

# Multi-Criteria Soft Constraints in Timetabling

Hana Rudová, Miroslav Vlk

Faculty of Informatics, Masaryk University  
Botanická 68a, Brno 602 00, Czech Republic  
{hanka,xvlk@fi.muni.cz}

## 1 Introduction

Timetabling problem is an optimization problem where various criteria must be considered and where it is not possible to satisfy all of the requirements. We would like to show a framework which allows to include more criteria in a declarative style with the help of soft constraints. Each objective criteria can be represented by a set of soft constraints. Such set includes requirements of the same type, for example preferences on assignments of events to times or preferences on possible overlapping of events in time. This set can be associated with a weight representing importance of the criteria. The flexible assignment of the weight allows to emphasize or suppress particular criteria and obtain desirable solution.

Our work is motivated by a real timetabling problem at Purdue University, USA [7]. This large scale problem includes about 820 classes. About 90,000 course requests by almost 30,000 students must be considered. A major objective in developing an automated system is to minimize the number of potential student course conflicts. There are also various requirements which are not possible to satisfy and some optimization is needed. For example, back-to-back classes of one student or instructor should not be scheduled in two distant buildings, or we should allow for a balanced distribution of times for particular subjects.

Let us start with some basic definitions of constraint programming approach. A constraint satisfaction problem (CSP) [2] is a triple  $(V, D, C)$  where  $V$  is a finite set of variables,  $D$  is a set of possible values for variables (domain), and  $C$  is a finite set of constraints restricting the values of variables. A solution to a CSP is a complete assignment of the variables that satisfies all the constraints. It may not be possible to find a solution which would satisfy all of the (hard) constraints. In this case, we can consider soft constraints which may not be necessarily satisfied. Various approaches to soft constraints have been introduced and studied [1]. We will consider the weighted constraints [3] which are associated with some cost. The goal is to find such assignment of values to variables that the total cost of unsatisfied constraints is minimal. This approach includes optimization component but the multi-criteria decision making is not considered. Our goal is to show an extension of weighted soft constraints which is able to handle multiple criteria naturally.

We will first describe particular criteria and define their evaluation function. Next section will propose evaluation function for all criteria with the help of inconsistency counters. Section 4 specifies particular soft constraints and their propagation algorithms based on inconsistency counters. Some preliminary experiments are presented in Section 5.

## 2 Criteria

We can define unary soft constraints expressing importance of particular values in the domain of variable. In timetabling problem, two distinct criteria can be represented with the help of such soft constraints. The first criteria introduces a desirable *time placement* and the second criteria specifies a desirable *classroom placement* for classes. Both criteria are defined on each assignment  $\theta$  of values to variables and they sum up initial time preferences  $time(t_i, \theta(t_i))$  and initial room preferences  $room(r_i, \theta(r_i))$  for all classes  $i$ :

$$f_{time}(\theta) = \sum_{\forall i} time(t_i, \theta(t_i)) \ , \quad f_{room}(\theta) = \sum_{\forall i} room(r_i, \theta(r_i)) \ .$$

Other soft constraint can state that the two classes  $i$  and  $j$  represented by variables for start times share the  $c_{ij}$  students. Naturally the  $c_{ij}$  gives the cost of this constraint because  $c_{ij}$  students do not want scheduling of these classes at the same time. All soft constraints of this type are included in the *student conflict* criteria. To define it, we will expect that  $overlap(\theta(t_i), \theta(t_j)) = 1$  holds if  $i$  and  $j$  overlap in the assignment  $\theta$ . Otherwise  $overlap(\theta(t_i), \theta(t_j)) = 0$  holds. Now the student conflict criteria can be written as

$$f_{sc}(\theta) = \sum_{\forall i,j:i < j} c_{ij} \times overlap(\theta(t_i), \theta(t_j))$$

A similar constraint with additional variables for classrooms can express the *student distance* criteria. It means that the scheduling of the two classes  $i, j$  in consequent time periods (back-to-back classes) should discourage placement in very distant buildings because the  $c_{ij}$  students share these classes. We again need an additional function  $btb(\theta(t_i), \theta(t_j))$  which is equal either to 1 for back-to-back classes or to 0 for remaining classes. Function  $far(\theta(r_i), \theta(r_j))$  expresses how distant are two rooms  $i, j$ . For student conflicts, it corresponds to 0 if they are close enough and equals to 1 if they are too distant<sup>1</sup>. Now the student distance criteria is

$$f_{btb}(\theta) = \sum_{\forall i,j:i < j} c_{ij} \times btb(\theta(t_i), \theta(t_j)) \times far(\theta(r_i), \theta(r_j)) .$$

In our timetabling problem, classes of different subjects/departments like Math or Physics are included. A balanced distribution of times for particular departments is needed not to assign to one subject much worse or better times than to others (for example, one subject should not have all late afternoon lectures). The *department spread* criteria

$$f_{dept}(\theta) = \sum_{\forall D} \sum_{k=est}^{lst} \max(0, current(\theta, D, k) - limit(D)) .$$

is associated with the soft constraints which mean that at most  $limit(D)$  classes of one department  $D$  can be scheduled at the same time. Function  $current(\theta, D, k)$  computes the number of classes scheduled by  $\theta$  at each time unit of a day  $k$  from  $est$  to  $lst$ .

### 3 Inconsistency Counters

We have defined several criteria which needs to be extended to some soft constraints [1]. Now we would like to reformulate particular criteria to allow inclusion in constraint propagation algorithms for soft constraints [4]. Our current proposal is based on partial forward checking algorithm [3] which is the basic algorithm applying so called inconsistency counts. Inconsistency counters store information about the current number of violations for each value in the actual domain of the variable. Once all variables are assigned, only one value remains in the domain of each variable. Summarization of all remaining inconsistency counters (one per each variable) gives evaluation of the solution. In this section, we redefine all of the criteria using inconsistency counters. Consequently they can be used in constraint propagation algorithms of particular soft constraints.

Initially the value of inconsistency counter for time variable  $t_j$  and each its value  $a$  is set to  $time(t_j, a)$ . Similarly we initialize inconsistency counters for each room variable by  $room(r_j, b)$ .

Let  $\theta$  is some current partial assignment and  $A(\theta)$  is the set of variables assigned by  $\theta$ .  $V \ll v$  will denote that the variables from the set  $V$  were assigned earlier than the variable  $v$ .

Now the number of inconsistencies for the student conflict criteria can be written as

$$IC_{sc}(t_j, a) = \sum_{t_i \in A(\theta): \{t_i\} \ll t_j} c_{ij} \times overlap(\theta(t_i), a) .$$

The student distance criteria can propagate into inconsistency counter if only one of the variables  $t_i, r_i, t_j, r_j$  remains unassigned. Let us expect we have assigned  $t_i, r_i, r_j$ . Then we can derive

$$IC_{btb}(t_j, a) = \sum_{\{t_i, r_i, r_j\} \ll t_j} c_{ij} \times btb(\theta(t_i), a) \times far(\theta(r_i), \theta(r_j)) .$$

---

<sup>1</sup>Let us note that the function  $far$  could express various degrees of acceptability.

If one of the room variables ( $r_j$ ) is unassigned we have

$$IC_{btb}(r_j, a) = \sum_{\{t_i, t_j, r_i\} \ll r_j} c_{ij} \times btb(\theta(t_i), \theta(t_j)) \times far(\theta(r_i), a),$$

Now let us define inconsistency counters for the department spread criteria. Let  $periods(i, b)$  gives all time periods of the class  $i$  starting at time period  $b$ . The current number of classes  $j$  from the department  $D$  which were assigned before the class  $i$  using  $\theta$  to time period  $a$ , is

$$current(\theta, D, a, i) = \|\{j | j \in D \wedge \{t_j\} \ll t_i \wedge a \in periods(j, \theta(t_j))\}\| .$$

The limit value  $limit(D)$  for each department  $D$  can be compared with the maximal value of  $current(\theta, D, a, j)$  for all time periods of course  $j$  starting at  $a$

$$max\_current(j, a) = \max_{b \in periods(j, a)} \{current(\theta, D, b, j)\}$$

where  $j \in D$  holds. As a consequence we get

$$IC_{dept}(t_j, a) = \begin{cases} 0 & max\_current(j, a) < limit(D) \\ 1 & max\_current(j, a) = limit(D) \end{cases}$$

Unique inconsistency counter was defined for each variable and its value. Inconsistency counter for the time variable  $t_j$  and its value  $a$  equals to

$$IC(t_j, a) = w_{time} \times time(t_j, a) + w_{sc} \times IC_{sc}(t_j, a) + w_{btb} \times IC_{btb}(t_j, a) + w_{dept} \times IC_{dept}(t_j, a)$$

and, for the room variable  $r_j$  and its value  $b$ , it corresponds to

$$IC(r_j, b) = w_{room} \times room(r_j, b) + w_{btb} \times IC_{btb}(r_j, b) .$$

$w_{time}, w_{sc}, w_{btb}, w_{dept}$ , and  $w_{room}$  are weights expressing importance of particular criteria.

Finally we obtain evaluation for each assignment  $\theta$

$$F(\theta) = \sum_{\forall j \text{ where } \theta(t_j)=a \wedge \theta(r_j)=b} IC(t_j, a) + IC(r_j, b)$$

## 4 Soft Constraints

We have implemented all of the criteria as the soft constraints of soft CLP( $FD$ ) solver [6]. This solver allows to maintain inconsistency counters for each time and room variable. Inconsistency counter is stored for each value which is in the current domain of the variable. It is initialized by the values of time placement and room placement criteria.

Student conflict criteria is implemented with the help of

`soft_disjunctives(T_i, D_i, ListT_j, ListD_j, ListC_ij)`

constraint posted for each class  $i$ . The class  $i$  is represented by the time variable  $T_i$  and its duration  $D_i$ . The lists `ListT_j` and `ListD_j` store time variables and durations for classes  $j$  which share some students with the class  $i$ . The list `ListC_ij` contains the list of numbers of shared students  $C_ij$  between class  $i$  and classes  $j$ . The solver for this constraint is activated once when the  $T_i$  is assigned a value. It increases inconsistency counters for time variables  $T_j$  from `ListT_j` which still have not assigned value and which could overlap with  $T_i$ . So, the values of  $T_j$  overlapping with  $T_i$  are computed and their inconsistency counters are increased by  $w_{sc} \times C_ij$ .

The soft constraint for student distance criteria has a similar structure to `soft_disjunctive` constraint

`soft_btbdist(T_i, R_i, D_i, ListT_j, ListR_j, ListD_j, ListC_ij)`

Meaning of the variables remains to be same. In addition, we have corresponding room variables  $R_i$  and  $ListR_j$ . This soft constraint is activated two times. When the first of the variables  $T_i$  and  $R_i$  is instantiated we are looking for classes  $j$  which have assigned both  $T_j$  and  $R_j$ . When the second of the variables  $T_i$  and  $R_i$  is instantiated, the classes  $j$  with just one assigned variables from  $T_j$  and  $R_j$  are found. In all cases, we have three variables assigned and we can check for an increase one of the  $IC_{btb}$  counters.

Department spread criteria is implemented via

```
soft_dept(Dept, ListT_i, ListD_i, Limit)
```

constraint posted for each department  $Dept$ . The lists  $ListT_i$  and  $ListD_i$  contains time variables and durations of all classes from the department  $Dept$  and  $Limit$  is the limit value for the  $Dept$ . The constraint is activated any time some time variable  $T_i$  from  $ListT_i$  is instantiated. As a consequence, the current number of classes assigned during time periods of  $T_i$  is increased. Also inconsistency counters are increased for classes  $j$  from  $ListT_i$  which have assigned just  $Limit$  classes for some value. However, this increase could be processed for each value of each time variable just once.

## 5 Experiments

Let us present some of our preliminary experiments we achieved on Purdue Spring 2005 data set. This data set includes 821 classes and 50 classrooms. The total amount of joint enrollments is 97371.

The goal of our experiments was to set different weights to particular criteria and look for comparison among them. Both experiments at Figure 1 show evaluation for achieved time placement and student conflicts

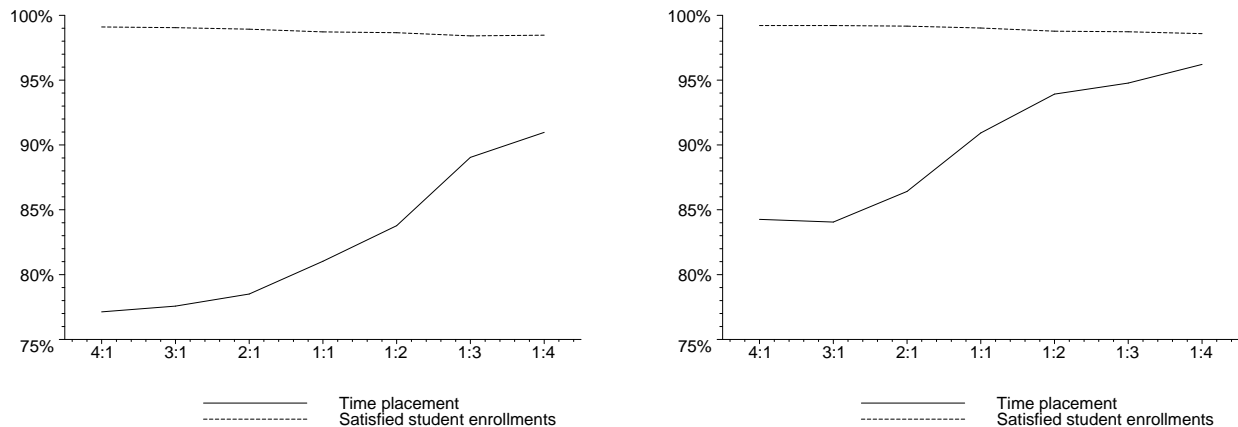


Figure 1: Comparison of criteria on time variables

(satisfied student enrollments corresponds to the supplement of student conflicts). Both criteria have set different weights  $w_{sc} : w_{time}$  ranging from 4 : 1 to 1 : 4. For example, 4 : 1 means that the student conflict criteria was four times more important than the time placement criteria. The first experiment at Figure 1 (left graph) shows the case where the weights for the student distance and department spread criteria are set to 1 (i.e.,  $w_{btb} = w_{dept} = 1$ ). The other experiment does not consider any of them (i.e.,  $w_{btb} = w_{dept} = 0$ ).

We can see that all criteria are strongly tight together. We are able to improve evaluation for time placement while worsening the number of student conflict and vice versa. Also we can see that the department spread criteria has a strong influence on the time placement. The department spread criteria is able to overweight some of the time placement preferences to achieve a better spread of classes of each department through the time.

## 6 Conclusion and Future Work

We have extended our soft  $CLP(FD)$  solver to handle various criteria from timetabling problems. This general proposal is based on inconsistency counters. It allows addition of other criteria by means of new types of soft constraints. We run some preliminary experiments on large scale Purdue timetabling problem. Our goal was to show that multi-criteria decision making could be implemented with the help of our soft constraints.

Our future work will consist in extension of constraint propagation algorithms for particular criteria. Next we will compare our results with local search-based solver for Purdue University timetabling problem [5].

## References

- [1] Stefano Bistarelli, Ugo Montanari, and Francesca Rossi. Semiring-based constraint solving and optimization. *Journal of ACM*, 44(2):201–236, March 1997.
- [2] Rina Dechter. *Constraint Processing*. Morgan Kaufmann Publishers, 2003.
- [3] Eugene C. Freuder and Richard J. Wallace. Partial constraint satisfaction. *Artificial Intelligence*, 58:21–70, 1992.
- [4] Javier Larrosa and Thomas Schiex. Solving weighted CSP by maintaining arc consistency. *Artificial Intelligence*, 159(1–2):1–26, 2004.
- [5] Tomáš Müller, Hana Rudová, and Roman Barták. Minimal perturbation problem in course timetabling. In Edmund Burke and Michael Trick, editors, *Practice and Theory of Automated Timetabling, Selected Revised Papers*. Springer-Verlag LNCS, 2005. To appear.
- [6] Hana Rudová. Soft  $CLP(FD)$ . In Susan Haller and Ingrid Russell, editors, *Proceedings of the 16th International Florida Artificial Intelligence Symposium, FLAIRS-03*. AAAI Press, 2003.
- [7] Hana Rudová and Keith Murray. University course timetabling with soft constraints. In Edmund Burke and Patrick De Causmaecker, editors, *PATAT 2002 — Proceedings of the 4th international conference on the Practice And Theory of Automated Timetabling*, pages 73–89, 2002.