Preliminary Particles Model of Unconstrained Examination Timetabling and Its Optimization Using Neural Networks

Suren Khachatryan
American University of Armenia
Yerevan, Armenia
e-mail: skhachat@aua.am

Nareh Salmasian
American University of Armenia
Yerevan, Armenia
nareh_salmasian@edu.aua.am

ABSTRACT
Construction of feasible examination timetables implies strong interaction between the involved exams. We propose a dynamical model of unconstrained examination timetabling, where exams are represented by interacting particles cyclically moving over the timeslots. A simple interaction rule is adopted – move to the next slot, if in a clash, and the corresponding evolutionary operators are derived. The model is applied to the Toronto benchmark problems, and evolution of the resulting systems is studied in terms of the number of clashes. Based on the observed main properties, a Propagating Particles Algorithm of an Academic Time Tabling (bi-PAT) is formulated without explicitly drawing on search or optimization paradigms. The algorithm is compared with 3 other methods originating from analogies with natural science phenomena, as well as 5 combinatorial algorithms. Competitive performance of bi-PAT with the best algorithms is observed in case of problems of moderate size. The reasons of significant deviation from the best results in case of large problems are discussed. In the last section a neural network-based optimization strategy is proposed.

Keywords
Scheduling, dynamical systems, neural networks.

1. INTRODUCTION
Timetabling problems generally and educational timetabling specifically have been subjects of active studies for many decades due to their high practical value. Different formulations of educational timetabling problems exist. In the survey of automated timetabling [1] they are classified into three main types – school, course and examination timetabling. It is added, however, that such division is not strict, and hybrid cases can appear in practice. Differences between the mentioned types are discussed in [2]. In the present paper we consider an unconstrained examination timetabling problem with the only objective to avoid students taking different exams at the same time. Usually, additional hard and soft constraints are imposed to such time tables. Several important examples are enlisted in [3].

The algorithms and research directions are summarized in an in-depth survey of examination timetabling methodologies [3]. Ordering heuristic approaches originate from the fundamental work [4], in which timetabling is treated from a graph coloring point of view. Five sorting strategies are discussed and compared in [5]: by saturation degree – the number of available timeslots [6]; by largest degree – the number of conflicts; by weighted largest degree – the number of students in conflict; by largest enrolment; and, finally, by random ordering. It is shown that, generally, sorting by saturation degree combined with backtracking strategy leads to more compact schedules and requires less computing time.

Combinatorial algorithms developed by analogies with natural science phenomena prove to be reasonably productive. Simulated annealing is a local search method widely used in scheduling and timetabling. It originates from an equation of state calculation algorithm for systems composed of interacting molecules with spherically symmetric potential field [7]. A hybrid method combining simulated annealing and constraint programming is proposed in [8]. Being tested on the same data the approach results in somewhat longer unconstrained timetables than those constructed in [5].

Genetic algorithms mimic natural evolutionary processes and the principles of genetics. Possible solutions are represented as chromosomes, and new chromosomes are generated by crossover and mutation mechanisms. Applications of genetic algorithms to educational timetabling problems are summarized in a brief survey [9]. An algorithm with a linear linkage encoding representation scheme is suggested in [10] for solving the graph coloring and exam timetabling problems. Another class of methods inspired by a biological concept consists of ant colony algorithms that behave similar to ants in their search for the shortest path to food. Such an approach is developed in [11] and used as an evidence that ant colony optimization should not be limited to routing problems, though being firmly focused to the latter. It is shown that proper configuration puts the algorithm among the best methods applied to the same unconstrained problems as in [5].

Timetabling implies strong interaction between the exams through clashes. The present work aims at modelling of this interaction and construction of the corresponding dynamical system. The main goal is to show that adequate dynamical description can lead to feasible solutions without explicitly drawing on search or optimization heuristics.

2. DYNAMICAL MODEL
Let us consider construction of a feasible examination time table – given N class lists, schedule the corresponding N exams within S timeslots in a clash-free way. Two exams clash, if they are scheduled for the same slot and there is at least one student included in both class lists. Let us then consider the exams as a 1D dynamical system with the enumerated timeslots being the discrete periodic coordinate of period S, and adopt a simple interaction rule – if in a clashing state, an exam shifts to the next slot. Formally, we introduce a force operator $\mathbf{F}_k$ that acts on the k-th exam and has the following form:

$$\mathbf{F}_k(t_k) = (t_k + b) \bmod S \qquad (1)$$

where t_k is the current timeslot of the k-th exam ($0 \le t_k < S$ and $0 \le k < N$); and binary constant b is equal to 1, if the exam is in a clashing state at t_k, and 0 – otherwise. Generally, operators $\mathbf{F}_k$ do not commute and, therefore, exam numbering matters. To allow moves of arbitrary depth, we also consider integer powers of $\mathbf{F}_k$:

$$\mathbf{F}_k^m(t_k) \equiv \mathbf{F}_k \otimes \mathbf{F}_k \otimes \ldots \otimes \mathbf{F}_k \ m \text{ times} = (t_k + c) \bmod S, \quad (2)$$

where c is the number of successive timeslots starting from t_k, for which exams clashing with the k-th one are scheduled ($0 \le c \le m$). Successive action of these operators will determine the evolution of the entire system we will be tracking down in terms of changes in the number of clashes. Therefore, we introduce parameterized cumulative evolutionary operators **E** of the following form:

$$\mathbf{E}(m,d) = \left(\prod_{k=0}^{N-1} \mathbf{F}_k^m(t_k) \right)^d , \quad (3)$$

where m and $d >> 1$ represent a mode and duration, respectively. In expanded terms, **E**(m,d) shifts in succession each of N exams up to m modulo S slots forward subject to clashes and repeats this cycle d times.

Figure 1 illustrates the action of **E**(m,d) on real-life timetable examples taken from the Toronto dataset [3]. The overall number of clashes left in the system after each of d iterations is plotted against the mode duration d. In all cases exams are initially scheduled for the first timeslot.

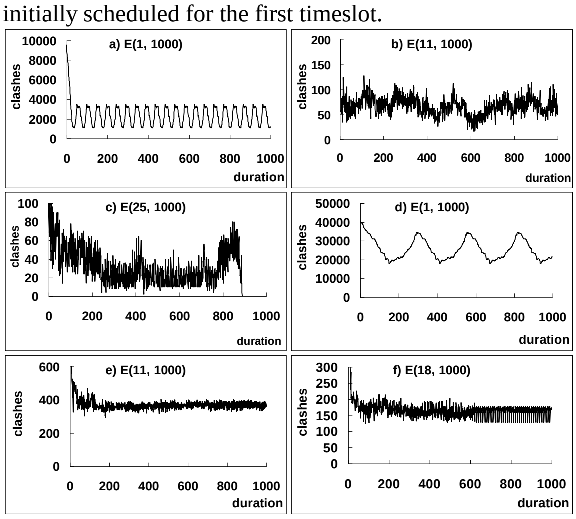

Figure 1. Examples of **E**(m,d) cumulative operators

Figures 1a, 1b and 1c correspond to "ear-f-83.stu" data file – a relatively small timetable with 190 exams, 24 slots and 9586 clashes in the initial state, while Fig. 1d, 1e and 1f – "car-f-92.stu" with 543 exams, 32 slots and 40610 clashes. They show that the systems pass through an initial transient stage followed by a steady-state periodic phase. Three mode types can be distinguished for the given duration d: periodic modes dominated by steady-state phase $d_{transient} << d_{steady\text{-}state}$ (Fig. 1a and 1d); quasi-periodic modes effectively limited to the transient phase $d_{transient} \approx d$ (Fig. 1b, 1c and 1e); and mixed modes with comparable stages $d_{transient} \approx d_{steady\text{-}state}$ (600 vs. 400 iterations in Fig. 1f). Any superposition of **E**(m,d) operators that constructs a clash-free examination timetable will identify a timetabling model. Particularly, the model can be reduced to a single operator.

3. BI-PAT ALGORITHM

The current work is based on the Toronto benchmark dataset summarized in Table 1. The problems are sorted by the maximal number of clashes – the main measure of complexity. To reveal the basic properties of **E**(m,d), we schedule all the exams of the given problem for the first timeslot – the initial state, and separately apply **E**(m,1000) for each of $1 \le m < S$. Figure 2 depicts the minimal number of clashes as a function of m for 8 relatively small problems from Table 1. Data points corresponding to periodic and mixed modes are in white, to quasi-periodic ones – in black. There can be drawn certain observations connecting the values of m with the mode type. If m and S are co-prime, then the corresponding modes are quasi-periodic. The only exception is m = 2, where the steady-state periodic phase is always present, at least for duration d = 1000, even if S is odd. All other modes are in resonance with S and, therefore, are either periodic or mixed. Particularly, if m is a factor of S, then the corresponding modes are strictly periodic. Secondly, there may be quasi-periodic modes that construct feasible timetables of the given problem. We call them feasible modes. Obviously, duration of the feasible modes is shorter for simpler problems.

Table 1. Toronto Benchmark Dataset

Problem	Exams	Students	Clash density	Max clashes	Timeslots
hec-s-92	81	2823	20%	2726	18
sta-f-83	139	611	14%	2762	13
ute-s-92	184	2749	8%	2860	10
lse-f-91	381	2726	6%	9062	18
yor-f-83	181	941	27%	9412	21
ear-f-83	190	1125	29%	9586	24
kfu-s-93	461	5349	6%	11786	20
tre-s-92	261	4360	18%	12262	23
rye-s-93	486	11483	7%	17744	23
car-f-92	543	18419	14%	40610	32
uta-s-93	622	21266	13%	48498	35
car-s-91	682	16925	13%	59628	35
pur-s-93	2419	30029	3%	164598	42

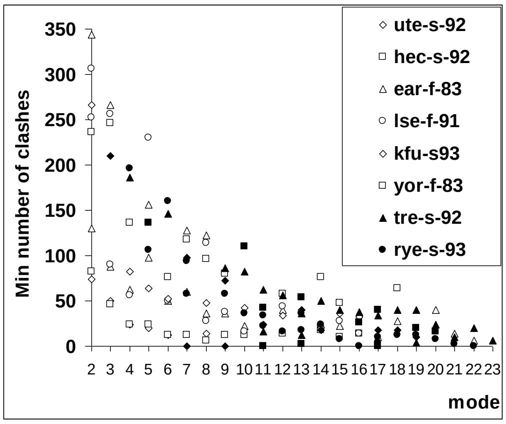

Figure 2. Dependence of the minimal number of clashes on modes (d = 1000) in small Toronto benchmark problems.

Assuming there is a solution for the given number of timeslots S, the obtained results allow formulating a modal version of a Propagating Particle Algorithm of an Academic Time Tabling – bi-PAT, as follows:

1. Construct the initial schedule by assigning all exams to the first timeslot.
2. Select a sufficiently large value of d and apply different **E**(m,d) operators to the initial schedule, where m and S are mutually prime numbers and $2 < m < 2S$.
3. If none of the applied operators constructs a feasible timetable, double d and repeat the step 2.

Implementation of bi-PAT in this way may require significantly long duration, since no upper bound for the number of iterations d is explicitly specified. For example, d = 40010 for lse-f-91 problem; and d = 1000000 is not sufficient for construction of a feasible car-f-92 timetable – the first large problem in Table 1. It is, therefore, important to consider superposition of different modes. Let us introduce two modal characteristics – the mean number of clashes C_0 averaged over the entire duration and the amplitude of periodic oscillations or quasi-periodic fluctuations A. Numerical experiments revealed the following main properties:

- Expectedly, C_0 practically does not depend on d and is a decreasing function of m, $1 \le m < S$.
- Generally, periodic modes oscillate around lower mean than quasi-periodic ones: $C_{0periodic} < C_{0quasi\text{-}periodic}$. Amplitudes of periodic oscillations, however, are smaller than the mean by at least an order of magnitude: $A_{periodic} << C_{0periodic}$. Therefore, in periodic modes the minimal number of clashes $C_{0periodic} - A_{periodic}$ does not vanish, unless the mean is of order of unity.
- Unlike periodic modes, amplitudes of quasi-periodic fluctuations depend on duration, and for sufficiently large d can become comparable with the mean, thus considerably minimizing the number of clashes (Fig. 1b) or completely resolving them (Fig. 1c).

Aiming at bringing together all positive tendencies of the enlisted properties, we come up with an alternative formulation of bi-PAT – its cumulative version:

1. Select a value of d, large enough to enter steady-state phase of periodic modes;
2. Construct the initial schedule by assigning all exams to the first timeslot, and apply $\mathbf{E}(1,d)$;
3. In a loop for each of $m < S$ modes apply $\mathbf{E}(m,d)$ to the schedule that minimized clashes during the previous iteration $\mathbf{E}(m\text{-}1,d)$.

As an example, the solution of hec-s-92 problem is depicted in Fig. 3. In all $\mathbf{E}(m,d)$ operators the same number of iterations d = 1000 is used. For each of them the state with minimal clashes was chosen as the initial schedule for the next one. Vertical bounds indicate the resulting mode duration. Figure 3 shows that the superposition of the modes adds steadily decreasing behavior to system's evolution.

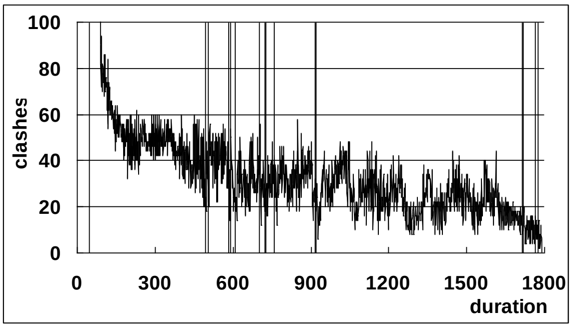

Figure 3. Cumulative bi-PAT applied to hec-s-92 timetable. The vertical lines indicate the bounds of successive modes.

Normally, the overall duration of the cumulative version is much longer that the shortest feasible mode. In some cases, however, reversed situation is observed.

4. TESTING RESULTS

One of the main objectives of the Toronto dataset is to minimize the number of timeslots needed for construction of unconstrained timetables [3]. We compare bi-PAT with the best results achieved by several other methods in Table 2. The five sequential algorithms with backtracking [5] are denoted as follows: LD – sorting by the largest degree, SD – by the saturation degree, WD – by the weighted largest degree, LE – by the largest enrolment, and RO – random ordering. The simulated annealing hybrid method [8] denoted as SA, the genetic algorithm [10] – GA, and the ant colony optimization method [11] – AC, are also included. The records are divided into two main parts – the first 9 small problems with less than 20000 clashes each, and the remaining 4 large problems.

Table 2. Minimal number of timeslots by different approaches

Problem	LD	SD	WD	LE	RO	SA	GA	AC	bi-PAT
hec-s-92	18	17	17	17	17	18	17	17	17
sta-f-83	13	13	13	13	13	13	13	13	13
ute-s-92	10	10	10	10	10	11	10	10	10
lse-f-91	17	17	17	17	17	18	17	17	17
yor-f-83	20	20	20	19	20	23	20	19	21
ear-f-83	22	22	22	22	22	24	23	22	23
kfu-s-93	19	19	19	20	19	21	20	19	20
tre-s-92	22	20	20	22	22	21	21	20	23
rye-s-93	21	21	22	22	22	22	23	21	22
car-f-92	31	28	30	31	32	31	36	28	35
uta-s-93	33	32	33	33	34	32	38	30	38
car-s-91	32	28	30	31	32	30	36	28	38
pur-s-93	36	35	38	38	N/A	N/A	N/A	N/A	48

All smaller timetables have been constructed for timeslots enlisted in Table 1 by both modal and cumulative versions of bi-PAT. However, there are only two cases among them that correspond to the best solutions – sta-f-83 and ute-s-92 with 13 and 10 timeslots, respectively. To construct timetables with less slots, we apply customized versions of bi-PAT, where different combinations of the $\mathbf{E}(m,d)$ operators are empirically selected and tried instead of the cumulative superposition. Here again, the state with minimal number of clashes of the previous mode becomes the initial one for the next evolutionary operator. Particularly, cyclic application of $\mathbf{E}(15,1000)$, $\mathbf{E}(16,1000)$, $\mathbf{E}(18,1000)$ and $\mathbf{E}(19,1000)$ constructs hec-s-92 timetable of 17 slots. Another customized bi-PAT improves rye-s-93 timetable: $\mathbf{E}(21,2000) \otimes \mathbf{E}(11,10)^3 \otimes \mathbf{E}(21,100)$. Consideration of longer duration may be an alternative strategy. For example, ear-f-83 timetable of 24 slots is constructed by $\mathbf{E}(25,889)$, while the one with 23 timeslots – by $\mathbf{E}(25,1227)$. Similarly, lse-f-91 timetable of 18 slots is constructed by cumulative bi-PAT with d = 1000, while the one with 17 timeslots requires 10 times more iterations.

The best results on the Toronto data are achieved by SA [5] and AC [11]. It follows from Table 2 that bi-PAT demonstrates reasonable performance on the relatively small problems. It matches the best results reported for the 4 smallest problems [5, 10-11]. As for the remaining 5 small problems, it is slightly worse than the sequential algorithms [5] and the ant colony method [11], but competitive with SA [8] and GA [10]. We find it important to emphasize that all the mentioned approaches are explicitly based on search or optimization paradigms, while the versions of bi-PAT reflect on evolution of the dynamical models (3).

Different behavior of the dynamical models is observed in case of the 4 large timetables with more than 40000 clashes. Only cumulative bi-PAT is applied to these problems, since the duration of feasible modes (if any) quickly grows with the problem complexity. The results obtained for car-f-92, uta-s-93 and car-s-91 timetables agree with those achieved by GA [10]. Both methods, however, are significantly outperformed by other approaches [5, 8, 11]. The largest timetable pur-s-93 is a special case. It requires 48 timeslots to construct a feasible timetable by cumulative bi-PAT with d = 1000, which is essentially inferior to the sequential algorithms excluding the one of random ordering [5].

5. PERFORMANCE OPTIMIZATION

The amount of iterations remains unspecified in both versions of bi-PAT. To enhance speed it up, different combinations of **E**(m,d) operators may be applied. Their successful empirical search, however, cannot be practically implemented in large timetables. Meantime, the modal version quickly converges to sub-optimal solutions, where the number of clashes is minimized. For example, no feasible solution is found within d = 1000 iterations on Fig. 1b, but in around d = 600 iterations a sub-optimal solution is found with only 12 clashes.

Based on sub-optimal solutions we propose below an optimization method using, as an example, the smallest timetable hec-s-92 – in the best case it should be constructed within 17 slots. We supplement the system with a Hopfield network – a one-layer neural network that has as many N binary neurons as many exams are included in the timetable [12]. Thus, it represents a single timeslot with i-th exam scheduled for it, if the corresponding i-th neuron has value 1.

We apply **E**(18,100). In less than d = 100 iterations a sub-optimal solution with only 2 clashes is found. Since the clashes are counted in pairs, all-but-one slots appear in feasible states. Each such slot represents a possible large clash-free clique of exams and, therefore, is used in the training set of the Hopfield network. The clashing slot generates two more training configurations with the first clashing exam removed from the first one and the second clashing exam – from the second one. If using an arbitrary, not sub-optimal, timetable with many clashes per slot, the inclusion of clashing slots into the training set becomes inefficient and the cliques drastically shrink.

Once the network is trained, it is used for generation of a more optimal initial state. As before, all exams are initially scheduled for the first slot – this is the first input to the trained network. All exams from the output are left in this slot and others are moved to the second one – this is the second input to the network. The process continues until the last slot. It is not guaranteed that the generated initial state is feasible, but the subsequent application of pi-PAT will require shorter duration.

6. CONCLUSION

In the present work we treat examination timetabling as a system of interacting particles and suggest an iterative and spatially periodic multimodal model of the interaction between the exams through the parameterized evolutionary operators **E**(m,d) (1)–(3). Originating from an intuitive concept of moving away from clashing counterparts, the model exhibits realistic behavior with certain similarities to dynamical systems governed by laws of classical mechanics. Particularly, we observe such well-known and studied phenomenon as resonant orbits or periodic modes, where the period of motion of individual particles m has common divisors with the problem's spatial period S – the number of available timeslots. Meantime, experiments with the Toronto benchmark problems demonstrate that quasi-periodic modes with co-prime m and S are closely related to the construction of feasible timetables. Each of 9 relatively small problems has at least one quasi-periodic mode of evolution that resolves all the clashes. Based on the revealed general properties, the Propagating Particles Algorithm of an Academic Time Tabling – bi-PAT, is formulated in two versions. In terms of constructing timetables of minimal length, it can compete with other approaches, when applied to problems of moderate size.

The main weakness of the algorithm is the unbounded specification of the duration parameter d. To optimize the search, we consider states in timeslots as patterns and propose using a Hopfield network for pattern matching. A strategy of the network training is presented based on the hec-s-92 case-study from the Toronto benchmark dataset. For more detailed formulation, we are currently developing an integrated environment that will combine the implementation of bi-PAT algorithm with the Hopfield network. Having this tool, we will present in the next papers the testing results and draw comparisons with other methods.

REFERENCES

[1] A. Schaerf, "A survey of automated timetabling", *Artificial Intelligence Review*, 87-127, 13(2), 1999.

[2] B. G. C. McCollum, "A perspective on bridging the gap between theory and practice in university timetabling", in *Practice and Theory of Automated Timetabling: Selected Papers from the 6th Int. Conf., Springer Lecture Notes in Computer Science*, 3-23, 3867, 2007.

[3] R. Qu, E. K. Burke, B. McCollum, L. T. G. Merlot, S. Y. Lee, "A survey of search methodologies and automated system development for examination timetabling", *Journal of Scheduling*, 55-89, 12 (1), 2009.

[4] D. J. A. Welsh, M.B. Powell, "The upper bound for the chromatic number of a graph and its application to timetabling problems", *The Computer Journal*, 41-47, 11, 1967.

[5] M.W. Carter, G. Laporte, S.Y. Lee, "Examination timetabling: algorithmic strategies and applications", *Journal of Operational Research Society*, 373-383, 47 (3), 1996.

[6] D. Brelaz, "New methods to colour the vertices of a graph", *Communications of the Association for Computing Machinery*, 251-256, 22, 1979.

[7] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, "Equation of state calculations by fast computing machines", *The Journal of Chemical Physics*, 1087-1092, 21 (6), 1953.

[8] L. T. G. Merlot, N. Boland, B. D. Hughes, P. J. Stuckey, "A hybrid algorithm for the examination timetabling problem", in *Practice and Theory of Automated Timetabling: Selected Papers from the 4th Int. Conf., Springer Lecture Notes in Computer Science*, 207-231, 2740, 2003.

[9] D. Corne, P. Ross, H. Fang, "Evolutionary timetabling: practice, prospects and work in progress", in *Proceedings of UK Planning and Scheduling SIG Workshop*, 726-731, 1994.

[10] O. Ulker, E. Ozcan, E. E. Korkmaz, "Linear linkage encoding in grouping problems: applications on graph colouring and timetabling", in *Practice and Theory of Automated Timetabling: Selected Papers from the 6th Int. Conf., Springer Lecture Notes in Computer Science*, 347-363, 3867, 2007.

[11] K. A. Dowsland, J. M. Thompson, "Ant colony optimization for the examination scheduling problem", *The Journal of the Operational Research Society*, 426-438, 56, (4), 2005.

[12] R. Kruse, Ch. Borgelt, F. Klawonn, Ch. Moewes, M. Steinbrecher, P. Held, "Computational Intelligence", Springer, 2011.