same table. So, the minimum sum of weights of falsified clauses indicates the minimum quality loss that can be reached taking into account the preferences of the students.

If the minimum sum of weights of falsified clauses in an optimal solution is 0, then this solution is fully-satisfied. Note that the clauses of the form $(\neg x_{ij}, 0)$ correspond to tables of 2 in which students i and j prefer to sit together, and the clauses of the form $(\neg x_{ijk}, 0)$ correspond to tables of 3 in which students i, j and k prefer to sit together. In practice, the clauses $(\neg x_{ij}, 0)$ and $(\neg x_{ijk}, 0)$ can be removed from the soft part and the encoding remains correct.

It is worth mentioning that the minimization approach proposed here can be extended to other combinatorial optimization problems. It is particularly useful when the resulting MaxSAT encoding has subsets of soft unit clauses whose literals are involved in cardinality constraints in the hard part, because it can reduce considerably the number of conflicts needed to find an optimal solution. The main difficulty of the minimizing encoding is to define a suitable weighting function that preserves the optimal solutions between the maximizing and the minimizing encodings.

Finally, it is worth mentioning that it is possible to define the previous MaxSAT encodings of the TCPC using the set of propositional variables $\{x_i^t | 1 \le i \le 28, 1 \le t \le 12\}$, where the intended meaning of x_i^t is that x_i^t is true if, and only if, student i sits at table t. However, all the experiments performed with encodings using this set of variables did not outperform the experiments performed with the encodings proposed in this section.

5. Experimental results

We carried out an experimental investigation to evaluate the proposed MaxSAT-based approach to the TCPC on both fully-satisfied and maximally-satisfied instances, and compared the performance of the maximizing and minimizing encodings on the selected instances. In the experiments, in order to analyze the scaling behavior, we considered different sizes of classrooms: the rows always have 2 tables of 2 and 1 table of 3, and the number of rows ranges from 1 to 18. So, the number of students per classroom ranges from 7 to 126. Besides, we assumed that each student gives a list of students she would like to sit with. We generated the preferences at random in such a way that we can guarantee that the generated instances have either fully-satisfied or maximally-satisfied solutions. We generated 50 different TCPC instances for each size of classroom, encoded them to weighted partial MaxSAT, and solved the resulting maximizing and minimizing encodings with the exact MaxSAT solver WPM3 [4] using a cutoff time of 900 seconds. All the experiments were performed in a 3.60GHz Intel(R) i7-4790 with 8GB RAM.

Table 1. Experimental results for fully-satisfied instances: Students: number of students; Hard: mean number of hard clauses per instance; Variables: mean number of variables per instance; Soft _Max: mean number of soft clauses per instance in the maximizing encoding; Soft _Min: mean number of soft clauses per instance in the minimizing encoding; Time_Max: mean time, in seconds, needed to solve an instance with the maximizing encoding; and Time_Min: mean time, in seconds, needed to solve an instance with the minimizing encoding. The number of solved instances, within a cutoff time of 900s, is shown in parentheses.

Students	Hard	Variables	Soft_Max	Soft_Min	Time_Max	Time_Min
7	246	117	21	11	0,01 (50)	0,01 (50)
14	1040	659	56	35	0,01 (50)	0,01 (50)
21	2594	1916	93	60	0,05 (50)	0,01 (50)
28	5214	4239	128	85	0,25 (50)	0,01 (50)
35	9127	7937	150	98	0,63 (50)	0,01 (50)
42	14934	13387	189	124	3,65 (50)	0,01 (50)
49	22772	20889	221	147	10 (50)	0,02 (50)
56	33069	30842	257	172	63 (49)	0,03 (50)
63	46079	43552	286	191	123 (43)	0,04 (50)
70	62232	59365	324	218	191 (31)	0,06 (50)
77	81833	78584	357	240	234 (25)	0,08 (50)
84	105200	101588	386	260	286 (20)	0,11 (50)
91	132775	128730	426	286	513 (10)	0,21 (50)
98	164741	160326	456	308	634 (5)	0,25 (50)
105	201629	196844	493	332	815 (2)	0,28 (50)
112	243565	238390	522	351	0 (0)	0,30 (50)
119	291154	285487	568	385	0 (0)	0,49 (50)
126	344370	338356	590	398	0 (0)	0,62 (50)

We selected the solver WPM3 because it was ranked in the first positions in the last MaxSAT Evaluations and was the best performing solver on our instances in preliminary tests. WPM3 reformulates the MaxSAT optimization problem into a sequence of SAT decision problems and introduces Pseudo-Boolean (PB) constraints to refine the lower bound after each execution of the SAT solver. In order to identify the most suitable PB constraints, WPM3 analyzes the unsatisfiable cores retrieved from the previous SAT executions.

Table 1 compares the maximizing and minimizing encodings on fully-satisfied instances. We observe that the minimizing encoding clearly outperforms the maximizing encoding: the minimizing encoding needs a mean time of less than one second to solve an instance, independently of the number of students in the classroom, but the maximizing encoding is only able to solve all the selected instances within the cutoff time if the number of students is less than or equal to 49. The maximizing encoding only solves 43, 31 and 5 instances out of 50 when the number of students is 63, 77 and 84, respectively. It was not able to solve any instance for more than 84 students.

Table 2. Experimental results for maximally-satisfied instances: Students: number of students; Hard: mean number of hard clauses per instance; Variables: mean number of variables per instance; Soft _Max: mean number of soft clauses per instance in the maximizing encoding; Soft _Min: mean number of soft clauses per instance in the minimizing encoding; Time_Max: mean time, in seconds, needed to solve an instance with the maximizing encoding; and Time_Min: mean time, in seconds, needed to solve an instance with the minimizing encoding. The number of solved instances, within a cutoff time of 900s, is shown in parentheses.

Students	Hard	Variables	Soft_Max	Soft_Min	Time_Max	Time_Min
7	225	113	18	11	0,01 (50)	0,01(50)
14	956	640	44	30	0,01 (50)	0,01(50)
21	2416	1879	67	47	0,02 (50)	0,01 (50)
28	4953	4184	91	64	0,07 (50)	0,01 (50)
35	8911	7892	118	83	0,34 (50)	0,02 (50)
42	14632	13321	145	103	1,29 (50)	0,04 (50)
49	22421	20814	170	121	2,78 (50)	0,15 (50)
56	32692	30760	203	144	15 (50)	0,59 (50)
63	45645	43457	222	159	44 (50)	0,71 (50)
70	61734	59256	252	179	83 (47)	4,92 (50)
77	81198	78445	266	190	78 (45)	8,66 (50)
84	104574	101452	296	212	97 (35)	26 (50)
91	132063	128574	323	232	182 (22)	20 (50)
98	164001	160166	350	251	182 (19)	67 (48)
105	200797	196664	374	268	292 (18)	79 (44)
112	242692	238203	396	284	281 (14)	31 (38)
119	290160	285272	426	307	236 (6)	59 (37)
126	343404	338146	452	325	422 (4)	39 (30)

Table 2 compares the maximizing and minimizing encodings on maximally-satisfied instances. We observe that the minimizing encoding scales much better than the maximizing encoding, and also needs less time to solve an instance. While the minimizing encoding is able to solve all the instances with less than 98 students, the maximizing encoding fails to solve some instances when there are 70 or more students. The difference performance profile between the maximizing and minimizing encodings is due to the number of conflicts that must be detected by WPM3 to find an optimal solution. As said above, an optimal solution of the maximizing encoding falsifies much more soft clauses and WPM3 usually has to solve a larger sequence of SAT problems in this case.

WPM3 has an incomplete version that stops the solver after a prefixed time. This option could be used to obtain good quality solutions when the complete version of WPM3 used in the experiments fails to find an optimal solution. The incomplete version often computes optimal solutions but it cannot certify that the solutions are optimal as the exact version does.

6. Reducing team formation to TCPC

We solved TCPC like a particular team formation problem that only considers the preferences of students to create teams. However, classroom team formation can involve more sophisticated and sensible criteria based, for example, on Organisational Psychology (OP). In this section, we show how a more involved OP-based problem, the Synergistic Team Composition Model (STCM) [17], can be mapped into our framework.

The dominant OP approaches to finding good teams rely on individual competences and personality traits. In the field of education, such competences refer to different types of intelligences, which can be roughly judged by teachers in order to avoid an invasive and expensive testing process. On the other hand, personality traits are usually measured by means of subjective self-assessment tests.

A Post-Jungian personality test is based on the cognitive mode model developed by the pioneering psychiatrist Carl Gustav Jung [23]. It has two pairs of complementary variables that determine psychological functions: Sensing/Intuition (SN) and Thinking/Feeling (TF); and two pairs of complementary variables that determine psychological attitudes: Perception/Judgment (PJ) and Extroversion/Introversion (EI). Psychological functions and attitudes form a four-dimension vector $\boldsymbol{p} = (EI, SN, TF, PJ) \in [-1, 1]^4$ that characterizes a personality. In [24, 25], Wilde proposes balancing the teams by incorporating individuals of different gender having diverse sensing/intuition and thinking/feeling, at least one introvert person and at least one extrovert, thinking and judging person.

For competences, we consider the Multiple Intelligences Theory [26]. In this theory, each person has a competence profile given by an eight-dimension vector $\boldsymbol{l} = \langle vl, lm, sv, bk, mu, ie, ia, na\rangle \in [0, 1]^8$, where each dimension represents a type of intelligence. We consider the following intelligences: *vl* is verbal-linguistic intelligence, *lm* is logical-mathematical intelligence, *sv* is spatial-visual intelligence, *bk* is bodily-kinesthetic intelligence, *mu* is musical intelligence, *ie* is interpersonal intelligence, *ia* is intrapersonal intelligence and *na* is naturalist intelligence.

The goal of our problem is to create teams for performing a given task taking into account the personality and competences of individuals. A task τ requires a set of competences ($c_i \in C_\tau$), where each competence has an associated weight $w_i \in [0, 1]$ that indicates its relevance for the task fulfillment and a desired level $l_i \in [0, 1]$. Furthermore, any task has a parameter $\lambda \in [0, 1]$ that balances the importance of the proficiency $u_{prof(k)}$ and congeniality $u_{con(k)}$ of team k, and is used to calculate the suitability $s(k) = \lambda \cdot u_{prof(k)} + (1-\lambda) \cdot u_{con(k)}$ of team k. Moreover, every task has a parameter $v \in [0,1]$ that balances the importance of under-competence and over-competence for the calculation of proficiency $u_{prof(k)}$, as we explain below.

A *competent student set* for a competence c_i is defined as $\delta(c_i) = \{a \in k \mid c_i \in \{l_i^a \mid l_i^a > 0\}\}$, where the zero is a typical arbitrary threshold. We define a responsibility *assignment* as a correspondence between students and required competences such that every competence is associated at least with a student in $\delta(c_i)$. We note by Θ_τ^k the set of competence assignments η for task τ and team k. The proficiency degree $\eta_{prof}(k,\tau)$ for a team k and task τ given a responsibility assignment η is one minus the sum of penalties associated to over-competence $o(\eta)$ and under-competence $u(\eta)$ of that team performing the task. We define under-competence and over-competence as follows:

$$u(\eta) = \sum_{i \in I_\tau} w_i \cdot \frac{\sum_{a \in \delta(c_i)} \mid min(l^a(c_i) - l_i, 0) \mid}{\mid \{a \in \delta(c_i) \mid l^a(c_i) - l_i \leq 0\} \mid} \quad o(\eta) = \sum_{i \in I_\tau} w_i \cdot \frac{\sum_{a \in \delta(c_i)} max(l^a(c_i) - l_i, 0)}{\mid \{a \in \delta(c_i) \mid l^a(c_i) - l_i \geq 0\} \mid}$$

Proficiency is defined as $u_{prof(k)} = \max_{\eta \in \Theta_\tau^k} (1 - (v \cdot u(\eta) + (1-v) \cdot o(\eta))$. Penalties are added because over-competence causes boredom and under-competence causes frustration. We note that students who are not responsible for a given competence for the task can remain free of paying penalties for that competence. Competence assignments can have different properties. In the education case, we are interested in *inclusive* assignments, which are the ones where each team member is responsible of at least one competence for the task.

Congeniality is defined as $u_{con}(k) = u_{SNTF}(k) + u_{ETJ}(k) + u_I(k) + u_{gender}(k)$, where:

- $u_{SNTF}(k) = \sigma_{SN}(k) \cdot \sigma_{TF}(k)$, the product of standard deviation for SN and TF personality components of the members of team k.
- $u_{ETJ}(k) = max_{a \in k^{ETJ}}[max(\boldsymbol{(0, \alpha, \alpha, \alpha) \cdot p}, 0), 0]$, where $\boldsymbol{\alpha \approx 0.5287/3}$ and $k^{ETJ} = \{a \in k \mid tf^a > 0, ei^a > 0, pj^a > 0\}$. It is the importance of having a strong ETJ individual.
- $u_I(k) = max_{a \in k^I}[max(\boldsymbol{(0, 0, -\beta, 0) \cdot p}, 0), 0]$, where $\boldsymbol{\beta = 3 \cdot \alpha = 0.5287}$ and $k^I = \{a \in k \mid ei^a \leq 0\}$. This is the importance of having a strong I (introvert) individual.
- $u_{gender}(k) = \gamma \cdot sin\,((\pi \cdot w(k))/(w(k) + m(k)))$ where $\gamma = 0.1$, $w(k)$ stands for the women number of women and $m(k)$ for the number of men in team k. This is the importance of having a satisfactory gender balance.

Example 6.1. Assume that we want to solve a task τ with two competences, $(c_1, l_1 = 0.8, w_1 = 0.5)$ and $(c_2, l_2 = 0.6, w_2 = 0.5)$, and an under-proficiency penalty of $v = 0.6$. In fact, the competences are the set of intelligences needed for task τ. We want to find the *inclusive* assignment maximizing $s(k) = \lambda \cdot u_{prof(k)} + (1-\lambda) \cdot u_{con(k)}$, where we consider $\lambda = 0.5$. In order to evaluate the suitability of a three-student team $k = \{S_1, S_2, S_3\}$, we have:

- $\langle \boldsymbol{S_1}, woman, \langle p(sn) = 0.4,\ p(tf) = -0.4,\ p(ei) = 0.5,\ p(pj) = -0.7 \rangle, [l(c_1) = 0.9, l(c_2) = 0.5] \rangle$

- $\langle \boldsymbol{S_2}, man, \langle p(sn) = -0.7,\ p(tf) = 0.6,\ p(ei) = 0.8,\ p(pj) = 0.4\rangle, [l(c_1) = 0.2, l(c_2) = 0.8]\rangle$
- $\langle \boldsymbol{S_3}, man, \langle p(sn) = 0.8,\ p(tf) = -0.7,\ p(ei) = -0.4,\ p(pj) = -0.6\rangle, [l(c_1) = 0.4, l(c_2) = 0.6]\rangle$

We want to assign students to task competences so that (1) each student is responsible for at least one competence (*inclusive*), (2) each competence is covered by at least one student (*assignment*), and (3) The proficiency degree $\eta_{prof}(k, \tau)$ for a team k and task τ given the assignment η is maximal in Θ_τ^k. For this example, we consider individual competences like an atomic task.

Table 3 shows every valid student assignment for both competences as well as its under-proficiency and over-proficiency penalty sum with $v = 0.6$. An assignment for both competences is not inclusive if some student has no competence assigned. The maximization of $u_{prof(k)} = \max_{\eta \in \Theta_\tau^k} (1 - (v \cdot u(\eta) + (1 - v) \cdot o(\eta))$ involves the minimization of the under-proficiency and over-proficiency penalty sum among the assignments. Table 4 shows every valid assignment $\eta(k, \tau)$ and, for the inclusive ones, the $cost(\eta(k, \tau)) = \sum_i cost(\eta(k, c_i))$, or total cost for the assignment.

Table 3. For each competence c_i and student assignment $\eta(k, c_i)$, under/over-proficiency costs are calculated and added. Bold lines are base cases where only one student is responsible; the rest of lines are calculated from these base lines. Only valid assignments are shown, excluding the case $S_1 = S_2 = S_3 = 0$.

i	$\eta(k, c_i)$			$u(\eta(k, c_i))$	$o(\eta(k, c_i))$	$cost(\eta(k, c_i))$
	S_1	S_2	S_3			
1	**0**	**0**	**1**	**0,12**	**0**	**0,12**
1	**0**	**1**	**0**	**0,18**	**0**	**0,18**
1	0	1	1	0,15	0	0,15
1	**1**	**0**	**0**	**0**	**0,02**	**0,02**
1	1	0	1	0,12	0,02	0,14
1	1	1	0	0,18	0,02	0,2
1	1	1	1	0,15	0,02	0,17
2	**0**	**0**	**1**	**0**	**0**	**0**
2	**0**	**1**	**0**	**0**	**0,04**	**0,04**
2	0	1	1	0	0,04	0,04
2	**1**	**0**	**0**	**0,03**	**0**	**0,03**
2	1	0	1	0,03	0	0,03
2	1	1	0	0,03	0,04	0,07
2	1	1	1	0,03	0,04	0,07

The former problem involving minimization of costs among assignments can be efficiently solved using the minimum cost flow model [27]. The minimum cost flow problem has a time complexity of $O(m \cdot log(n) \cdot (m + n \cdot log(n)))$ on a network with n nodes and m arcs [28], where $n = |k| + |I|$ (team size and competences number in task τ) and $m = \sum_i |\delta(c_i)|$. Furthermore, this problem of cost minimization among assignments can be avoided if we consider as valid just the assignments where every student is responsible of all the task competences.

Table 4. Valid assignments $\eta(k,\tau)$ from $\eta(k,c_1)$ and $\eta(k,c_2)$ showing costs $u(\eta)+o(\eta)$. ~~Inc~~ stands for not inclusive. $\boldsymbol{u_{prof}(k,\tau)=0.94}$ $(1-0.06)$.

	$\boldsymbol{\eta(k,c_1)}$						
$\boldsymbol{\eta(k,c_2)}$	1	2	3	4	5	6	7
1	~~Inc~~	~~Inc~~	~~Inc~~	~~Inc~~	~~Inc~~	0.2	0.17
2	~~Inc~~	~~Inc~~	~~Inc~~	~~Inc~~	0.18	~~Inc~~	0.21
3	~~Inc~~	~~Inc~~	~~Inc~~	**0.06**	0.18	0.24	0.21
4	~~Inc~~	~~Inc~~	0.18	~~Inc~~	~~Inc~~	~~Inc~~	0.2
5	~~Inc~~	0.21	0.18	~~Inc~~	~~Inc~~	0.23	0.2
6	0.19	~~Inc~~	0.22	~~Inc~~	0.21	~~Inc~~	0.24
7	0.19	0.25	0.22	0.09	0.21	0.27	0.24

We calculate now *congeniality* $u_{con}(k)=u_{SNTF}(k)+u_{ETJ}(k)+u_I(k)+u_{gender}(k)$, where:

- $u_{SNTF}(k)=\sigma_{SN}(k)\cdot\sigma_{TF}(k)\approx 0.7767\cdot 0.6807\approx \mathbf{0.5287}$.
- $u_{ETJ}(k)=max_{a\in k^{ETJ}}[max(\boldsymbol{(0,\alpha,\alpha,\alpha)\cdot p},0),0]$, where $\boldsymbol{\alpha\approx 0.1762}$ and $k^{ETJ}=\{S_2\}$. Calculating we get $u_{ETJ}(k)=\mathbf{0.3172}$.
- $u_I(k)=max_{a\in k^I}[max(\boldsymbol{(0,0,-\beta,0)\cdot p},0),0]$, where $\boldsymbol{\beta=0.5287}$ and $k^I=\{S_3\}$. Thus, $u_I(k)=\mathbf{0.2115}$.
- $u_{gender}(k)=\gamma\cdot sin\left((\pi\cdot w(k))/(w(k)+m(k))\right)$ where $\gamma=0.1$, $w(k)$ stands for the number of women and $m(k)$ for the number of men in team k. Thus, $u_{gender}(k)=0.1\cdot sin\left(\frac{\pi}{3}\right)\approx$ $\mathbf{0.0183}$.

We calculate $u_{con}(k)\approx 0.5287+0.3172+0.2115+0.0183\approx 1.0757$ and $s(k)\approx(0.5\cdot 0.94)+(0.5\cdot 1.0757)\approx 1,00785$. We now multiply this number by 1000 and round to the nearest integer as it is needed by the majority of optimizers. Then we will use a final desirability degree $\boldsymbol{s(k)=1008}$. In summary, we can use the same encoding of TCPC, adjusting the size of teams and number of members and replacing the weights in soft clauses in such a way that every possible team has as weight its desirability degree. If we had teams of different sizes we should scale $s(k)$ as we did in the TCPC case.

7. Related work

We can find similarities between TCPC and a classical matching theory problem known as the Stable Roommate Problem (SRP). SRP is about finding a stable matching for an even-sized set. A matching is a partition of the set into disjoint pairs of roommates. We say a matching is stable if there are not two individuals who are not roommates and both prefer each other to their roommate under the current matching. TCPC and SRP try to match elements within a single set in groups of a given size (size two for SRP).

Irving [29] described an algorithm to solve SRP with a time complexity of $O(n^2)$. Nevertheless, this algorithm solves a decision problem; it determines whether a stable matching exists, and if so, it returns that matching. We are not solving a decision problem but an optimization one. Furthermore, our problem is about matching elements in groups of any given size or a combination of sizes. Standard SRP uses rooms of size two but this problem becomes NP-complete for rooms of size three [30]. The NP-completeness proof uses the partition into triangles problem as in the NP-completeness proof of TCPC.

Note that TCPC is about finding an optimal assignment given a certain criterion, but matching theory algorithms deal with the notion of stability. An optimization version of SRP has not always an optimal solution among the stable solutions, so that they can miss optimality. We show below an example.

Given a totally ordered preference list of possible mates for each student, the desirability of A to be with B in a team k is calculated as the size for k minus the position in that list, starting by an index equal to one. Thus, the last position in the list has a desirability of zero. We show this information in the graph of Figure 1 for just four students.

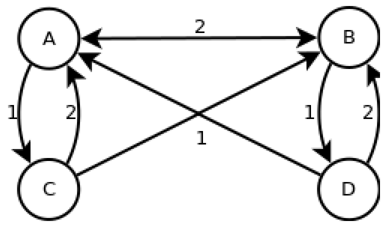

Figure 1. Degree of individual convenience of students in a team.

Table 5 shows desirability degrees for every possible size-two partition of the graph. Partition suitability is calculated as the sum of arities for each node into the subgraphs for each partition. We observe, for a standard SRP and a usual team suitability degree calculation, by summing satisfactions, that neither stability implies optimality nor optimality implies stability.

Table 5. Optimality and stability of size-two partitions of graph in Figure 1.

Partition	Desirability Calculation	Optimal	Stable
DA , BC	$(1+0)+(1+1)=\mathbf{3}$	✗	✗
DB , AC	$(2+0)+(1+2)=\mathbf{5}$	✓	✗
DC , AB	$(0+0)+(2+2)=\mathbf{4}$	✗	✓

There are criteria for the evaluation of team suitability that are different from the ones described in the previous section for STCM. Another psychological theory, proposed by Belbin [31], insists on the importance of roles in team composition processes [32]. Belbin exposes nine important roles that an individual can play in a team: plant, resource investigator, coordinator, shaper, monitor evaluator, implementer, team-worker, specialist and completer-finisher. According to Belbin's theory, people play such roles with three different performances: preferred team roles, manageable roles and least preferred roles. An effective team needs to be role balanced, having at least one individual playing any role with a given minimum performance, which can be preferred or manageable. Based exclusively on this theory, Alberola et al. [15] also pose this problem as a multi-agent coalition structure generation problem, developing a tool for practical use in an educational environment [33]. This tool uses Bayesian learning to estimate the predominant roles for each student from the peer-evaluation history made by their former teammates.

8. Concluding remarks

We have proposed two different ways of encoding the TCPC as a weighted partial MaxSAT problem, proved its NP-hardness, and carried out experiments to evaluate our approach using an exact MaxSAT solver. The results show that the minimizing encoding outperforms the maximizing encoding, and our method is useful because it does not need a dedicated algorithm; it is declarative, hence all stakeholders can be involved and understand the way the problem is specified; it is flexible because different classroom configurations can be solved with it; and it is efficient because it provides an optimal solution in a reasonable amount of time. It is also remarkable that the idea of creating a minimizing MaxSAT encoding from a maximizing MaxSAT encoding is new and can be applied to encode a wide range of optimization problems to MaxSAT.

In the future, we plan to model the problem using MinSAT [34, 35] instead of MaxSAT, and explore the possibility of using our method to encode similar team composition problems. In practice, our method could be combined with profiling techniques [36] to solve the group formation problem in *Computer Supported Collaborative Learning* applications. Our contributions could also be applied to other projects which have taken a different approach to solve related problems using other AI techniques (see [15, 16, 33] and the references therein for further details).

Acknowledgements

This work was supported by the Generalitat de Catalunya grant SGR 172, and the MINECO-FEDER project RASO TIN2015-71799-C2-1-P. The second author is supported by grant PSPA-PNB-CGVyDI.324.2016 from Universidad Autónoma Metropolitana (Cuajimalpa), Mexico.

References

[1] Abramé A, Habet D. On the Resiliency of Unit Propagation to Max-Resolution. In: Proceedings of the 24th International Joint Conference on Artificial Intelligence, IJCAI-2015, Buenos Aires, Argentina. 2015 pp. 268–274.

[2] Alsinet T, Manyà F, Planes J. A Max-SAT Solver with Lazy Data Structures. In: Proceedings of the 9th Ibero-American Conference on Artificial Intelligence, IBERAMIA 2004, Puebla, México. Springer LNCS 3315, 2004 pp. 334–342. doi:10.1007/978-3-540-30498-2_34.

[3] Alsinet T, Manyà F, Planes J. An efficient solver for Weighted Max-SAT. *Journal of Global Optimization*, 2008. **41**:61–73. doi:10.1007/s10898-007-9166-9.

[4] Ansótegui C, Didier F, Gabàs J. Exploiting the Structure of Unsatisfiable Cores in MaxSAT. In: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI, Buenos Aires, Argentina. 2015 pp. 283–289. ISBN:978-1-57735-738-4.

[5] Ansótegui C, Gabàs J, Malitsky Y, Sellmann M. MaxSAT by improved instance-specific algorithm configuration. *Artificial Intelligence*, 2016. **235**:26–39. URL `https://doi.org/10.1016/j.artint.2015.12.006`.

[6] Argelich J, Li CM, Manyà F, Planes J. The First and Second Max-SAT Evaluations. *Journal on Satisfiability, Boolean Modeling and Computation*, 2008. **4**:251–278.
`f88bc073bc50ab748bea3077a320a8a36133.pdf`

[7] Argelich J, Li CM, Manyà F, Planes J. Analyzing the Instances of the MaxSAT Evaluation. In: Proceedings of the 14th International Conference on Theory and Applications of Satisfiability Testing, SAT-2011, Ann Arbor, MI, USA. Springer LNCS 6695, 2011 pp. 360–361. doi:10.1007/978-3-642-21581-0_29.

[8] Argelich J, Manyà F. Exact Max-SAT Solvers for Over-Constrained Problems. *Journal of Heuristics*, 2006. **12**(4–5):375–392. doi:10.1007/s10732-006-7234-9.

[9] Bonet ML, Levy J, Manyà F. A Complete Calculus for Max-SAT. In: Proceedings of the 9th International Conference on Theory and Applications of Satisfiability Testing, SAT-2006, Seattle, USA. Springer LNCS 4121, 2006 pp. 240–251. doi:10.1007/11814948_24.

[10] Li CM, Manyà F. MaxSAT, Hard and Soft Constraints. In: Biere A, van Maaren H, Walsh T (eds.), Handbook of Satisfiability, pp. 613–631. IOS Press, 2009. doi:10.3233/978-1-58603-929-5-613.

[11] Li CM, Manyà F, Planes J. New Inference Rules for Max-SAT. *Journal of Artificial Intelligence Research*, 2007. **30**:321–359. doi:10.1613/jair.2215.

[12] Li CM, Manyà F, Soler JR. A Clause Tableaux Calculus for MaxSAT. In: Proceedings of the 25th International Joint Conference on Artificial Intelligence, IJCAI-2016, New York City, NY, USA. 2016 pp. 766–772. ISBN:978-1-57735-770-4.

[13] Martins R, Joshi S, Manquinho VM, Lynce I. Incremental Cardinality Constraints for MaxSAT. In: Principles and Practice of Constraint Programming - 20th International Conference, CP, Lyon, France. 2014 pp. 531–548. doi:10.1007/978-3-319-10428-7_39.

[14] Morgado A, Heras F, Liffiton MH, Planes J, Marques-Silva J. Iterative and core-guided MaxSAT solving: A survey and assessment. *Constraints*, 2013. **18**(4):478–534. doi:10.1007/s10601-013-9146-2.

[15] Alberola JM, del Val Noguera E, Sánchez-Anguix V, Julián V. Simulating a Collective Intelligence Approach to Student Team Formation. In: Hybrid Artificial Intelligent Systems - 8th International Conference, HAIS 2013, Salamanca, Spain. Proceedings. 2013 pp. 161–170. doi:10.1007/978-3-642-40846-5_17.

[16] Andrejczuk E, Rodríguez-Aguilar JA, Roig C, Sierra C. Synergistic Team Composition. In: Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS, São Paulo, Brazil. 2017 pp. 1463–1465.

[17] Andrejczuk E, Rodríguez-Aguilar JA, Roig C, Sierra C. Synergistic Team Composition. In: Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS '17. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2017 pp. 1463–1465. URL `http://dl.acm.org/citation.cfm?id=3091125.3091330`.

[18] Manyà F, Negrete S, Roig C, Soler JR. A MaxSAT-Based Approach to the Team Composition Problem in a Classroom. In: Autonomous Agents and Multiagent Systems - AAMAS 2017 Workshops, Visionary Papers, São Paulo, Brazil, Revised Selected Papers. Springer LNCS 10643, 2017 pp. 164–173. doi:10.1007/978-3-319-71679-4_11.

[19] Garey MR, Johnson DS. Computers and Intractability: A Guide to the Theory of NP-completeness. Freeman, San Francisco, 1979. ISBN-10:0716710455, 13:978-0716710455.

[20] Sinz C. Towards an Optimal CNF Encoding of Boolean Cardinality Constraints. In: Proceedings of the 11th International Conference on Principles and Practice of Constraint Programming, CP-2005, Sitges, Spain. Springer LNCS 3709, 2005 pp. 827–831. doi:10.1007/11564751_73.

[21] Abío I, Nieuwenhuis R, Oliveras A, Rodríguez-Carbonell E. A Parametric Approach for Smaller and Better Encodings of Cardinality Constraints. In: Proceedings of the 19th International Conference on Principles and Practice of Constraint Programming, CP, Uppsala, Sweden. 2013 pp. 80–96. doi:10.1007/978-3-642-40627-0_9.

[22] Bofill M, Coll J, Suy J, Villaret M. SAT Encodings of Pseudo-Boolean Constraints with At-Most-One Relations. In: Proceedings of the 16th International Conference, CPAIOR 2019, Thessaloniki, Greece. Springer LNCS 11494, 2019 pp. 112–128. doi:10.1007/978-3-030-19212-9_8.

[23] Jung C. Psychological types. Princeton University Press. *Princeton*, 1971.

[24] Wilde D. Teamology: The Construction and Organization of Effective Teams. *Springer-Verlag, London*, 2009. ISBN: 978-1-84800-386-6.

[25] Wilde D. Post-Jungian Personality Theory for Individuals and Teams. *SYDROSE LP*, 2013. ISBN-061594454X, 9780615944548.

[26] Gardner H, Hatch T. Educational implications of the theory of multiple intelligences. *Educational researcher*, 1989. **18**(8):4–10. URL `https://doi.org/10.3102/0013189X018008004`.

[27] Ahuja R, Magnanti T, Orlin J. Network flows: theory, algorithms, and applications. *Prentice hall*, 1993. ISBN: 0-13-617549-X.

[28] Orlin JB. A faster strongly polynomial minimum cost flow algorithm. *Operations research*, 1993. **41**(2):338–350. URL `https://doi.org/10.1287/opre.41.2.338`.

[29] Irving RW. An efficient algorithm for the "stable roommates" problem. *Journal of Algorithms*, 1985. **6**(4):577–595. URL `https://doi.org/10.1016/0196-6774(85)90033-1`.

[30] Iwama K, Miyazaki S, Okamoto K. Stable roommates problem with triple rooms. In: Proceedings of the 10th Korea-Japan Joint Workshop on Algorithms and Computation. 2007 pp. 105–112.

[31] Belbin R. Team Roles at Work: A Strategy for Human Resource Management. *Butterworth-Heinemann, Oxford*, 1993.

[32] Aritzeta A, Swailes S, Senior B. Belbin's team role model: Development, validity and applications for team building. *Journal of Management Studies*, 2007. **44**(1):96–118. URL `https://doi.org/10.1111/j.1467-6486.2007.00666.x`.

[33] Alberola JM, del Val E, Sánchez-Anguix V, Palomares A, Teruel MD. An artificial intelligence tool for heterogeneous team formation in the classroom. *Knowl.-Based Syst.*, 2016. **101**:1–14. doi:10.1016/j.knosys.2016.02.010.

[34] Li CM, Zhu Z, Manyà F, Simon L. Minimum Satisfiability and Its Applications. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, IJCAI-2011, Barcelona, Spain. 2011 pp. 605–610. doi:10.5591/978-1-57735-516-8/IJCAI11-108.

[35] Li CM, Zhu Z, Manyà F, Simon L. Optimizing with Minimum Satisfiability. *Artificial Intelligence*, 2012. **190**:32–44. URL `https://doi.org/10.1016/j.artint.2012.05.004`.

[36] Costaguta R. Algorithms and Machine Learning Techniques in Collaborative Group Formation. In: Advances in Artificial Intelligence and Its Applications - 14th Mexican International Conference on Artificial Intelligence, MICAI 2015, Cuernavaca, Morelos, Mexico. Proceedings, Part II. 2015 pp. 249–258. doi:10.1007/978-3-319-27101-9_18.