A CONSTRAINT-BASED APPROACH FOR CLASSES SETTING-UP PROBLEMS IN SECONDARY SCHOOLS

Nogareda, A.-M.[*] & Camacho, D.[**]

[*] Ecole Hôtelière de Lausanne, HES-SO, University of Applied Sciences Western Switzerland

[**] Escuela Politécnica Superior, Universidad Autónoma de Madrid, Spain

E-Mail: ana-maria.nogareda@ehl.ch, david.camacho@uam.es

Abstract

In this paper, we present the problem of allocating students to classes in Swiss secondary schools, where students have different profiles due to their level in some fields or to the options they attend. The pedagogical objective is to have a high diversity of profiles within a class and similarity between classes. In order to achieve this goal, the problem is modelled as a resource allocation problem (RAP), where students are resources, using a constraint satisfaction optimisation approach (CSOP). The RAP is then solved in two different ways, with a solver for CSOP, and with an ant colony optimisation algorithm (ACO). Eight real datasets are used to compare their performance. The ACO algorithm provides better solutions than the CSOP solver in a shorter time. Results show that the pheromones used in the ACO help to find better solutions in a much smaller amount of time. The short computation time enables the school's directors to simulate different compositions of their future classes before having the final results of the last exams.

1. INTRODUCTION

The Resource Allocation Problem (RAP) consists in assigning a limited number of resources to objects, which can be activities as in communication systems [1, 2], or tasks as in scheduling problems [3, 4]. The resources can be for example goods [5], people [3, 4], money [6], or seats in courses [7, 8].

Usually a RAP has an **objective** as maximizing satisfaction or productivity. It can be Multi-Objective (MORAP) and have objectives that might be opposite, as in asset management, maximize revenue and minimize risk [6], or in staff scheduling, minimize operational cost and improve working conditions [4]. In the case of a MORAP, the objectives can be combined into a weighted function so that the problem becomes a single-objective RAP. The Pareto optimality can also be considered to solve a MORAP, in which case, the solutions that have to be found are the non-dominated ones. If an objective is improved in a non-dominated solution, at least one other objective will be penalised.

If the objective is influenced by both sides, objects and resources, the RAP is then often referred to as a matching problem [9]. For example, when seats are allocated to students in a university, students may submit preferences for the courses, in which case, the simple objective might be to maximize students' satisfaction [8]. If professors have also preferences for students, because of their background or their grades, then both types of preferences, students' and professors', have to be considered in order to find a good solution to the problem.

In addition to the objective, in most optimisation problems, hard **constraints** have to be considered. If any of those constraints is not fully satisfied in a solution, then this solution will not be feasible. We can find for example capacity constraints [5, 7, 8], or skills constraints [4].

In the recent years, different types of approaches to **solve** the RAP and especially the MORAP have been proposed. Dynamic programming has been used to optimise a portfolio in a project financing firm [10]. Neighbourhood search [11], particle swarm optimisation [12] and genetic algorithms [13] have been applied to the classical problem of assigning workers to jobs where the objectives are to maximize the efficiency and to minimize the cost. Evolutionary algorithms have also been used for the allocation of projects to students where five different objectives had to be considered [14] and for the allocation of resources to projects over several periods of time [15].

In this paper, we propose a novel application of two popular and widely used computational intelligence methods to a real optimisation problem that is modelled as a RAP. This problem occurs in secondary schools in Canton de Vaud (Vaud), a French part of Switzerland. A new educational system was launched in 2013, [16, 17]. Before 2013, students were allocated to classes and then a traditional timetabling problem was solved with commercial software in almost all schools in Vaud, since all students in a class attended the same lessons. In the new system, there still exist classes, but a student will be part of one main class and of several other classes depending on the lessons he has to attend. Students in the same main class will therefore be split and grouped differently with other students for the some activities scheduled during the week. This split and group process is not specific to Switzerland, it e in high schools in Greece [18], Germany [19] or Australia [20]. This process allows having students in the same main class who are split for options or because of their different level in a field. In Switzerland, those options and fields represent sometimes more than half of the timetable of a student.

In most cases, the splitting and grouping of students has an impact on the timetable, but is not considered as an objective. In Vaud, the pedagogical objective is to have main classes as mixed as possible and to have a similar diversity between classes. A student can be described with a profile with the options he selected and his level in the different fields, mixed classes are preferred in order to avoid situations where students with the same profile are allocated to the same class. Students can thus be seen as resources allocated to main classes, and the objective of the RAP is the diversity within each class and the similarity between classes. Regarding diversity, classes are competitors since an assignment may lead to one class perfectly mixed penalizing the others. Regarding similarity, classes have to distribute students in a fair way among them in order to satisfy this objective as much as possible. In [17], a complete model that combined the allocation and the timetabling problems was proposed, but not solved.

In this new work, we propose a model and two resolution approaches for the allocation of students to classes in order to have classes as mixed and similar as possible. The problem is first modelled as a Constraint Satisfaction Optimisation Problem (CSOP), [21]. The CSOP is then solved with a CSOP solver, Gecode [22], and with an Ant Colony Optimisation (ACO) algorithm, [23-25]. ACOs' algorithms have often been used to solve different types of optimisation problems. In the early years of this metaheuristic, most of the problems where ACO was used could be represented with a graph, as the Travelling Salesman Problem [26], but very quickly ACO algorithms spread and were used in many different fields, for example an ant colony system has been used to define the location of hubs in a network and to allocate each node of this network to one of the hubs [27]. ACO algorithms have also been adapted to solve a MORAP [28], a timetabling problem [29] or a course allocation problem [8].

The rest of the paper is structured as follows. Section 2 describes the problem in Vaud. Section 3 presents the model for the problem and for the CSOP and the ACO approaches. Section 4 contains the description of the results. Section 5 contains the conclusions and presents the future work.

2. DESCRIPTION OF THE CLASSES' SETTING UP PROBLEM

In Vaud, a new program for secondary schools has been implemented in August 2013. Ninety schools are impacted by this new program, which corresponds to three grades (around years 12 to 15). This new program is briefly described in this section, for more details about it, see [17].

Students are divided into two **sections**. The *Voie Générale* (VG) prepares students to a vocational school. The *Voie Prégymnasiale* (VP) prepares them to a high school.

Each student attending a VP level will be assigned to one **option** (OS) he can select among 4 topics. The VG students will select 2 different options (OCOM) among 6 topics. If a VG student has high grades, he may attend one OS with the VP students instead of both OCOMs.

For a few subjects, VG students' **level** is taken into consideration and they will attend lessons in groups of students with the same level. As for the options, VG students may attend lessons with VP students if their level is good enough.

For the other subjects, with no level and no options, students are grouped in classes, called section classes. All students of a section class will attend the lessons of those subjects together. All section classes of the same grade and section should have a similar size.

The pedagogical objective is the **diversity** of each VG section class regarding the level fields and the options and the **similarity** between VG section classes of the same grade. For example, if we consider 2 section classes and a level field, 24 students with level L_1 and 8 students with level L_2, a class with an optimal diversity will be made of 8 L_1 students and 8 L_2 students, but the second will have then 16 L_1 students. Optimal similarity means that each section class will have 12 L_1 students and 4 L_2 students.

The timetabling with diversity becomes more difficult because the schedule of several section classes has to be the same for some lessons. In the previous example for the optimal similarity solution, both section classes need to have the same schedule for the considered field, all L_2 students will attend the same lessons (as will L_1 students). In addition to that, there is an impact on the resources needed since if L_1 and L_2 lessons have the same schedule, the school needs more teachers for this subject.

Considering levels and options, in a school we may have up to 4 VP profiles and 432 VG profiles for students. Currently, the school's director assigns students to classes manually and creates the timetable with commercial software. As the diversity objective of the assignment has a big impact on the timetable's constraints, this objective is sometimes not considered.

3. THE MODEL FOR THE CLASS ALLOCATION PROBLEM

This section describes the proposed model for the Class Allocation Problem (CLAP) in detail. Section 3.1 describes the concepts of the Vaud problem. Section 3.2 defines the solution and the hard constraints and section 3.3 explains how the diversity and similarity objectives are modelled as a cost for the optimization problem. Sections 3.4 and 3.5 describe respectively the CSOP and the ACO models.

3.1 Concepts

The main concepts of the allocation problem are:

Grade, $g \in G$. In Vaud, there are 3 grades in secondary schools: 9, 10 and 11.

Categories, $k \in K$. A category corresponds to a group of students who have an option, a level or a section and grade in common. Examples of categories are grade-section (*9-VP*), grade-OS (*11-Latin*), grade-OCOM (*10-Visual Arts*), and levels (*9-Math-L_1*). A category is thus defined with its grade $KG(k)$, its type $KT(k) \in \{section, OS, OCOM, level\}$ and a

maximal number of students per class $KS_{max}(k)$. For every grade, there is only one category whose type is $section$.

Classes, $q \in Q$. A class is a grouping of students and is defined with a category $QK(q)$. All students who belong to a category must be allocated to one class of this category. The number of classes of a category k depends on the category. In the rest of the paper, the classes that are considered are only the section classes, since the way the students are split in the other classes has no impact on the objectives.

Students, $s \in S$. A student is defined with the categories he belongs to $SK(s)$. A student belongs to one single category whose type is $section$.

3.2 Solution and hard constraints

A **solution** μ to the allocation problem contains the assignment of students to classes:

$$\mu = \{(s, q) \in S \times Q\}$$

$$(s, q) \in \mu \Leftrightarrow \text{student } s \text{ is assigned to class } q \tag{1}$$

The **hard constraints** in an optimization problem must always be satisfied in order to have a feasible solution. In our allocation problem, there are only two types of hard constraints:

- A student must be assigned to one class of each category he belongs to, Eqs. (2) and (3)
- The number of students per class is limited and depends on its category, Eq. (4)

$$\forall s \in S, \forall k \in SK(s): \quad \exists! (s, q) \in \mu: QK(q) = k \tag{2}$$

$$\forall (s, q) \in \mu: \quad QK(q) \in SK(s) \tag{3}$$

$$\forall q \in Q: \quad Card(\{s \in S: (s, q) \in \mu\}) \leq KS_{max}(QK(q)) \tag{4}$$

3.3 Objective

Soft constraints are constraints that should be satisfied as much as possible. Usually a solution where all the soft constraints are satisfied does not exist. Therefore a cost is associated with a soft constraint that is not satisfied and the objective of the problem is to minimize the total cost of the solution. In the allocation problem, the objective is to have diversity within section classes and similarity between them. The similarity objective contains implicitly the diversity objective. Indeed if all classes are similar, then they are equally diverse.

For an allocation μ, the number of students in a class q that belong to a category k is given by Eq. (5). We can thus define the average number of students of a category k that should be in a section class as the number of students belonging to k divided by the number of section classes of the grade of k, Eqs. (6), where Q_k is the list of section classes that belong to the same grade as k, as defined in Eq. (7).

$$QN(q, k) = Card(\{(s, q) \in \mu: k \in SK(s)\}) \tag{5}$$

$$\overline{QN}(k) = \frac{Card(\{s \in S: k \in SK(s)\})}{Card(Q_k)} \tag{6}$$

Where $\quad Q_k = \{q \in Q: KG(QK(q)) = KG(k) \wedge KT(QK(q)) = "section"\}) \tag{7}$

Two section classes q_i, q_j that belong to the same grade are similar for a category k if they have the same number of students for this category, $QN(q_i, k) = QN(q_j, k)$. The section classes of a grade are similar if they are all similar for any category. In an optimal solution for

similarity, $QN(q,k)$ is thus the same for all classes $q \in Q$ and all categories $k \in K$. The absolute deviation of a category k defined in Eq. (8) is used to measure how far a class is from the optimal similarity. The similarity objective is modelled with the minimisation of the sum of the absolute deviation for all categories as defined in Eq. (9).

$$AbsDev(k) = \frac{1}{Card(GS(k))} \sum_{\{q \in GS(k)\}} |QN(q,k) - \overline{QN}(k)| \tag{8}$$

$$Cost(\mu) = \sum_{k \in K} AbsDev(k) \tag{9}$$

3.4 The CSOP model

As defined in [21] and [23], a Constraint Satisfaction Optimization Problem (CSOP) can be described with three sets and one function $(X; D; C; f)$. X is the set of variables. The domain D_x is the set of values that can be assigned to each variable $x \in X$. C is the set of constraints that limits the assignment of values to variables. A solution to a CSOP is the assignment to each variable of one value that belongs to its domain such that the constraints of C are fully satisfied. The fitness function f maps a value to each possible solution and the objective is to optimize the value of f.

For the CLAP, **the variables** $X = \{X_1, \dots, X_n\}$ are the section class allocated to students, where $n = Card(S)$ is the number of students, e.g. X_s is the section class allocated to student s. The **domain** D_{X_s} is thus the list of the section classes corresponding to the student s as defined in Eq. (10). **The constraints** of this model are the capacity of the classes; the number of students in a class cannot exceed its capacity, Eq. (11).

$$\forall s \in S: \quad D_{X_s} = \{q \in Q : QK(q) \in SK(s) \wedge KT(QK(q)) = "section"\} \tag{10}$$

$$\forall q \in Q: \quad Card(\{X_i \in X : X_i = q\}) \leq KS_{max}(QK(q)) \tag{11}$$

3.5 The ACO model

As explained in [24], *Ant System* is the first Ant Colony Optimization (ACO) algorithm introduced by Dorigo in 1992, and is a probabilistic algorithm based on the foraging behaviour of real ants. When ants find food, they deposit pheromones on the paths they use, and those pheromones are then detected by the other ants in order to try and find the food source. A variant of the ACO algorithm, MAX-MIN Ant System, was introduced in [26], where only the best ants deposit pheromones, and the quantity of pheromone has an upper and a lower limit.

In the ACO algorithm presented in this paper in Fig. 1, a path corresponds to an assignment (s,q) of a student to a section class, and each assignment has a quantity of **pheromones** on it which will be used to calculate the probability of including this assignment in the solution of an ant. At each iteration, each ant of the colony will build a solution step by step selecting one assignment (s,q) for each student, line 13. At the beginning of the algorithm, all pheromones $\tau_{s,q}$ are initialised to a given value, lines 5-7. At the end of an iteration, part of the pheromones evaporate, which means that the value of $\tau_{s,q}$ is reduced, lines 24-26. The fitness of the solutions found by the ants during the iteration will be computed, line 19, and compared, and the best ant will deposit pheromones on the assignments (s,q) that belong to its solution, the value of $\tau_{s,q}$ is thus increased, lines 27-29.

In addition to the pheromones, the ants will also use the **visibility** η_q they have on their own partial solution. Indeed when it is building a solution, each ant knows which assignments

are in its partial solution and the capacity of the classes. As classes should never be overbooked, the ant has to know somehow if a class can still be assigned to a student or not. The visibility has thus to be updated after each allocation of a student to a class and its value is given by Eq. (12), lines 2-4 and 15-17. The **probability** to select a section class q for a student s is then given by Eq. (13), where Q_s is the list of section classes that belong to the same grade as s, as defined in Eq. (14).

$$\eta_q = \begin{cases} 1 & \text{if there are still seats available in } q \\ 0 & \text{if } q \text{ has no more seats available} \end{cases} \tag{12}$$

$$p(s, q) = \frac{\tau_{s,q} \cdot \eta_q}{\sum_{i \in Q_s} \tau_{s,i} \cdot \eta_i} \tag{13}$$

Where $$Q_s = \{q \in Q : QK(q) \in SK(s) \land KT(QK(q)) = \text{"section"}\} \tag{14}$$

When an assignment (s, q) belongs to the best solution of an iteration, pheromone is deposited on it at the end of the iteration, the values of $\tau_{s,q}$ and therefore of $p(s, q)$ are increased. In the next iteration, q has then a higher probability to be selected for s than in the iteration that just finished. The parameters used in this ACO algorithm are the number of iterations, the number of ants, the evaporation rate ρ, and the deposition rate φ.

Input: List of students with their categories, List of section classes with their capacity
Output: Allocation μ: Assignment of each student to one section class
1: $\mu \leftarrow \emptyset$
2: **for** $q \in Q$ **do**
3:     $\eta_q = 1$ // Visibility initialisation
4: **end for**
5: **for** $(s, q) \in (S \times Q)$ **do**
6:     $\tau_{s,q} = \tau_{init}$ // Pheromones initialisation
7: **end for**
8: **for** $i = 1, \ldots, NbIterations$ **do**
9:     $\mu_{iter} \leftarrow \emptyset$
10:     **for** $j = 1, \ldots, NbAnts$ **do**
11:         $\mu_{ant} \leftarrow \emptyset$
12:         **for** $s = 1, \ldots, NbStudents$ **do**
13:             Select a section class q of the grade of s
14:             $\mu_{ant} \leftarrow \mu_{ant} \cup (s, q)$
15:             **if** $Card(\{(s_i, q) \in \mu_j\}) = KS_{max}(QK(q))$ **then**
16:                 $\eta_q = 0$ // Visibility update
17:             **end if**
18:         **end for**
19:         Calculate the fitness function $f(\mu_{ant}) = \sum_{k \in K} AbsDev(k)$
20:         **if** $f(\mu_{ant}) < f(\mu_{iter})$ **then**
21:             $\mu_{iter} \leftarrow \mu_{ant}$
22:         **end if**
23:     **end for**
24:     **for** $(s, q) \in (S \times Q)$ **do**
25:         $\tau_{s,q} \leftarrow (1 - \rho) \cdot \tau_{s,q}$ // Pheromones evaporation
26:     **end for**
27:     **for** $(s, q) \in \mu_{iter}$ **do**
28:         $\tau_{s,q} \leftarrow \tau_{s,q} + \frac{\varphi}{f(\mu_{iter})}$ // Pheromones deposition
29:     **end for**
30:     **if** $f(\mu_{iter}) < f(\mu)$ **then**
31:         $\mu \leftarrow \mu_{iter}$
32:     **end if**
33: **end for**

Figure 1: Model; the ACO Algorithm.

4. EXPERIMENTATION

4.1 The data set: Canton de Vaud, Switzerzland

The eight data sets used in this paper come from the *Direction Générale de l'Enseignement Obligatoire* from Canton de Vaud in Switzerland. They correspond to two different grades of four schools. The available data is the list of students with the categories they belong to and the number of section classes per grade. One student belongs usually to 5 categories, few of them belong to only 3 or 4. Table I contains the number of students, of section classes and of categories per school and per grade-section.

Table I: Experimentation; size of the data sets.

Grade		School			
		2112	**1213**	**3411**	**3621**
9-VG	Students	113	71	78	68
	Section classes	6	4	5	4
	Categories	16	14	15	17
10-VG	Students	137	90	78	66
	Section classes	8	5	5	4
	Categories	15	16	15	16

4.2 The results

The computation was done on an Intel® Core™ i5-5300U CPU 2.30 GHz. Gecode was used to solve the CSOP model, [22]. Gecode used three threads, the ACO approach only one. In Gecode, the selection of the variables is always the same and depends only on the sequence of students in the file, the selection of their value for the branching is done randomly, and the runtime was limited to 10 minutes. 80 iterations with 60 ants each were used for the ACO approach, one run needed between 1 and 3 seconds.

The algorithms were launched 10 times for each data set. Table II contains the results: average, standard deviation, best and worst results. Regarding the average, the ACO clearly outperforms the CSOP, which is between 14 % and 44 % worse. Regarding the best solution found among the 10 runs, the CSOP outperforms or is identical to the ACO 4 times.

Table II: Experimentation; results, average, standard deviation, best and worst result with 10 runs.

School	Grade	ACO			CSOP 10 minutes		
		Av. ± St. Dev.	Best	Worst	Av. ± St. Dev.	Best	Worst
2112	9-VG	**58.4 ± 5.7**	**52**	**68**	79.0 ± 10.2	62	96
	10-VG	**75.2 ± 4.8**	67	**81**	103.0 ± 19.0	**61**	139
1213	9-VG	**26.8 ± 4.1**	**22**	**34**	37.2 ± 5.7	30	46
	10-VG	**39.2 ± 5.3**	**33**	47	56.6 ± 13.5	**33**	79
3411	9-VG	**38.2 ± 3.8**	**34**	**44**	51.4 ± 4.3	44	58
	10-VG	**37.8 ± 3.5**	**33**	**45**	51.4 ± 9.9	37	69
3621	9-VG	**29.2 ± 3.5**	**22**	**34**	33.2 ± 8.2	**22**	46
	10-VG	**27.0 ± 2.9**	**22**	**32**	36.6 ± 8.9	**22**	56

Figs. 2 and 3 show how students are spread in the different section classes for each category in a solution found by ACO. For each category, the closer the points are, the better the solution is. Category 1 corresponds to the section category.

259

For category 1, the size of the section classes is between 18 and 19 students in Fig. 2, and they are thus similar. For the other categories, the difference in the number of students per class is between 1 and 4, e.g. in category 7, class 3 has 9 students of this category and class 2 has 12 of them.
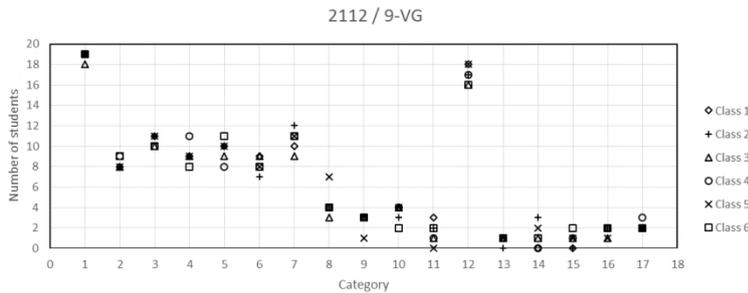

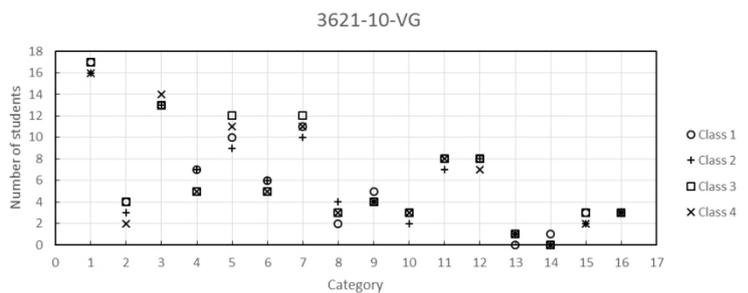Figure 2: Experimentation; results for school 2112, grade 9-VG, with ACO.

Figure 3: Experimentation; results for school 3621, grade 10-VG, with ACO.

Fig. 4 shows the cost of the current best solution during one run of the ACO algorithm for two different sections of two schools. The convergence takes place after 1 to 3 seconds, depending on the school and the grade considered.
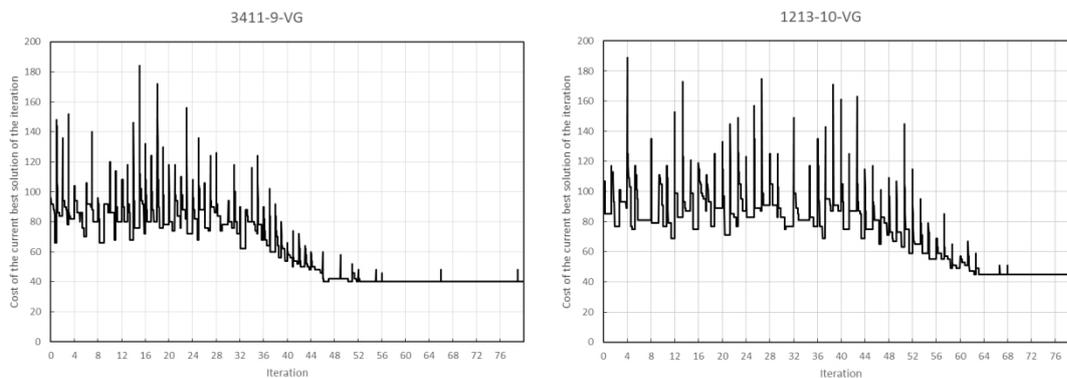
Figure 4: Experimentation; evolution of the solution cost for the ACO algorithm in 1 run.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have presented two approaches to solve a resource allocation problem in educational institutions where students have different profiles in different topics. The objective is to have all classes with a similar composition for all topics, which means that students with the same profile on a topic have to be spread among the available classes. This maximises also the diversity of each class, since a maximum number of different profiles will be allocated to each of them.

Two approaches were used. The ACO algorithm provides better solutions than the CSOP solver in a shorter time. As both of them use randomness for the allocation of resources, our

results prove that the pheromones in the ACO approach help to find very good solutions in a much smaller amount of time.

As the computation time is short for the ACO approach, the school's directors, who are in charge of the allocation of students and of the timetable, can simulate multiple scenarios of possible sets of profiles in their school. They can then analyse the impact of those scenarios on the pedagogical objective with the solution proposed in this paper and on the timetable problem with the commercial software available in the different schools.

We are currently including in the model part of the resolution of the timetable problem in order to analyse the impact of similar classes on the timetabling constraints and on the resources needed. We plan also to extend the ACO algorithm to include the timetabling problem and to compare then our results with the results obtained with the commercial software currently used in most schools of Canton de Vaud.

In our future work, we plan to compare the results obtained with our ACO and CSOP approaches with other existing approaches, such as genetic algorithms.

ACKNOWLEDGEMENT

REFERENCES

[1] Zhang, H.; Jiang, C.; Beaulieu, N. C.; Chu, X.; Wang, X.; Quek, T. Q. S. (2015). Resource allocation for cognitive small cell networks: A cooperative bargaining game theoretic approach, *IEEE Transactions on Wireless Communications*, Vol. 14, No. 6, 3481-3493, doi:10.1109/TWC.2015.2407355

[2] Ng, D. W. K.; Lo, E. S.; Schober, R. (2016). Multiobjective resource allocation for secure communication in cognitive radio networks with wireless information and power transfer, *IEEE Transactions on Vehicular Technology*, Vol. 65, No. 5, 3166-3184, doi:10.1109/TVT.2015.2436334

[3] Heimerl, C.; Kolisch, R. (2010). Scheduling and staffing multiple projects with a multi-skilled workforce, *OR Spectrum*, Vol. 32, No. 2, 343-368, doi:10.1007/s00291-009-0169-4

[4] Maenhout, B.; Vanhoucke, M. (2013). An integrated nurse staffing and scheduling analysis for longer-term nursing staff allocation problems, *Omega*, Vol. 41, No. 2, 485-499, doi:10.1016/j.omega.2012.01.002

[5] He, L. X.; He, S. H. (2015). Solving water resource scheduling problem through an improved artificial fish swarm algorithm, *International Journal of Simulation Modelling*, Vol. 14, No. 1, 170-181, doi:10.2507/IJSIMM14(1)CO5

[6] Ehrgott, M.; Klamroth, K.; Schwehm, C. (2004). An MCDM approach to portfolio optimization, *European Journal of Operational Research*, Vol. 155, No. 3, 752-770, doi:10.1016/S0377-2217(02)00881-0

[7] Budish, E.; Cantillon, E. (2012). The multi-unit assignment problem: Theory and evidence from course allocation at Harvard, *American Economic Review*, Vol. 102; No. 5, 2237-2271, doi:10.1257/aer.102.5.2237

[8] Nogareda, A.-M.; Camacho, D. (2016) Optimizing satisfaction in a multi-courses allocation problem combined with a timetabling problem, *Soft Computing*, 1-10, doi:10.1007/s00500-016-2375-8

[9] Sönmez, T.; Ünver, M. U. (2011). Matching, allocation, and exchange of discrete resources, Benhabib, J.; Bisin, A.; Jackson, M. O. (Eds.), *Handbook of Social Economics*, Vol. 1, Elsevier, Amsterdam, 781-852

[10] Basso, A.; Peccati, L. A. (2001). Optimal resource allocation with minimum activation levels and fixed costs, *European Journal of Operational Research*, Vol. 131, No. 3, 536-549, doi:10.1016/S0377-2217(00)00093-X

[11] Liang, Y.-C.; Chuang, C.-Y. (2013). Variable neighborhood search for multi-objective resource allocation problems, *Robotics and Computer-Integrated Manufacturing*, Vol. 29, No. 3, 73-78, doi:10.1016/j.rcim.2012.04.015

[12] Fan, K.; You, W.; Li, Y. (2013). An effective modified binary particle swarm optimization (mBPSO) algorithm for multi-objective resource allocation problem (MORAP), *Applied Mathematics and Computation*, Vol. 221, 257-267, doi:10.1016/j.amc.2013.06.039

[13] Osman, M. S.; Abo-Sinna, M. A.; Mousa, A. A. (2005). An effective genetic algorithm approach to multiobjective resource allocation problems (MORAPs), *Applied Mathematics and Computation*, Vol. 163, No. 2, 755-768, doi:10.1016/j.amc.2003.10.057

[14] Rachmawati, L.; Srinivasan, D. (2005). A hybrid fuzzy evolutionary algorithm for a multi-objective resource allocation problem, *Proceedings of the 5th International Conference on Hybrid Intelligent Systems (HIS'05)*, 6 pages, doi:10.1109/ICHIS.2005.10

[15] Ghorbani, S.; Rabbani, M. (2009). A new multi-objective algorithm for a project selection problem, *Advances in Engineering Software*, Vol. 40, No. 1, 9-14, doi:10.1016/j.advengsoft.2008.03.002

[16] Canton de Vaud, from *http://www.vd.ch/themes/formation/scolarite-obligatoire/contenus-denseignement/*, accessed on 17-07-2016

[17] Nogareda, A. M.; Camacho, D. (2015). Constraint-based model design for timetabling problems in secondary schools, *Proceedings of the International Symposium on Innovations in Intelligent SysTems and Applications* (INISTA), 6 pages, doi:10.1109/INISTA.2015.7276777

[18] Beligiannis, G. N.; Moschopoulos, C. N.; Kaperonis, G. P.; Likothanassis, S. D. (2008). Applying evolutionary computation to the school timetabling problem: The Greek case, *Computers & Operations Research*, Vol. 35, No. 4, 1265-1280, doi:10.1016/j.cor.2006.08.010

[19] Jacobsen, F.; Bortfeldt, A.; Gehring, H. (2006). Timetabling at German secondary schools: Tabu search versus Constraint programming, *Proceedings of the 6th International Conference on the Practice and Theory of Automated Timetabling*, 439-442

[20] Kingston, J. H. (2004). A tiling algorithm for high school timetabling, *Proceedings of the 5th International Conference on the Practice and Theory of Automated Timetabling*, 208-225

[21] Tsang, E. (1993). *Foundations of Constraint Satisfaction*, Academic Press, London

[22] Schulte, C.; Tack, G.; Lagerkvist, M. Z. (2013). Modeling and programming with Gecode, from *http://www.gecode.org/doc-latest/MPG.pdf*, accessed on 26-12-2016

[23] Di Gaspero, L. (2015). Integration of metaheuristics and constraint programming, Kacprzyk, J.; Pedrycz, W. (Eds.), *Springer Handbook of Computational Intelligence*, Springer, Berlin, 1225-1237, doi:10.1007/978-3-662-43505-2_62

[24] Dorigo, M.; Birattari, M.; Stutzle, T. (2006). Ant colony optimization, *IEEE Computational Intelligence Magazine*, Vol. 1, No. 4, 28-39, doi:10.1109/MCI.2006.329691

[25] Solnon, C.; Jussien, N. (2010). *Ant colony optimization and constraint programming*, John Wiley & Sons, Hoboken, doi:10.1002/9781118557563

[26] Stutzle, T.; Hoos, H. (1997). MAX-MIN ant system and local search for the traveling salesman problem, *IEEE International Conference on Evolutionary Computation*, 309-314, doi:10.1109/ICEC.1997.592327

[27] Randall, M. (2008). Solution approaches for the capacitated single allocation hub location problem using ant colony optimisation, *Computational Optimization and Applications*, Vol. 39, No. 2, 239-261, doi:10.1007/s10589-007-9069-1

[28] Chaharsooghi, S. K.; Kermani, A. H. M. (2008). An effective ant colony optimization algorithm (ACO) for multi-objective resource allocation problem (MORAP), *Applied Mathematics and Computation*, Vol. 200, No. 1, 167-177, doi:10.1016/j.amc.2007.09.070

[29] Socha, K.; Sampels, M.; Manfrin, M. (2003). Ant algorithms for the university course timetabling problem with regard to the state-of-the-art, Cagnoni, S.; Johnson, C. G.; Romero Cardalda, J. J.; Marchiori, E.; Corne, D. W.; Meyer, J.-C.; Gottlieb, J.; Middendorf, M.; Guillot, A.; Raidl, G. R.; Hart, E. (Eds), *Applications of Evolutionary Computation*, Springer, Berlin, 334-345, doi:10.1007/3-540-36605-9_31