



Centro de Investigação Operacional

A Comparison of Discrete and Continuous Neural Network
Approaches to Solve the Class/Teacher Timetabling Problem

Marco Paulo Carrasco and Margarida Vaz Pato

CIO – Working Paper 4/2001

A Comparison of Discrete and Continuous Neural Network Approaches to Solve the Class/Teacher Timetabling Problem¹

M.P. Carrasco^{a,c}

^a*Escola Superior de Gestão, Hotelaria e Turismo, University of Algarve, Largo Eng. Sárra Prado, n.21, 8500 Portimão, Portugal*

M.V. Pato^{b,c}

^b*Instituto Superior de Economia e Gestão, Technical University of Lisbon, Rua do Quelhas n.6, 1200 Lisboa, Portugal*

^c*Centro de Investigação Operacional, University of Lisbon, Bloco C2, Sala 2.2.57, Campo Grande, 1749 Lisboa, Portugal*

Abstract

This study explores the application of neural network-based heuristics to the class/teacher timetabling problem (CTTP). The paper begins by presenting the basic CTTP characteristics in terms of hard and soft constraints and proposing a formulation for the energy function required to map the problem within the artificial neural network model. There follow two distinct approaches to simulating neural network evolution. The first uses a Potts mean-field annealing simulation based on continuous Potts neurons, which has obtained favorable results in various combinatorial optimization problems. Afterwards, a discrete neural network simulation, based on discrete winner-take-all neurons, is proposed. The paper concludes with a comparison of the computational results taken from the application of both heuristics to hard hypothetical and real CTTP instances. This experiment demonstrates that the discrete approach performs better, in terms of solution quality as well as execution time.

Key words: Timetabling, Metaheuristics, Neural Networks

Email addresses: pcarras@ualg.pt (M.P. Carrasco), mpato@iseg.utl.pt (M.V. Pato).

¹ This paper is part of the first author's PhD research

1 Introduction

Timetabling is a typical real world scheduling activity that arises at least once a year at every educational institution. The three most common educational timetabling problem categories are examination timetabling, course timetabling and class/teacher timetabling, as they are commonly called. The basic distinction occurs in terms of the elements to be scheduled, that is, exams, course options or regular lessons. The examination timetabling problem consists in scheduling the exams for a set of courses, over a limited time period, while avoiding the overlapping of exams for each student [20]. For a detailed description of this problem see the surveys made by Carter and Laporte [10] or Burke et al. [3]. In course timetabling problems, for each student a set of lectures is previously defined. Then all lectures included within the institution's set of courses must be scheduled in such a way that the overlapping of lectures for courses with students in common is minimized (vide, for instance, Downsland [13]; Kiaer and Yellen [19]). This problem arises in universities or other educational institutions with flexible curricula.

The current paper focuses on the last type, the class/teacher timetabling problem, CTTP for short. It considers the scheduling of a set of lessons (class/teacher assignments) subject to hard and soft constraints, and can be modelled in the context of combinatorial optimization. Such problems have, as a rule, been solved by using exact mathematical methods. However, tackling real world situations often requires the application of problem-specific heuristics to overcoming the difficulties of structure complexity or large scale often present in real instances.

In practice, the timetabling task is often performed manually, through a slow trial-and-error procedure. However, in recent decades, several metaheuristics have been proposed to tackle this task [22]. Among these, following the work of Hopfield and Tank [17], artificial neural networks have evidenced relative success in solving complex combinatorial optimization problems [23], including some timetabling issues [15],[16].

The paper begins with a description of the characteristics of the CTTP, in Section 2. In Section 3, the Hopfield and Tank neural network (HTNN) model is summarized and a specific energy function for the CTTP is proposed. Then, in Section 4, two heuristic methods of tackling the CTTP, based on the above neural network formulation, are investigated, namely, a continuous Potts mean-field annealing approach (CPMF) and a discrete winner-take-all neuron approach (DWTAN). Section 5 introduces a set of hard hypothetical and real CTTP instances and the results obtained from the application of the two neural heuristics are compared. Finally, Section 6 concludes with some considerations.

2 The Class/Teacher Timetabling Problem

The class/teacher timetabling problem, referred to as CTTP, is the subject of this paper. It is usually found in schools with less flexible curricula, as is the case of most secondary schools and some universities, where the majority of the pre-defined lessons for each course are compulsory for the classes in question and can therefore in no way overlap.

In terms of complexity, the CTTP shows a difficult structure which was proved to be NP-hard by Even et al. [14], even for some simplified versions. Furthermore, due to the large dimension often assumed by real-world instances, as well as the need to include additional specific-problem constraints, the CTTP has triggered a wide range of solution techniques. In fact, several methodologies have been proposed to tackle this type of timetabling problem, from graph theory based methods (Werra [28]), to binary programming (Tripathy [24]; Birbas et al. [2]) and constraint-based approaches (Yoshikawa et al. [26]), simulated annealing (Abramson [1]), tabu search (Costa [12]), genetic algorithms (Colorni et al. [11]), and neural networks (Gislén et al. [15],[16]; Carrasco and Pato [8]).

The specific class/teacher timetabling problem addressed in this work, already described in [9], can be defined as the optimization problem relative to the scheduling of a set of lessons (prior assignments of one or more rigid classes of students to one teacher and one subject) over a weekly set of time periods, using suitable rooms, while satisfying a broad spectrum of constraints. Due to the distinct nature of each constraint, two levels of importance can be identified, as follows.

- (i) The hard constraints assure the aspects related to the feasibility of a CTTP solution. For this reason all hard constraints must always be satisfied in order to obtain legal timetabling solutions. The hard constraints mentioned are described below.
 - (h1) The classes' curricula must be respected, i.e. all lessons assigned to each class are fully scheduled.
 - (h2) Each teacher and class is assigned to no more than one lesson and one room within one time period. This is a hard constraint common to all timetabling problems.
 - (h3) Lessons can be of different but fixed durations.
 - (h4) For pedagogical reasons, each subject is scheduled to be given no more than once a day.
 - (h5) Rooms are suitable for the lessons assigned. Each lesson may require a particular type of room, in terms of seating or special resources. For

example, as a rule, computer-related lessons require a room containing computers and specific software.

(h6) The teaching shift for each class is respected. In some schools there are teaching shifts that must always be satisfied. Such is the case when scheduling lessons for day or evening classes.

(h7) Each class and teacher must have a lunch break. This constraint imposes a free time period, with a minimum duration of one hour, to be used for the lunch activity.

(h8) Each class's and teacher's unavailability period is respected. In fact, classes and, far more often, teachers have periods of unavailability arising from periodic professional or academic commitments.

(ii) At a secondary level of importance one finds soft constraints, representing the optional aspects that both teachers and classes would like to have satisfied in their timetables. While a wide range of non-compulsory constraints may emerge in real life, this work only addresses the most important, common constraints, presented below.

(s1) The occurrence of gaps between teaching periods should be minimal for both classes and teachers. This condition is desirable, as it frees larger intervals of available time for study or other professional activities, such as research, besides resulting in more compact timetables. In addition, the number of days containing lessons should also be minimized, thus allowing teachers and classes to have some lesson-free days.

(s2) Class and teacher preferences should be satisfied. In almost every school, classes and, to a greater extent, teachers express their preferences, which should be met whenever possible.

(s3) The number of teacher and class shifts between different teaching locations should be minimized. In addition, for institutions with multiple, geographically distant teaching locations a minimum shift time should be respected to enable teachers and classes to move between different teaching places.

In this context the CTTP can be defined as the combinatorial optimization problem encountered in the determination of a timetable that satisfies all hard constraints, while attempting to satisfy the maximum number of soft constraints. For this reason, the degree of satisfaction of soft constraints is closely related to the final quality of the timetabling solution. Hence, the highest quality CTTP solution must not only assure (h1) to (h8) but also satisfy the maximum of the timetabling optional requirements expressed by (s1) to (s3).

Note that the above conditions define the most important features regarding our specific CTTP problem. However, other particularities can emerge from

real timetabling and be implemented by similar approaches to those proposed in the following sections.

3 CTPP Neural Network Encoding

The application of artificial neural networks to solve combinatorial optimization problems was first introduced by the seminal paper of Hopfield and Tank [17]. This work presented a neural network model for optimization purposes, characterized by a fully interconnected network with N binary neurons, whose basic features will now be briefly described. Neuron i has a net input value U_i and an output value or state V_i . The input value U_i results from the weighted sum of output values arising from the other neurons, added to a negative bias current I_i :

$$U_i = \sum_{j \neq i}^N w_{ji} V_j + I_i, \quad (1)$$

where, w_{ji} is the weight (synapse connection) from neuron j to i . The output V_i , which is bounded by zero and one, is calculated from U_i by an activation function. Basically there are two types of model versions of the Hopfield and Tank neural network (HTNN): discrete versions using a hard limiter activation function, and continuous versions, which traditionally use a sigmoid-type activation function and a gain parameter.

Hopfield and Tank proved that, for a symmetrical weight matrix ($w_{ij} = w_{ji}$), the neural network converges to stable states, that is, to local minimum values of the neural energy function (2):

$$E(\bar{V}) = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N w_{ij} V_i V_j - \sum_{i=1}^N I_i V_i \quad (2)$$

This minimization ability plays a central role in tackling combinatorial optimization problems through neural networks. For this purpose, one must formulate an appropriate energy function, whose minimum value corresponds to the optimum of the optimization problem. Usually this energy function derives from the constraints of the problem in the form of a penalized sum, added to the objective expression. Using such an approach requires a careful selection of the penalizing parameters, to obtain a balance between constraints and objective terms.

For the CTPP case and, on the assumption that L lessons are to be scheduled (involving prior assignments of T teachers to C classes), in P time periods

and R rooms, the basic HTNN requires $L \times P \times R$ generic two-state neurons [7].

However, using a particular factorization scheme proposed in [16], the neural network for the CTPP only requires $L \times P$ lesson-period neurons, given by V_{lp}^x , and $L \times R$ lesson-room neurons, V_{lr}^y . This method is far more efficient than the previous basic encoding as it significantly reduces the number of neurons involved. Furthermore, it also allows the direct implementation of some of the hard CTPP constraints mentioned in Section 2 (more precisely constraints (h5), (h6) and (h8)), through an initial freezing of the respective neurons in an appropriate state.

The remaining CTPP hard and soft constraints are used to define the energy function, with the exception of constraint (h1) which will be addressed in the next section. Constraints (h2) and (h3) are implemented simultaneously using penalty terms E_1 and E_2 . First, to prevent room occupation conflicts the following penalty term is included in the energy function, to be minimized afterwards:

$$E_1 = \frac{1}{2} \sum_{p=1}^P \sum_{r=1}^R \sum_{\substack{l, l'=1 \\ l \neq l'}}^L V_{lr}^y V_{l'r}^y S_{lp} S_{l'p} \quad , \quad (3)$$

where $S_{lp} = \sum_{k=0}^{Dur(l)-1} V_{l,p-k}^x$ and $Dur(l)$ gives the duration of lesson l .

In order to prevent class and teacher time conflicts, the energy component E_2 is defined as

$$E_2 = \frac{1}{2} \sum_{p=1}^P \sum_{\substack{l, l'=1 \\ l \neq l'}}^L Q_{ll'}^{teca} S_{lp} S_{l'p} \quad , \quad (4)$$

where $Q_{ll'}^{teca} = \begin{cases} 2 & \text{if lessons } l \text{ and } l' \text{ share the same teacher and class} \\ 1 & \text{if lessons } l \text{ and } l' \text{ share the same teacher or class} \\ 0 & \text{otherwise.} \end{cases}$

In addition, the energy term E_3 attempts to avoid scheduling more than one lesson of the same subject on the same day (constraint (h4)):

$$E_3 = \frac{1}{2} \sum_{c=1}^C \sum_{\substack{l, l' \in LC^{lass}(c) \\ l \neq l'}} Q_{ll'}^{sub} \sum_{d=1}^D \left(\sum_{p \in TDay(d)} S_{lp} \right) \left(\sum_{p \in TDay(d)} S_{l'p} \right) \quad , \quad (5)$$

where $LClass(c)$ is the set of lessons attended by class c , $TDay(d)$ represents the set of time-periods of day d , D is the number of teaching days in the week,

$$\text{and } Q_{ll'}^{sub} = \begin{cases} 1 & \text{if lessons } l \text{ and } l' \text{ belong to the same subject} \\ 0 & \text{otherwise.} \end{cases}$$

To oblige satisfaction of (h7), related to the existence of lunch breaks, two auxiliary sets of variables expressing lesson participation of each class or teacher along the weekly time-periods, are defined as:

$$A_{cp}^{cla} = \sum_{l \in LClass(c)} S_{lp} \quad , \quad A_{tp}^{tea} = \sum_{l \in LTeacher(t)} S_{lp} \quad , \quad (6)$$

where $LTeacher(t)$ is the set of lessons taught by teacher t .

By using these variables, the energy term E_4 ensures that a lunch break is created each day d within the set of time periods defined by $TLunchF(d)$ and $TLunchL(d)$, as follows:

$$E_4 = \frac{1}{2} \sum_{d=1}^D \left[\sum_{c=1}^C \left(\prod_{p=TLunchF(d)}^{TLunchL(d)} A_{cp}^{cla} \right) + \sum_{t=1}^T \left(\prod_{p=TLunchF(d)}^{TLunchL(d)} A_{tp}^{tea} \right) \right] . \quad (7)$$

The next energy terms provide the minimization feature of the CTTP, guaranteeing satisfaction of the soft constraints. Once again, these energy terms measure each soft constraint cost within a CTTP solution.

The first soft constraint (s1) whose goal is time contiguity of the lessons - both for classes and teachers - is enforced by the addition of E_5 to the energy expression:

$$E_5 = \frac{1}{2} \sum_{c=1}^C \left(SumL_c^{cla} - \left(\sum_{d=1}^D \sum_{p=TDaYF(d)}^{TDaYL(d)-1} A_{cp}^{cla} A_{c,p+1}^{cla} \right) \right) + \frac{1}{2} \sum_{t=1}^T \left(SumL_t^{tea} - \left(\sum_{d=1}^D \sum_{p \in TDaYF(d)}^{TDaYL(d)-1} A_{tp}^{tea} A_{t,p+1}^{tea} \right) \right) , \quad (8)$$

where $SumL_c^{cla}$ and $SumL_t^{tea}$ are the weekly total number of lessons assigned to class c and teacher t , respectively. The functions $TDaYF(d)$ and $TDaYL(d)$ give the first and last time periods of day d . Note that expression E_5 measures daily lesson contiguity as well as the number of days occupied with lessons.

As for the desire to satisfy the classes' and teachers' time preferences (constraint (s2)) two matrices, given as Pre_{cp}^{cla} and Pre_{tp}^{tea} , are used within term E_6 . Each element of these matrices is an integer (from 0 to 5) expressing the preferences of class c or teacher t to be involved in lessons during time period p , thus giving:

$$E_6 = \frac{1}{2} \left[\sum_{c=1}^C \sum_{p=1}^P Pre_{cp}^{cla} A_{cp}^{cla} + \sum_{t=1}^T \sum_{p=1}^P Pre_{tp}^{tea} A_{tp}^{tea} \right]. \quad (9)$$

The last energy term E_7 , which implements the minimization of teachers' and classes' shifts between teaching locations, is defined as follows:

$$E_7 = \frac{1}{2} \sum_{c=1}^C \sum_{\substack{l, l' \in LC_{Class}(c) \\ l \neq l'}} Q_{ll'}^{loc} \sum_{d=1}^D \left(\sum_{p \in TDay(d)} S_{lp} \right) \left(\sum_{p \in TDay(d)} S_{l'p} \right) + \frac{1}{2} \sum_{t=1}^T \sum_{\substack{l, l' \in LT_{Teacher}(t) \\ l \neq l'}} Q_{ll'}^{loc} \sum_{d=1}^D \left(\sum_{p \in TDay(d)} S_{lp} \right) \left(\sum_{p \in TDay(d)} S_{l'p} \right), \quad (10)$$

where $Q_{ll'}^{loc} = \begin{cases} 0 & \text{if lessons } l \text{ and } l' \text{ have the same location} \\ 1 & \text{otherwise.} \end{cases}$

Finally, the complete neural network energy function for the CTTTP is expressed by (11):

$$E(\bar{V}^x, \bar{V}^y) = \alpha (E_1 + E_2 + E_3 + E_4) + \beta (E_5 + E_6 + E_7). \quad (11)$$

Minimization of this CTTTP customized energy function leads to final neural network states corresponding to low conflict configurations and high quality CTTTP solutions. It should be noted that, in this CTTTP, one was only required to differentiate between two levels of constraints, hard and soft constraints (linked with α and β weights, respectively). However, if intermediate levels of constraints were required, they could be implemented by means of an appropriate weight balance.

4 Neural Network Simulation Approaches

To solve a combinatorial optimization problem such as the CTTTP, by using a type of HTNN, different improvements can be made use of. First, this section

describes a continuous Potts mean-field method (CPMF), which has produced good results when applied to complex optimization problems [21], including the CTTP [16],[8]. Then a discrete winner-take-all neuron approach (DWTAN) is proposed as an alternative to tackle the problem.

Note that, as will be seen later, in both models CTTP hard constraint (h1) is automatically satisfied, either by the Potts or the winner-take-all neuron method. As a result, the equations required to ensure (h1), forcing activation of only one neuron out of a row of P or R neurons, are always satisfied, as follows:

$$\sum_{p=1}^P V_{lp}^x = 1 \quad , \quad \sum_{r=1}^R V_{lr}^y = 1 \quad , \quad (12)$$

where, again, V_{lp}^x is the state of the lesson-period neuron (l, p) and V_{lr}^y represents the state of the lesson-room neuron (l, r) .

4.1 Continuous Potts Mean-Field Annealing Approach

As suggested by Peterson et al. [16], to improve the basic HTNN for the CTTP, the two-state Hopfield neurons were replaced by Potts multi-state neurons.

To find minimal states of the neural system according to a mean-field annealing method controlled by a temperature parameter T' [25], simulation of the dynamics of the neural network is performed by iterating the following equations:

$$V_{lp}^x = \frac{e^{U_{lp}^x}}{\sum_{p=1}^P e^{U_{lp}^x}} \quad , \quad U_{lp}^x = -\frac{1}{T'} \frac{\partial E}{\partial V_{lp}^x} \quad , \quad (13)$$

$$V_{lr}^y = \frac{e^{U_{lr}^y}}{\sum_{r=1}^R e^{U_{lr}^y}} \quad , \quad U_{lr}^y = -\frac{1}{T'} \frac{\partial E}{\partial V_{lr}^y} \quad . \quad (14)$$

Here, each of the neural sub-networks, with neural states given by V_{lr}^y and V_{lp}^x respectively, is iterated according to the CPMF generic algorithmic procedure outlined in figure 1.

The starting temperature T'_0 was determined by trial and error, $T'_0 = 1$. In addition, a maximum number of iterations must be given to act as a stopping criterion, here $MaxIter = 100$.

START $T'_0 = 1; MaxIter = 100; Iter = 0$

Initialize all neuron states

WHILE ($Iter < MaxIter$) **DO****IF** ($NSum^x = \frac{1}{N^x} \sum_{i=1}^{N^x} V_i^2 < 0.9$) **THEN**Randomly perform one update per neuron of lesson-period sub-network by using equations (13) with T'_{iter} **IF** ($NSum^y = \frac{1}{N^y} \sum_{i=1}^{N^y} V_i^2 < 0.9$) **THEN**Randomly perform one update per neuron of lesson-room sub-network by using equations (14) with T'_{iter} $Iter = Iter + 1$ $T'_{iter+1} = T'_{iter} \times 0.95$

Discretize all neuron states

END

Fig. 1. CPMF algorithm

Both neural sub-networks are initiated with a small random perturbation around an equilibrium state value. The equilibrium values are calculated by inverting the number of active neurons (not previously frozen) in each row of respective state matrices, which elements are represented by V_{lp}^x or V_{lr}^y .

Next, the main iterative step of the algorithm is executed while the maximum iteration parameter is not exceeded. This loop consists of updating, for a specific temperature level, all neuron states V_{lp}^x and V_{lr}^y using equations (13) and (14), according to a random, asynchronous process. On the assumption that N^x and N^y represent the total number of neurons within the lesson-room and lesson-period neural sub-networks, updating is performed as long as the respective global neural saturation, defined respectively as $NSum^x$ and $NSum^y$, is below 0.9. This situation occurs while the neurons assume overall non-discrete values, thus requiring more iterations to define final discrete output values (1 or 0). At the end of this loop, the temperature level and the iteration counter are changed.

The final step forces the discretization, according to equations (12), of all neurons not in 0 or 1 state, thus allowing direct extraction of the binary solution for the CTTP.

4.2 Discrete Winner-Takes-All Neuron Approach

An analogy exists between discrete and continuous versions of HTNN. In fact, in continuous versions, as the temperature parameter is lowered the neurons change from initial continuous real values in $[0, 1]$ to discrete 0, 1 values until, for extremely low temperatures, the neural states become almost discrete. The main disadvantage of using discrete HTNN versions for optimization purposes

concerns the difficulty of avoiding low quality local solutions, which often appear as a result of premature convergence. On the other hand, discrete versions are much more simple and efficient to implement computationally than continuous ones.

Bearing in mind the above considerations the proposed neural heuristic, which is an alternative to the continuous one previously described in (4.1), uses discrete competition winner-take-all neurons, whose dynamics for both sub-networks are defined by the following equations:

$$V_{lp}^x = \begin{cases} 1 & \text{if } U_{lp}^x = \max_{i \in P} \{U_{li}^x = -\frac{\partial E}{\partial V_{li}^x}\} \\ 0 & \text{otherwise,} \end{cases} \quad (15)$$

$$V_{lr}^y = \begin{cases} 1 & \text{if } U_{lr}^y = \max_{i \in R} \{U_{li}^y = -\frac{\partial E}{\partial V_{li}^y}\} \\ 0 & \text{otherwise.} \end{cases} \quad (16)$$

For a lesson l , the definition of these winner-take-all neurons forces the activation of only one of the respective row of neurons, which represent the R or P assignment alternatives associated with that lesson l . For each updating procedure using equations (15) or (16), the triggered winner neuron (which is turned on with a state equal to 1) corresponds to the alternative with the minimal additional cost for the energy function. All the remaining neurons of row l are updated to a state equal to 0. In terms of the basic HTNN model, this process can be compared to a context in which a row of neurons connected by extreme mutual lateral inhibition weights, activates the neuron with maximal excitation, while the remainder is, as a result, forced to be turned off.

This discrete approach also includes an improvement mechanism to escape from premature, low quality local optima. This is implemented through a condition-activated sub-procedure. The complete algorithmic procedure is outlined in figure 2.

The algorithm starts by defining the maximum number of iterations, $MaxIter = 1000$, and the maximum number of iterations without energy function improvement, $MaxEconst = 20$. The amount of random rows associated with lessons to be arbitrarily changed in order to avoid a potentially low quality local optimum is defined as $RandRows = 10$.

After randomly constructing an initial discrete solution, the algorithm enters the main loop. This consists in random, row by row updating of all neuron states, according to equations (15) or (16). Note that two flags, $StableX$ and $StableY$, are used to detect stable configurations of the sub-networks. Next, the energy function value is calculated and an eventual new best solution is

```

START
   $MaxIter = 1000; Iter = 0; MaxEconst = 20; EConst = 0$ 
   $RandRows = 10; EMin = +\infty$ 
  Generate initial discrete states for both neural sub-networks
  WHILE ( $Iter < MaxIter$ ) DO
     $StableX = True; StableY = True$ 
    FOR  $i = 1$  TO  $L$  DO
      WHILE (there exists a row in the lesson-period sub-network to be updated) DO
        Randomly choose a non-updated row  $k$ 
        Update neuron states of row  $k$ ,  $V_{kp}^x$ , using (15)
        IF (active neuron in row  $k$  is changed) THEN  $StableX = False$ 
      WHILE (there exists a row in the lesson-room sub-network to be updated) DO
        Randomly choose a non-updated row  $k$ 
        Update neuron states of row  $k$ ,  $V_{kr}^y$  using (16)
        IF (active neuron in row  $k$  is changed) THEN  $StableY = False$ 
      Calculate energy  $E(\bar{V}^x, \bar{V}^y)$ 
      IF  $E(\bar{V}^x, \bar{V}^y) < EMin$  THEN DO
        Save best solution;  $EMin = E(\bar{V}^x, \bar{V}^y); EConst = 0$ 
      ELSE
         $EConst = EConst + 1$ 
      IF ( $StableX$  AND  $StableY$ ) OR ( $EConst > MaxEconst$ ) THEN DO
        Reset neurons with non-zero energy costs
        Randomly choose  $RandRows$  rows and active one neuron per row
         $EConst = 0$ 
       $Iter = Iter + 1$ 
    END

```

Fig. 2. DWTAN algorithm

saved, otherwise the $Econst$ counter is increased. Then, if both neural networks are stable or the maximum number of iterations without energy function improvements is exceeded, two actions occur. Firstly, the neurons with an associated cost on the energy function are turned off and, secondly, a number of $RandRows$ rows are randomly selected and respective neuron states modified so as to allow the algorithm to search for a new network configuration. The final solution is the one with the lower energy function value, thus corresponding to a higher quality CTTP solution.

5 Computational Results

To compare the computational behavior of the aforesaid neural network heuristics, two sets of class/teacher timetabling instances were used. The first set consists in five hard hypothetical CTTP instances, with distinct dimensions.

The second is a collection of three real CTTTP instances taken from a university institution in Portugal.

5.1 CTTTP Instances Description

The hypothetical CTTTP instances were extracted from the OR-Library [5] (available at <http://mscmga.ms.ic.ac.uk/info.html>). The main characteristics and number of neurons required to map the instances are presented in Table 1.

Table 1

Hard hypothetical CTTTP instances - characteristics and number of neurons.

Problem instance	Lessons	Periods	Rooms	Teachers	Classes	Total number of neurons
HDTTP4	120	30	4	4	4	3600
HDTTP5	150	30	5	5	5	4500
HDTTP6	180	30	6	6	6	5400
HDTTP7	210	30	7	7	7	6300
HDTTP8	240	30	8	8	8	7200

These hypothetical instances presume a single room previously assigned for each lesson, hence the neural simulation is restricted to the lesson-period sub-network only.

For the purpose of evaluating the difficulty level associated with each situation, some generic timetabling indexes are defined and calculated in Table 2 for the hard hypothetical CTTTP instances. These instances are taken from basic constraint satisfaction problems (with active hard constraints (h1),(h2) and (h5)), whose solution corresponds to a full lesson, conflict-free timetable without soft requirements.

The time-slot occupation index measures the level of time-slot occupancy by dividing the number of lessons to be scheduled by the number of time-slots (rooms \times time-periods). Another issue capable of affecting the difficulty of a CTTTP instance is the number of alternative rooms that can be assigned to each lesson. To evaluate this aspect, the lesson-room rigidity index is calculated by dividing the number of lessons by the total number of room alternatives. As described in Table 2, indexes for all instances reveal both difficult, fully constrained CTTTP instances.

Three examples of real instances are presented in Table 3, derived from first semester data of the Escola Superior de Gest3o, Hotelaria e Turismo of Algarve

Table 2
Hard hypothetical CTTP instances - difficulty issues.

Problem instance	Time-slot occupation	Lesson-room rigidity index	Active constraints
HDTTP4	1	1	<i>h1, h2, h5</i>
HDTTP5	1	1	<i>h1, h2, h5</i>
HDTTP6	1	1	<i>h1, h2, h5</i>
HDTTP7	1	1	<i>h1, h2, h5</i>
HDTTP8	1	1	<i>h1, h2, h5</i>

University in Portugal, on the institution’s two campuses (Faro and Portimão).

Table 3
Real CTTP instances - characteristics and number of neurons.

Problem instance	Lessons	Periods	Rooms	Teachers	Classes	Number of neurons		
						Room	Period	Total
PREAL	37/48*	12	4	6	6	148	444	592
ESGHT1	174/209*	50	9	51	32	1566	8700	10266
ESGHT2	568/626*	50	27	107	92	15363	28450	43813

* Number of equivalent single period lessons.

PREAL is a small dimension instance construct based on a fraction of the ESGHT2 courses. However the number of available rooms was reduced in order to increase difficulty level. For this case, hard constraints (h1),(h2),(h3),(h5) and soft constraint (s1) were applied, as shown in Table 4. ESGHT1 represents the restricted CTTP occurring on Portimão campus. Its structure is characterized by all hard and soft constraints except (s3). Finally, ESGHT2 represents a complete CTTP instance in which all constraints mentioned in Section 2 are applied. The last columns of Table 3 show the number of lesson-room and lesson-period neurons required to map instances, which all include lessons of multiple lengths.

Table 4 indicates the difficulty indexes registered, revealing that all instances have a significant level of time-slot occupation, especially PREAL, which is totally saturated. Furthermore, levels of lesson-room rigidity tend to increase instance intricacy by imposing significant compulsory restrictions through hard constraint (h5).

Table 4
Real CTTTP instances - difficulty issues.

Problem instance	Time-slot occupation index	Lesson-room rigidity index	Active constraints
PREAL	1	0.27	$h1, h2, h3, h5, s1$
ESGHT1	0.46	0.20	$h1 \dots h8, s1, s2$
ESGHT2	0.46	0.14	$h1 \dots h8, s1 \dots s3$

5.2 Comparison of Results

For the purpose of computational implementation, the two neural heuristics described above were programmed on Delphi 5.0, and executed on a Pentium III 800Mhz with 128 MRam.

The tests were conducted by performing 20 complete runs for each instance and neural heuristic. Table 5 shows the results obtained for the hypothetical CTTTP instances. The energy function weight parameters for all experiments were defined as $\alpha=100$ and $\beta=1$.

Table 5
Hard hypothetical CTTTP instances - computation results after 20 runs

Problem instance	CPMF approach			DWTAN approach		
	Energy value		Run time	Energy value		Run time
	min.	average	average (sec.)	min.	average	average (sec.)
HDTTP4	5	10.7	50	0	0	18
HDTTP5	8	13.2	154	0	0.4	102
HDTTP6	11	18.7	308	0	1.65	213
HDTTP7	18	25.6	426	0	2.1	305
HDTTP8	15	28.6	1454	0	3.25	1237

The DWTAN approach clearly achieves a higher performance when compared to the CPMF approach, both in terms of solution quality and execution time. In fact, for all instances the DWTAN heuristic attained the optimal solution, in particular for instance HDTTP4, where all runs obtained a zero conflict solution. Note that, due to full saturation of these instances, the DWTAN heuristic parameters *MaxIter* and *RandRows* were changed to 10000 and 2, respectively.

Table 6 shows the comparative behavior of the heuristics applied to the real CTTTP instances restricted to the set of hard constraints.

Table 6

Real CTTTP instances restricted to hard constraints - computation results after 20 runs

Problem instance	CPMF approach			DWTAN approach		
	Energy value		Run time	Energy value		Run time
	min.	average	average (sec.)	min.	average	average (sec.)
PREAL	1	1.7	136	0	0.2	5
ESGHT1	0	0.25	1251	0	0	13
ESGHT2	0	0.65	6583	0	0	116

The above findings illustrate the capability of the neural methods to produce feasible CTTTP solutions for the proposed real instances. Once again, the figures testify to the DWTAN method’s superior performance. It should be noted that for the larger cases (ESGHT1 and ESGHT2) this approach always found a feasible solution in less than 2 minutes.

Finally, the set of real instances were tackled bearing in mind all their constraints (hard and soft) by both heuristics. The results are summarized in Table 7.

Table 7

Real CTTTP instances - computation results after 20 runs

Problem instance	CPMF approach			DWTAN approach			Manual
	Energy value		Run time	Energy value		Run time	approach
	min.	average	time (sec.)	min.	average	time (sec.)	Energy value
PREAL	131	162	251	27	50.4	27	–
ESGHT1	302	358.2	3310	272	289	143	328
ESGHT2	770	853.5	16214	641	667.2	517	864

The comparative results, once again, indicate that the DWTAN is more effective in producing high quality CTTTP solutions than the CPMF approach. For instance, for ESGHT2, the discrete approach was, on average, able to find high quality solutions in less than 10 minutes, while the continuous heuristic took over four hours to find an inferior quality solution. For purposes of comparison, the last column of Table 7 indicates the energy value corresponding to the institution’s actual manual solution produced over a week by a three-man team. As observed, the quality of the solutions found by both neural heuristics is globally superior to the quality obtained by manual procedures. This is particularly evident in the case of the DWTAN solution, which significantly reduced soft constraint cost.

6 Conclusions

This article has presented an experimental comparison of two neural network-based heuristics to tackling the class/teacher timetabling problem. First, the CTTP characteristics were described and a specific energy function was advocated for neural optimization purposes. Then the CPMF heuristic and the DWTAN approach developed were described and compared using sets of hard hypothetical and real CTTP instances. Attention is drawn to the fact that although far more tests were performed, within reality, only three typical instances are reported here.

Clearly, in terms of solution quality the results given indicate that the DWTAN is preferable as the most effective neural heuristic, both for hard hypothetical and real CTTP instances. In addition the authors find that DWTAN heuristic is significantly faster than CPMF heuristic, particularly in the larger CTTP instances. On extending the comparison to include the manual approach, DWTAN heuristic's output was found to be very satisfactory. It produced highly improved CTTP solutions within a fraction of the time taken by the manual approach.

Performance of the neural methods was also experimentally compared, for real instances. The two contexts involved hard constraints only, and hard and soft constraints simultaneously. Considering only hard constraints, both neural heuristics performed satisfactorily. Particular attention is drawn to the DWTAN, which always found feasible solutions for two instances, within a few seconds. This fact suggests that the approach constitutes an effective generation method for constructing non-conflicting timetables.

In future, further research will be developed on the application of the proposed winner-take-all neural heuristic to other large CTTP instances, which are already being tackled. Moreover, the natural parallelization capability of neural systems may be explored to improve algorithm performance.

References

- [1] D. Abramson. Constructing School Timetables Using Simulated Annealing: Sequential and Parallel Algorithms. *Management Science*, 37:83–113, 1982.
- [2] T. Birbas, S. Daskalaki and E. Housos. Timetabling for Greek High Schools. *Journal of the Operational Research Society*, 48:1191–1200, 1997.
- [3] E. Burke, D. Elliman and R. Weare. Examination Timetabling in British Universities - A Survey. *The Practice and Theory of Automated Timetabling*, E.

- Burke and P. Ross (Eds.), LNCS Springer Verlag, Edinburgh, United Kingdom, 1153:76–90, 1996.
- [4] E. Burke and M. Carter (Eds.). *The Practice and Theory of Automated Timetabling II*. Volume 1408. LNCS Springer Verlag, Toronto, Canada, 1998.
- [5] J. Beasley. OR-Library: Distributing Test Problems by Electronic Mail. *Journal of the Operational Research Society*, 41(11):1069–1072, 1990.
- [6] E. Burke and P. Ross (Eds.). *The Practice and Theory of Automated Timetabling*. Volume 1153. LNCS Springer Verlag, Edinburgh, United Kingdom, 1996.
- [7] M. P. Carrasco. Redes Neurais na Elaboração de Horários Escolares. Master thesis, Instituto Superior de Economia e Gestão - Universidade Técnica de Lisboa, 1995.
- [8] M. P. Carrasco and M. V. Pato. A Potts Neural Network Heuristic for the Class/Teacher Timetabling Problem. *Proceedings of the 4th Metaheuristics International Conference*, 139–142, 2001.
- [9] M. P. Carrasco and M. V. Pato. A Multiobjective Genetic Algorithm for the Class/Teacher Timetabling Problem. *The Practice and Theory of Automated Timetabling III*, E. Burke and E. Erben (Eds.), LNCS Springer Verlag, Konstanz, Germany, 2079:3–17, 2001.
- [10] M. Carter and G. Laporte. Recent Developments in Practical Examination Timetabling. *The Practice and Theory of Automated Timetabling*, E. Burke and P. Ross (Eds.), LNCS Springer Verlag, Edinburgh, United Kingdom, 1153:3–21, 1996.
- [11] A. Colomi, M. Dorigo and V. Maniezzo. Genetic Algorithms and Highly Constrained Problems: The Timetabling Case. *Proceedings of the First International Conference on Parallel Problem Solving from Nature*, 55–59, 1982.
- [12] D. Costa. A Tabu Search Algorithm for Computing an Operational Timetable. *European Journal of Operational Research*, 76:98–110, 1994.
- [13] K. Dowsland. A Timetabling Problem in which Clashes are Inevitable. *Journal of the Operational Research Society*, 41(10):907–908, 1990.
- [14] S. Even, A. Itai and A. Shamir. On the Complexity of Timetabling and Multicommodity Flow Problems. *SIAM Journal of Computation*, 5(4):691–703, 1976.
- [15] L. Gislén, C. Peterson, and B. Söderberg. Teachers and Classes with Neural Networks. *International Journal of Neural Systems*, 1:167–176, 1989.
- [16] L. Gislén, C. Peterson and B. Söderberg. Complex Scheduling with Potts Neural Networks. *Neural Computation*, 4:805–831, 1992.
- [17] J. Hopfield and D. Tank. Neural Computation of Decisions in Optimization Problems. *Biological Cybernetics*, 52:141–152, 1985.

- [18] C. Looi. Neural Network Methods in Combinatorial Optimization. *Computer Operations Research*, 19(3/4):191–208, 1992.
- [19] L. Kiar and J. Yellen. Weighted Graphs and University Course Timetabling. *Computer Operations Research*, 19(1):59–67, 1992.
- [20] L. Paquete and C. Fonseca. A Study of Examination Timetabling with Multiobjective Evolutionary Algorithms. *Proceedings of the 4th Metaheuristics International Conference*, 149–153, 2001.
- [21] C. Peterson and B. Söderberg. A New Method for Mapping Optimization Problems onto Neural Networks. *International Journal of Neural Systems*, 1(3):3–22, 1989.
- [22] A. Schaerf. A Survey of Automated Timetabling. *Artificial Intelligence Review*, 13:87–127, 1999.
- [23] K. Smith. Neural Networks for Combinatorial Optimization: A Review of More Than a Decade of Research. *INFORMS Journal on Computing*, 11:15–34, 1999.
- [24] A. Tripathy. School Timetabling - a Case in Large Binary Integer Linear Programming. *Management Science*, 30(12):1473–1498, 1984.
- [25] D. Van Den Bout and T. Miller. Improving the Performance of the Hopfield-Tank Neural Network Through Normalization and Annealing. *Biological Cybernetics*, 62:123–139, 1989.
- [26] M. Yoshikawa, E. Kaneko, Y. Nomura and M. Watanabe. A Constraint-Based Approach to High-School Timetabling Problems: A Case Study. *Proceedings of the 12th Conference on Artificial Intelligence*, 94:1111–1116, 1994.
- [27] D. Werra. An Introduction to Timetabling. *European Journal of Operational Research*, 19:151–162, 1985.
- [28] D. Werra. Some Combinatorial Models for Course Scheduling. *The Practice and Theory of Automated Timetabling*, E. Burke and P. Ross (Eds.), LNCS Springer Verlag, Edinburgh, United Kingdom, 1153:296–308, 1996.
- [29] D. Werra. The Combinatorics of Timetabling. *European Journal of Operational Research*, 96:504–513, 1997.