An Ensemble of Neural Networks as Part of a GA-based Model to Solve the School Timetabling Problem

Janis Zuters

University of Latvia, 19 Raina blvd.,
LV-1050 Riga, Latvia
janis.zuters@lu.lv

Abstract. Timetabling is a highly constrained combinatorial problem, which is often solved by evolutionary techniques such as genetic algorithms (GAs). This article reviews a specific aspect of the school timetabling problem – examples of ready-made timetables could be useful for creating new ones. The author describes a GA-based model, which involves an ensemble of neural networks (neural ensemble) as part of the fitness function. The neural networks of the ensemble are trained on existing timetables to substitute, in part, for the hard-defined rules. Neural networks are introduced to diminish the effort of defining the criteria for the evaluation of timetables. Experimentation on the proposed timetabling model is still being conducted, and the paper describes the model in detail with respect to the effect of usage of neural networks.

1 Introduction

The timetabling problem is a highly constrained combinatorial problem and has an NP-complete degree of complexity [1]. There are various kinds of timetabling problems that differ in terms of constraints to be observed and objectives that are involved.

GAs are often applied to solve optimization problems such as the timetabling problem [1], [2]. An important requirement for the problem domain to be solved using GAs is the possibility to evaluate (rate) solutions at any phase of the evolutionary process [5]. There are problems for which evaluation of solutions is a complicated task. To build evaluation rules, strict criteria are required, and these could be hard to extract or even recognize.

The author proposes a GA-based timetabling model, where an ensemble of neural networks substitutes, in part, for the hard-defined rules of the fitness function. Neural networks are introduced to diminish the effort of defining the criteria for the evaluation of timetables.

To build the timetabling model, timetables of primary and secondary schools were explored, thus the proposed model is very specific and destined for use in such schools. As timetables of different schools are similar in terms of the sets of subjects involved, it makes the ready-made timetable to be useful to some extent in scheduling process of another school.

2 School timetabling problem

2.1 School Timetabling Problem – Constraint Satisfaction Problem

In general scheduling problem, events must be arranged around a set of time slots, so as to satisfy a number of constraints and optimize a set of objectives. Events to be arranged are lessons. Resources to be referred by constraints and objectives are classes, teachers and classrooms.

The school timetabling problem belongs to the class of constraint satisfaction problems (CSPs).

A CSP is defined by a set of variables, $X_1,X_2,...,X_n$, and a set of constraints $C_1,C_2,...,C_m$. Each variable X_i has a nonempty domain D_i of possible values. Each constraint C_i involves some subset of the variables and specifies the allowable combinations of values for that subset. A state of the problem is defined by an assignment of values to some or all of the variables $\{X_i=v_i,X_j=v_j,...\}$. A solution to a CSP is a complete assignment that satisfies all the constraints. Some CSPs also require a solution that maximizes an objective function. [3] For the school-timetabling problem, the variables X_i are lessons, and domains D_i are the sets of available timeslots for scheduling.

In this article, we focus on the objective function (or, in GA terminology, the fitness function). In timetabling terminology, constraints and objectives are referred to as hard and soft constraints respectively. [4] Hard constraints are mandatory in order to obtain consistent timetables. Soft constraints serve as criteria to evaluate timetables and, although they are not mandatory, they greatly influence the quality and usability of timetables. A portion of soft constraints is depicted in Table 1.

Table 1. A portion of soft constraints for school timetabling

#	Description of the constraint
1	Minimize the gaps between lessons for students
2	Minimize the gaps between lessons for teachers
3	As far as possible, align lessons to the beginning of the shift
4	Arrange lessons uniformly over the week
5	Observe limits for the number of consecutive lessons for teachers
6	Observe predefined recommendations as much as possible (e.g., teacher X prefers mornings)
7	Balance the layout of lessons in terms of themes (e.g., don’t schedule to many exact subjects in row)
8	Some lessons need more then one period

In order to clarify the further description of the proposed model, the timetabling problem has been simplified in the following respect: there are no subclasses or unified classes, which means that exactly the one and whole class participates in each lesson.

2.2 Genetic Algorithms to Solve the School Timetabling Problem

The general schema of GA-based timetabling. GAs are typical means to solve timetabling problems. Fig. 1 shows the schema for timetabling using GAs.

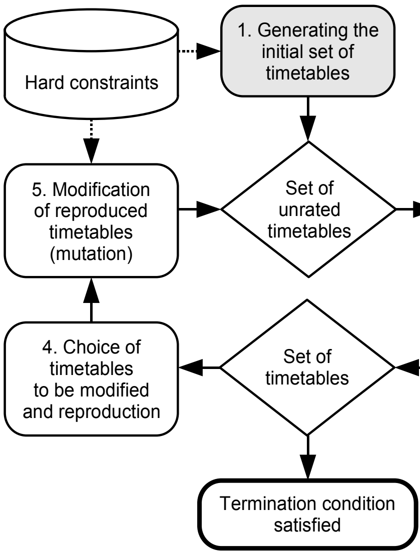

Fig. 1. The schema for GA in timetabling [5]

The fitness function. The fitness function is some kind of realization of the given soft constraints. For similar problems, the fitness function is usually designed as a penalty function (the lower the value, the better the solution). Assuming that a separate evaluation function is built for each soft constraint, the total fitness function φ(·) will be described as follows:

$$\varphi(\mathrm{T}) = \sum_{i=1}^{c} \alpha_i \varphi_i(\mathrm{T}), \tag{1}$$

where $\varphi_i(\cdot)$ – the evaluation function according to the constraint i; α_i – the weight of the constraint i; T – the timetable to evaluate; the number of soft constraints.

As the timetable consists of a number of timetables of classes, the evaluation function for a separate constraint will usually be in the following form:

$$\varphi_i(\mathrm{T}) = \sum_{k=1}^{g} \psi\left(\varphi_{ik}(\mathrm{T}_k)\right), \tag{2}$$

where $\varphi_{ik}(\cdot)$– the evaluation function for the class k according to the constraint i; ψ(·) – the exaggeration level of the penalty value; T_k – the timetable for the class k; g – the number of classes.

3 Neural Networks as Part of a Timetabling Model

3.1 Existing Timetables – Source of Information

Existing timetables (even of different schools) contain useful information for creating new ones. The existing timetables encode the manner in which timetables are built.

Representation of a timetable. A timetable is a set of lessons arranged over time. A lesson includes four types of attributes – students, teacher, room, and subject. Thus, with respect to these attributes, we can look at timetables in four aspects. Unfortunately, not all information from existing timetables is useful for creating other timetables. The sets of classes, teachers and rooms differ among various schools, which eliminate the possibility of direct use of the information of those aspects. On the other hand, the sets of available subjects are similar, thus the information on subjects could be of use in creating other timetables.

The timetable for one class with respect to subjects can be represented as follows:

$$T_k = \langle s_i \rangle_{i=1}^{p}, \tag{3}$$

where s_i – the subject for class k in period i (can be null); p – the number of available periods (e.g., in one week).

(3) represents the arrangement of subjects over the time. Can this extract of information be useful for scheduling other timetables? In case of a human scheduler, the answer is – yes. Exploring the substance of the soft constraints depicted in Table 1, we could realize that the arrangement of subjects plays some role, at least an indirect one, in ensuring respect for these constraints.

Subject category instead of subjects. To evaluate timetables according to the soft constraints listed in Table 1, it would be enough to some extent to use subject categories instead of subjects to represent a timetable (e.g., see the constraint #7).

Let's introduce the second simplification. Assume that each subject belongs to some category, so we have the following function:

$$\sigma : Subjects \to SubjectCategories\,. \tag{4}$$

(E.g., mathematics belongs to the exact subjects, and history to the humanities.)

Then the timetable for one class instead of (3) would be represented as:

$$T_k = \langle \sigma(s_i) \rangle_{i=1}^{p}. \tag{5}$$

Using the subject categories instead of subjects will facilitate the computation (the number of subject categories is significantly lower then the number of subjects), as well as normalize and standardize the representation of timetables (timetables come from different schools, and there are subjects that are similar, so there's no need to code them differently). Moreover, categorization of subjects provides the scheduling algorithm with the additional information on relationship among various subjects.

3.2 The Fitness Function Softened by Neural Networks

The expected effect of using neural networks. The possible benefits of the proposed model where the neural ensemble is used as part of the fitness function can be viewed from two aspects:

1. replace one or more soft constraints by the neural ensemble, thus facilitating the effort of defining the constraints,
2. assuming the complete set of constraints, it is hard and practically impossible to define – complement existing constraints with the neural ensemble in order to improve upon timetable evaluation.

The effect of the 2nd aspect is very hard to verify, as it would need a massive estimation of experts. Therefore, the proposed model will be examined just according to the 1st aspect.

Proposing the softened fitness function. The idea is to replace some of hard-defined evaluation functions φ_i by the neural ensemble φ_0 trained on existing timetables. The proposed softened fitness function $\varphi^{F+}(\cdot)$, in which a subset of soft constraints F is replaced by the neural ensemble, will be depicted as follows:

$$\varphi^{F+}(\mathrm{T}) = \varphi^{F}(\mathrm{T}) + \alpha_0 \varphi_0(\mathrm{T}), \qquad \mathbf{(6)}$$

where $\varphi_0(\cdot)$ – the evaluation function realized by the neural ensemble; α_0 – the weight of the neural ensemble; $\varphi^F(\cdot)$ – the reduced fitness function (eliminating constraints of F):

$$\varphi^{F}(\mathrm{T}) = \sum_{i=1}^{c} \begin{Bmatrix} 0; & \mathit{if}\ i \in F \\ \alpha_i \varphi_i(\mathrm{T}); & \mathit{otherwise} \end{Bmatrix}. \qquad \mathbf{(7)}$$

The exploited attribute “softened” (for the fitness function) comes from “soft computing”, not from “soft constraints”.

To make out the effect of using neural networks in timetabling, experiments on GA-based timetabling with the fitness functions $\varphi^F(\cdot)$ and $\varphi^{F+}(\cdot)$ will be conducted, and then the results will be evaluated according to the benchmark fitness function $\varphi(\cdot)$. As the desirable effect, the experiments using φ^{F+} would show better results then those using φ^F.

The experimentation will show whether neural networks, which are trained on existing timetables, are capable evaluating timetables or, more precisely, of substituting hard defined evaluation functions to some extent.

The only source of information is ready-made timetables of various schools from recent years. A set of existing and valid timetables is incomplete as the training set for evaluative neural networks, because it contains just positive patterns. This will require modifications in the learning algorithm or architecture of neural networks. The construction of neural networks trained on positive patterns is described in [6]. To obtain evaluative neural networks, supervised neural models will be used (such as multi-layer perceptrons or radial-basis function networks).

Neural ensemble instead of a single neural network. As timetables of different schools are of very different sizes, the construction of a single neural network (as part of the fitness function) to evaluate timetables would be cumbersome. To standardize input, the idea is to use a different neural network for each grade of classes (1..12).

Each class belongs to some grade, as represented by the following function:

$$\gamma : Classes \to \{1,2,...,12\}. \tag{8}$$

Thus φ_0 is realized by several neural networks and would look as follows:

$$\varphi_0(\mathrm{T}) = \sum_{k=1}^{g} \psi\left(\varphi_{0,\sigma(k)}(\mathrm{T}_k)\right), \tag{9}$$

where $\varphi_{0,m}(\cdot)$ – the evaluation function realized by the neural network for grade m; k – the class for which the timetable is being evaluated.

4 Experimental Work

Experimentation on the proposed model is still being conducted. This section describes the already performed and expected activities on this item.

4.1. Summary of the Progress of the Research

- The core GA-based timetabling system has been realized.
- The evaluative neural network model has been designed [6].
- A portion of soft constraints has been realized.
- A version of timetable representation for neural processing has been proposed, and the evaluative neural network model has been adapted for timetable evaluation.
- Preliminary experimental results have been obtained.

4.2. The Algorithm for the Experimentation to be Conducted

Let's describe the GA to be used in experiments with the following function:

$$\chi : L^* \times F^* \times Boolean \to \mathrm{T}^*, \tag{10}$$

which takes an unscheduled timetable as the first parameter and returns a scheduled one; $\chi(\cdot,F,FALSE)$ denotes GA run with the fitness function φ^F, but $\chi(\cdot,F,TRUE)$ – with φ^{F+}.

Provided that we have a set of available timetables for training and a set of realized constraints F, the algorithm for the experimental process is described in Fig. 2.

```
do n times
  train neural ensemble on available timetables
  do m times
```

```
choose a set of constraints F to eliminate
choose weights α={α[0],α[1],...α[c]}
X:=χ(L,F,FALSE)
Y:=χ(L,F,TRUE)
record {F,α,L,φ(X),φ(Y)} for further analysis
```

Fig. 2. The algorithm of experimentation on a fixed unscheduled timetable L.

4.3. Preliminary Experimentation

This section describes the configuration of the experimental environment, and the preliminary results of experiments, in which the capability of neural ensemble to evaluate school timetables was tested.

Available training set. 10 different ready-made school timetables were available consisting of 208 different class timetables. Each class timetable has a fixed degree 1..12 and is represented like in (5). To provide such a representation, all the subjects were grouped into 12 subject categories. Each period of a timetable was represented as a tuple of 12 values corresponding to 12 subject categories (value 1 – if the subject category is represented in the period, 0 – if not). Thus a class timetable for one week is a tuple of 600 values (5 days $\times$ 10 periods per day $\times$ subject categories).

Configuration of the neural ensemble. The neural ensemble used in experiments consisted of 12 neural networks (9). Each neural network was an MLP (600 inputs, one hidden layer with 5 neurons, and one neuron in the output layer). Each neural network was trained on a different portion of patterns of the training set (according the grade). Each neural network was trained on positive patterns (of the proposed training set) only with desired value of 0 (see also [6]).

Configuration of the GA-based timetabling system. The size of population was 10. 2 individuals (timetables) were replaced in each cycle. Mutation was the only genetic operator. 4 functions were defined to evaluate timetables (simplified versions of soft constraints 1, 3, 4, 7 of Table 1).

The goal of the preliminary experimentation. Run GA-based timetabling experiments (without using neural networks within the fitness function). Compare the outputs of the 4 predefined evaluation functions φ_1, φ_3, φ_4, φ_7 (see Table 1) with the evaluation provided by the neural ensemble. Comparison was done through matching the sequences of timetables of a population ordered by evaluations. The goal of comparing the sequences is to make out the ability of a neural ensemble to substitute for the predefined evaluation functions.

Preliminary experimental results. Let's denote ideal match of sequences with 100%, and the average match between two random sequences with 0%. Evaluation functions φ_1, φ_3, φ_4 (see Table 1) showed the match with the neural ensemble approximately 0% (this means – no match). Evaluation function φ_7 showed the match with the neural ensemble approximately 20% (this means – insignificant match). Although the match is insignificant, it shows a certain ability of neural networks to participate in evaluation of timetables. At the same time, the results show that large additional experimental work is required to configure the system to be satisfactory.

5 Conclusion

Since ready-made timetables contain a lot of information about the manner in which timetables are built, there's reason to believe that incorporating neural networks into the timetabling system will bring the expected effect, making it possible to replace a subset of hard-defined soft constraints effectively.

The proposed model is characterized by high specialization, and it isn't regarded as a universal means to solve scheduling problems. The model is being developed with the emphasis on practical usability in primary and secondary schools.

Substantial additional experimental work is required to configure a neural ensemble to be able to extract features from existing timetables properly, but notable results have already been achieved.

References

1. Arous, N., Ellouze, N., Abdallah, S.: Evolutionary Potential Timetables Optimization by Means of Genetic and Greedy Algorithms. Proceedings of the 1999 International Conference on Information Intelligence and Systems (ICIIS) (1999)
2. Fernandes, C., Caldeira, J. P., Melicio, F., Rosa, A.: High School Weekly Timetabling by Evolutionary Algorithms. Proceedings of the 1999 ACM symposium on Applied computing (1999) 344-350
3. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. 2nd ed. Prentice-Hall, Inc. (2002)
4. Tam, V., Ting, D.: Combining the Min-Conflicts and Look-Forward Heuristics to Effectively Solve A Set of Hard University Timetabling Problems. Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'03) (2003)
5. Zuters, J.: An Adaptable Computational Model for Scheduling Training Sessions. Proceedings of Vidzeme University College "ICTE in Regional Development" (2005) 110-113
6. Zuters, J.: Towards Multi-Layer Perceptron as an Evaluator Through Randomly Generated Training Patterns. Proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases (AIKED 2006) (2006) 254-258