Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in DISCRETE MATHEMATICS

Electronic Notes in Discrete Mathematics 58 (2017) 119–126 www.elsevier.com/locate/endm

Variable neighborhood descent search based algorithms for course timetabling problem: Application to a Tunisian University

Rahma Borchani[a,b] [1] Abdelkerim Elloumi[a] [2]
Malek Masmoudi[b] [3]

[a] *Laboratory of Modeling and Optimization for Decisional, Industrial and Logistic Systems (MODILS), Faculty of Economics and Management of Sfax, Tunisia*

[b] *Université de Lyon, F-42023, Saint Etienne, France; Université de Saint Etienne, Jean Monnet, F-42000, Saint-Etienne, France; LASPI, F-42334, IUT de Roanne*

Abstract

University course timetabling problem refers to schedule a set of lectures, tutorials and practical works to a limited number of teachers, classrooms and time slots over a planning horizon while satisfying a set of hard and soft constraints. In this paper, we investigate Variable Neighborhood Descent approach to tackle a specific course timetabling problem related to the case of the Faculty of Economics and Management Sciences of Sfax in Tunisia. The objective is to minimize the total number of holes and the number of isolated lessons for all student groups. We have developed eleven specific neighborhood structures: six of these are designed for correcting the holes, while the other five are designed for adjusting the isolated lessons. Computations are made on real instances and numerical results show that our approach outperforms the existing solution with the elimination of 52.47 % of holes and isolated sessions on average.

Keywords: Course timetabling, VND, neighborhood structure, isolated lesson

http://dx.doi.org/10.1016/j.endm.2017.03.016
1571-0653/

1 Introduction

Course Timetabling Problem is considered as one of the hardest problems faced by academic institutions throughout the world. In the II International Timetabling Competition (ITC2007), this problem is decomposed into two sub-categories problems: Post Enrolment-based Course Timetabling and Curriculum-based Course Timetabling. The latter is the one considered in this paper. It involves scheduling a set of courses into a weekly timetable, where each course must be assigned to a teacher, a classroom and a time slot, subject to a set of hard and soft constraints. A variety of timetables problem exists since several specific constraints vary from one institution to another. A variety of solving approaches exist too and have been grouped into three classes: the class of exact methods, the class of approximate methods, and the pseudo class of hybrid methods. Among exact solution strategies, we cite for example the Integer Programming [4], [12], and the Vertex Coloring [3]. For the second class, we cite for example Tabu Search algorithm [9], Genetic Algorithm [11], and Memetic Algorithm [10]. For hybrid techniques, we cite for example the combination of Genetic Algorithm with Tabu Search algorithm [1], Variable Neighborhood Search with Simulated Annealing [2], and Mathematical Programming with Lagrangian Relaxation and Simulated Annealing [7].

In course timetabling domain and according to the ITCs Competitions, local search methods outperform the state-of-art heuristic solvers [6]. As presented by [8], the Variable Neighborhood Search (VNS) is versatile and successful compared to other local search techniques thanks to the strategy to consider different neighborhood structures providing a very effective way for escaping local optima.

In this paper, we investigate the variable neighborhood descent algorithm for course timetabling problems to outperform the manually generated solutions in the Faculty of Economics and Management of Sfax in Tunisia (FEMSS). The remainder of this paper is organized as follows. Section 2 describes the problem considered in this paper. Section 3 presents our solution approach. Section 4 contains the numerical results. Section 5 presents the conclusions and future directions.

[1] Email: borchani.rahma@gmail.com
[2] Email: abdelkarim.elloumi@laposte.net
[3] Email: malek.masmoudi@univ-st-etienne.fr

2 Problem description

The construction of course timetabling at the FEMSS is performed by administrative staff. It involves seventy sections on different disciplines. The curriculum of each section includes several modules where each module may include three types of activities : lectures, tutorials and practical works. Students from each section are partitioned into groups of thirty students maximum, and each group containing more than 20 students is partitioned into subgroups of fifteen students maximum. Lectures are given to the hole students in the same section. Tutorials are given to students in the same group and practical works to students in the same subgroups. Several courses within the same module must be scheduled once or twice per week, but maximum once per day. At FEMSS, 77 rooms are available; 5 big rooms reserved only for lectures, 54 medium rooms used for tutorials and lectures, and 18 rooms reserved for practical works. The timetable at the FEMSS is partitioned into 30 periods of 90 minutes with three periods per half a day, and there is no class on Sunday, Wednesday afternoon and Saturday afternoon. The number of sections, groups, subgroups, courses, lectures, tutorials and practical works during the last three academic years (2012, 2013 and 2014) are presented in Table1.

Table 1
FEMSS' dataset for academic years 2012, 2013 and 2014.

Instance	2012-1	2012-2	2013-1	2013-2	2014-1	2014-2
Number of sections:	65		69		70	
Number of subgroups:	150		176		185	
Number of courses	1619	1486	1880	1653	1904	1637
Number of lecture	1000	895	1104	969	1116	891
Number of tutorial	501	544	652	630	680	691
Number of practical works	118	47	124	54	108	55

The FEMSS course timetabling problem is handled by ([5], [4]). The authors provided a new mathematical model, but the real problem is too Hard to be solved by a commercial software package such as CPLEX.Thus, authors in [4] have developed a heuristic procedure called Lecture Timetabling Heuristic to find a feasible solution for the sub-problem including only lectures. In [5], they have extended the formulation by including both lectures and tutorials. They have implemented and tested those proposed solutions on real data of the FEMSS and they have considered only these four constraints :

- All lectures and tutorials included in the curriculum of each section must be taught.

- No professor can teach more than one activity in any time slot.
- No classroom can be used for more than one activity in any timeslot.
- No group can be taught more than one activity in any timeslot.

In this paper, we include three types of activities : lectures, tutorials and practical works and we introduce two new soft constraints. First one, minimize the number of free time slots between two consecutive activities (called hole). Second one ; minimize the number of single activity within a half day (called isolated lesson).

Motivated by the recommendations of [4], the complexity of the problem, and the high proprieties of the local search techniques, we opted for the development of a Variable Neighborhood Descent approach to solve this real problem.

3 Variable Neighbourhood Descent approach

The Variable Neighborhood Descent (VND) methods presented by Hansen et al. [7], explore the search space in a deterministic change of the neighborhood structures. In our work, we have used two variants of VND algorithms, namely the sequential VND and the random VND:

- Sequential VND (Seq-VND): The algorithm change neighborhoods structure in a sequential order. It starts with the first neighborhood N1, and precedes to the neighborhood Nk+1 unless it does not fall on a local optimum with the neighborhood Nk. Otherwise, it returns to N1.
- Random VND (Rand-VND): The algorithm has the same principle of the Sequential VND, but using a random change of the neighborhood structures.

Those two approaches use the first-improvement strategy and accept only an improved solution. Our search procedures incorporate eleven neighborhood structures, six of them (N1 to N6) are designed for correcting the holes, while the other five (N7 to N11) are designed for adjusting the isolated lessons. The solution provided by the search procedures should be a local optimum with respect to all used neighborhoods. Those neighborhood structures are applied by respecting the conditions on [4] and the following constraints:

- The teacher assigned to the moved activity must not be changed and thus be available in the new period.
- The new assigned room must respect the required type, be available in the new period, and has a capacity greater or equal than the size of the assigned group.
- A moved activity that has two periods cannot be held on the same day.

All those neighborhood structures are presented with illustrative examples in Figure 1.

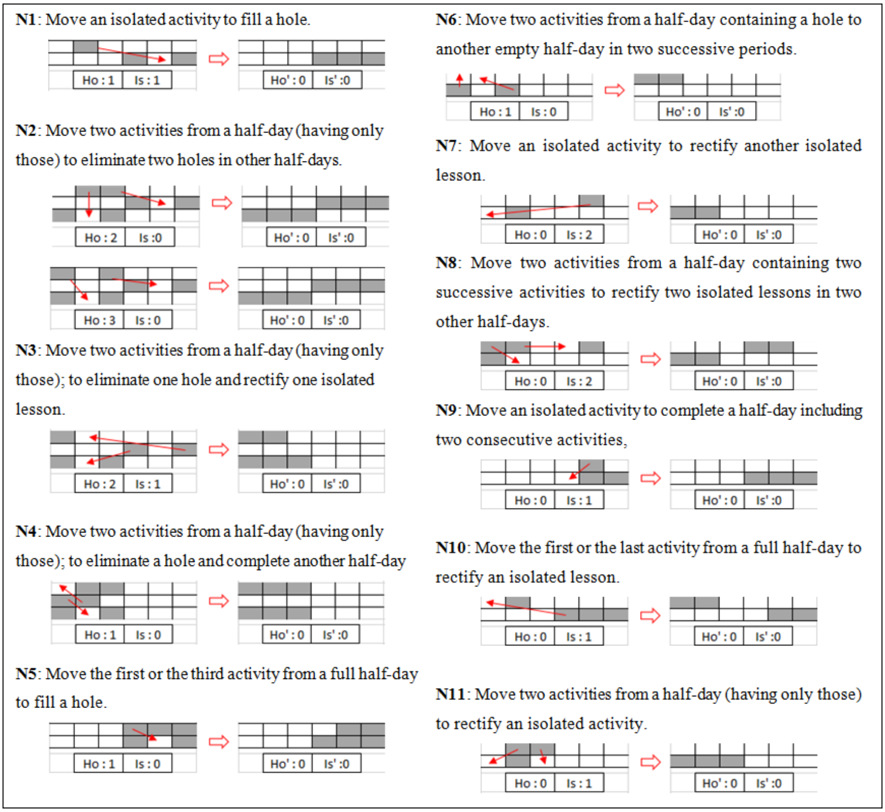

Fig. 1. Neighborhood structures: N1 to N11

Our VND approach uses the timetabling provided by the administrative staff as initial solution. The main idea consists of improving the schedule of all groups of students by moving tutorials and practical works schedules. First, for each student group, we seek the holes and isolated lessons and put them on two different lists. Second, a student group is selected using one of the three different algorithms that are explained later. The local search method is used to eliminate holes in the first hand, then to reduce the isolated activities. After each move, we update the availability of professors, classrooms, lists of holes and isolated activities. We repeat this step until we cannot eliminate neither holes nor isolated activities in the selected group. Finally, we repeat the second step to improve another student group until we finish all groups. The three algorithms are considered to select groups of students:

- Algorithm1: browses different groups sequentially respecting the order of appearance in the university database.

- Algorithm2: assigns a group weight that depends on the number of holes in each processed group. A random weight is chosen and the search procedure is applied to the corresponding group.
- Algorithm3: the transition from one group to another follows the decreasing order of the sum of holes (HO) and isolated lessons (IS).

Figure 2 shows our whole Variable Neighborhood Descent algorithm.

```
Input: x: = Initial solution, the set of neighborhood structures N_k, k = 1 ... k11;
List: List of subgroups sorted according to Algorithm1, Algorithm2 or Algorithm3;
For each subgroup s in List
  Hole_List_s:= list of holes corresponding to subgroup s; Isolated_List_s:= list of isolated sessions corresponding to subgroup s
 Repeat                                          // Repeat the search procedure to correct the holes in the subgroup s
   ho = 0;                                       // counter of holes in the Hole_List_s
   While (ho <  Hole_List_s.count and Hole_List_s.count  > 0)
     k=0; FD = false;                            // FD: Boolean variable; FD =true if we correct the current hole else FD = false
     Repeat
       If (Seq-VND) then k: = k +1;              //increment k if we choose to apply the sequential VND
       Else k: = find k at random, k in 1 ... 6; // find k at random if we choose to apply the random VND
       EndIf
       x':= Apply neighborhood N_k to x;
       If (f (x)' < f (x)) Then x: = x';   k: = 0; FD = true; Update Hole_List_s and Isolated _List_s;
       EndIf
     Until all neighborhood structures are used
       If (FD = = false) then ho += 1;           //if we cannot correct the current hole we pass to the next one in the Hole_List_s
       EndIf
   EndWhile
 Until no improvement is obtained
 Repeat                                          // Repeat the search procedure to eliminate isolated lessons in the subgroups s
   Is= 0;                                        // counter of isolated lesson in the Isolated_List_s
   While (is <  Isolated_List_s.count and Isolated_List_s.count > 0)
     k=6; FD = false;                            // FD =true if we correct the current isolated lesson else FD = false
      Repeat
        If (Seq-VND) then k: = k +1;             // increment k if we choose to apply the sequential VND
        Else k: = find k at random, k in 7 ... 11; // find k at random if we choose to apply the random VND
        EndIf
        x':= Apply neighborhood N_k to x;
         If (f (x)' < f (x)) then x: = x';   k: = 0; FD = true; Update Hole_List_s and Isolated_List_s;
        EndIf
      Until all neighborhood structures are used
        If (FD = = false) then is += 1           //if we cannot correct the current isolated lesson, pass to the next one in the
        EndIf                                      Isolated _List_s
   EndWhile
 Until no improvement is obtained
EndFor
```

Fig. 2. Variable Neighborhood Descent algorithm

The VND algorithm treats each subgroup of students and eliminates corresponding holes then isolated sessions. The VND algorithm have been really integrated into the FEMSS timetabling tool. Numerical results are shown in the next session.

4 Numerical results

The program was coded using VB.Net 2010 with the aid of the SQL Server 2005 as a database management system. Table 2 shows the numerical results of the 3 algorithms with 2 search procedures on six instances (semesters 2012.1, 2012.2, 2013.1, 2013.2, 2014.1 and 2014.2) in comparison to the administration solution.

Table 2
Comparison between Initial solution and different solutions provided by the combination of the three algorithms with the two search procedures.

Instances	Initial Sol	Algorithm 1						Algorithm 2						Algorithm 3						IMPROVEMENT
		Seq-VND		Rand-VND				Seq-VND		Rand-VND				Seq-VND		Rand-VND				
		Cost	Cpu	Best	Cpu	Average	Standard deviation	Cost	Cpu	Best	Cpu	Average	Standard deviation	Cost	Cpu	Best	Cpu	Average	Standard deviation	Deviation from the initial solution (%)
2012.1	237.00	142.00	01:08	146.00	02:06	146.80	0.84	**141.00**	02:16	145.00	05:22	145.60	0.89	**141.00**	03:32	141.00	05:25	141.80	0.84	40.51
2012.2	223.00	69.00	00:59	69.00	01:34	69.40	0.55	69.00	01:31	68.00	03:25	68.40	0.55	69.00	02:57	67.00	03:29	**67.40**	0.55	69.78
2013.1	365.00	147.00	01:33	148.00	02:43	148.60	0.55	146.00	03:58	141.00	09:44	**142.00**	1.00	146.00	05:40	146.00	09:50	147.20	1.10	61.10
2013.2	300.00	143.00	04:43	143.00	04:35	144.00	0.71	143.00	04:14	295.00	10:56	295.60	0.55	183.00	07:24	141.00	09:04	**142.00**	1.00	52.67
2014.1	399.00	366.00	01:25	366.00	02:36	366.80	0.45	366.00	03:38	300.00	11:58	300.60	0.89	**198.00**	05:06	199.00	07:55	200.60	1.34	50.38
2014.2	331.00	325.00	00:59	325.00	02:16	326.20	1.10	325.00	03:45	325.00	09:13	325.80	0.45	**192.00**	04:23	192.00	09:24	193.00	1.00	41.99
Average	309,20	198.67	02:19	199.50	03:03	200.30		198.33	03 :22	212.33	08:43	213.00		154.83	05:23	147.67	07:51	148.67		52.74

Two criteria are considered to make the comparison: the sum of Holes and isolated lessons, and the computation time (CPU). We deduce that the proposed algorithms outperform the initial solution, and Algorithm 3 with Seq-VND procedure outperforms the other combinations, with the elimination of 52,47% of holes and isolated sessions on average.

5 Conclusions and perspectives

A real world problem of educational timetabling related to the Faculty of Economics and Management Sciences at Sfax (FEMSS) is tackled. A variable Neighborhood Descent approach is investigated to improve the feasible solution provided by the administrative staff by reducing the number of holes and the number of isolated lessons. Experimental results demonstrate the efficiency of our approach.
Further work will be on the extension of our approach to reduce the number of occupied Saturdays and ensure a laps of times for lunch.

References

[1] Abdullah, S. and H. Turabieh, *On the use of multi neighbourhood structures within a tabu-based memetic approach to university timetabling problems*, Information Sciences **191** (2012), pp. 146–168.

[2] Brito, S. S., G. H. Fonseca, T. A. Toffolo, H. G. Santos and M. J. Souza, *A sa-vns approach for the high school timetabling problem*, Electronic Notes in Discrete Mathematics **39** (2012), pp. 169–176.

[3] Burke, E. K., J. Mareček, A. J. Parkes and H. Rudová, *A supernodal formulation of vertex colouring with applications in course timetabling*, Annals of Operations Research **179** (2010), pp. 105–130.

[4] Dammak, A., A. Elloumi and H. Kamoun, *Lecture timetabling at a tunisian university*, International Journal of Operational Research **4** (2009), pp. 323–345.

[5] Dammak, A., A. Elloumi, H. Kamoun and J. A. Ferland, *Course timetabling at a tunisian university: A case study*, Journal of Systems Science and Systems Engineering **17** (2008), pp. 334–352.

[6] Di Gaspero, L. and A. Schaerf, *Neighborhood portfolio approach for local search applied to timetabling problems*, Journal of Mathematical Modelling and Algorithms **5** (2006), pp. 65–89.

[7] Gunawan, A., K. M. Ng and K. L. Poh, *A hybridized lagrangian relaxation and simulated annealing method for the course timetabling problem*, Computers & Operations Research **39** (2012), pp. 3074–3088.

[8] Hansen, P. and N. Mladenović, *Variable neighborhood search*, in: E. K. Burke and G. Kendall, editors, *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, Springer US, Boston, MA, 2005 pp. 211–238.

[9] Lü, Z. and J.-K. Hao, *Adaptive tabu search for course timetabling*, European Journal of Operational Research **200** (2010), pp. 235–244.

[10] Özcan, E., A. J. Parkes and A. Alkan, *The interleaved constructive memetic algorithm and its application to timetabling*, Computers & Operations Research **39** (2012), pp. 2310–2322.

[11] Perzina, R., *Solving the university timetabling problem with optimized enrollment of students by a self-adaptive genetic algorithm*, in: E. K. Burke and H. Rudová, editors, *Practice and Theory of Automated Timetabling VI: Lecture notes in computer science*, Springer Berlin Heidelberg, 2007 pp. 248–263.

[12] Phillips, A. E., H. Waterer, M. Ehrgott and D. M. Ryan, *Integer programming methods for large-scale practical classroom assignment problems*, Computers & Operations Research **53** (2015), pp. 42–53.