timetabling, scheduling, evolutionary algorithms

Maciej NORBERCIAK*

ARTIFICIAL INTELLIGENCE TECHNIQUE FOR PLANNING DUTIES IN HOSPITAL – PRELIMINARY RESULTS

Scheduling doctors duties in a hospital are complicated and time-consuming tasks. The person responsible for creating a duty timetable is facing one major problem when allocating doctors to time periods: the agreement between several constraining (and often mutually excluding) requirements must be found. In this paper a solution methodology for the monthly duty assignment of doctors is presented. The typical problem is described in detail, along with specific hospital environment, from which datasets for experiments have been taken. A hybrid approach that utilizes strengths of a few artificial intelligence techniques was used to solve the problem. In particular, a population of initial solutions is generated heuristically and then improved using evolutionary algorithm. Experimental results are presented along with a discussion on the computational efficiency, operational acceptability and quality of the solutions.

1. INTRODUCTION

The typical timetabling problem consists in assigning a set of activities/actions (e.g. work shifts, duties, classes) to a set of resources (e.g. physicians, teachers, rooms) and time periods, fulfilling a set of constraints of various types. Constraints stem from both nature of timetabling problems (e.g. two events using the same resources cannot be planned at the same time) and specificity of the institution involved. In other words, timetabling (or planning) is a process of putting in a sequence or partial order a set of events to satisfy temporal and resource constraints required to achieve a certain goal, and is sometimes confused with scheduling, which is the process of assigning events to resources over time to fulfil certain performance constraints (however, many scientists consider scheduling as a special case of timetabling and vice versa) [17].

Timetable problems are subject to many constraints that are usually divided into two categories: “hard” and “soft”. Hard constraints are rigidly enforced and have to be satisfied in order the timetable to be feasible, for example no resource can be demanded to be in more than one place at any one time. Soft constraints are those that are desirable but not absolutely essential (e.g. an event may need to be scheduled in a particular time period or one event may need to be scheduled before/after the other). In real-world situations it is usually impossible to satisfy all soft constraints (often because they are mutually excluding.

* Department of Computer Science, Wroclaw University of Technology, *norberciak@ci.pwr.wroc.pl*

Timetabling problems arouse interest of many researchers. Although most commonly educational timetabling (e.g. university course or examination timetabling) and job-shop type problems are dealt with, the planning and scheduling in the medical hospital environment are also quite popular to be seen about. Recent research concerns medical therapy planning, operation theatres scheduling, drug logistics and personnel (particularly nurses and physicians) shift allocation [4] [9] [12] [18] [19]. Artificial Intelligence (AI) research community is quite active in the area of timetabling and scheduling and has developed a variety of approaches for solving such problems. They can be roughly divided into four types [3]:

- sequential methods – these methods order events using domain heuristics and then assign the events sequentially into valid time periods (also called timeslots) so that no events in the period are in conflict with each other; events are most often ordered in such way that events that are most difficult to schedule are assigned into timeslots first (this course of action is called direct heuristic based on successive augmentation) [2];
- cluster methods – in this methods events are collected in clusters where any two events in a particular cluster do not conflict with each other; the main drawback of these approaches is that the clusters of events are formed and fixed at the beginning of the algorithm and that may result in a poor quality timetable [14];
- constraint-based approaches – in these methods a timetabling problem is modelled as a set of variables (i.e., events) that have a finite domain to which values (i.e., resources such as time periods) have to be assigned to satisfy a number of constraints; a number of rules is defined for assigning resources to events and when no rule is applicable to the current partial solution a backtracking is performed until a solution is found that satisfies all constraints; as the satisfaction of all constraints may not be possible, algorithms are generally allowed to break some constraints in a controlled manner in order to produce a complete timetable [8][20];
- meta-heuristic methods – variety of meta-heuristic approaches such as simulated annealing, tabu search, evolutionary algorithms and hybrid approaches have been investigated for timetabling; meta-heuristic methods begin with one or more initial solutions and employ search strategies to find optimal solution, trying to avoid local optima in the process [5][6][14][15][17][19].

Recently the application of case-based reasoning to timetabling has become increasingly popular [1] [10] [11].Most approaches use heuristics because traditional combinatorial optimization methods often have a considerable computational cost. Although they can produce high quality solutions, they are not suitable for solving large, highly constrained problems. It is believed that AI medical planning is at a quite mature level and it is expected will soon be applied to practical systems.

This paper presents a solution to a duties planning problem in polish hospital. In chapter 2 the problem is described in detail along with constraints connected with it. Chapter 3 contains proposed framework’s description. Results of experiments with this approach are presented in fourth chapter. Last chapter contains summary and points to direction of future work.

2. THE PROBLEM

A typical hospital department employs about a dozen or so physicians of various specialties. On each day one or more doctors has a duty. Number of doctors on duty may vary from day to day. A planning horizon (i.e. a period of time for which the problem must be solved) amounts one month. If specialties of physicians in particular department are not homogenous (e.g. casualty ward employs surgeons and anaesthesiologists) there are often requirements for specialty of doctors on duty. The following hard constraints are defined:

- all the timeslots (i.e. days) have a proper number of physicians of appropriate specialties assigned;
- no physician has a duty in two (or more) consecutive days;
- no physician has more than two duties in the week;
- at least one physician on each duty is able to perform duties single-handed (that means that particular doctor has a certain degree of medical education and is experienced and responsible enough).

In order to consider and model fairness and job satisfaction issues, the following soft constraints are introduced:

- all the physicians have more or less the same number of duties assigned;
- weekends are fairly distributed among the physicians – duties on weekends are most often paid better than on other days; on the other hand most people like to have free weekends; as the matter of fact the same issue concerns Thursdays (because after Thursday’s duty Friday is free) and Fridays (because duty on Friday means spoiled Saturday);
- physicians have duties on preferred days of the month and, symmetrically, they have no duties assigned in timeslots they don’t want to have duties;
- if more than one physician has a duty assigned in particular time period social preferences are taken into consideration (doctors have duties with persons they like).

Duties on special days, like Eastern, Christmas, New Year’s Eve, etc. are preassigned, according to schedules from previous years so no one has duty on the same holiday two years in a row.

3. AUTOMATIC PLANNING METHOD

Approach described in this chapter is based on evolutionary algorithm (EA) framework, which turns out to be useful as a general-purpose optimization tool, due to its high flexibility accompanied by conceptual simplicity. Unfortunately, ‘classic’ EA are not appropriate tool for solving constrains satisfaction problems (CSP), like the one described in previous chapter. It stems from many factors. EA evaluation function in case of CSP is often penalty-based, and slight changes (like mutation) in particular genotype result in small changes in evaluation function, so the selective pressure is rather weak. Also, changes may cause not only improvement, but also deterioration of the solution in terms of the evaluation. Forcing bigger changes (by introducing macromutation or altering the form of evaluation function) is not very effective, because it shakes the whole evolution process. Enlargement of population size may work as a remedy for weak selective pressure but it comes with extended computational cost. All these factors support modification or even elimination of "classic" genetic roulette, leaving population-based representation of solutions and genetic operators form EA framework. Penalty-based evaluation function in such cases becomes a tool not only evaluating solutions but also directing the genetic operators [16].

Each physician has the following data assigned:

- social preferences – for each other physician one of five possible social preferences is defined (in order from most to least desirable): "absolutely", "yes", "no matter", "no" and "no way", where "absolutely" means that physician wants to have duties only with that partner and "no way" means that physician doesn't want to have duties with that partner at any circumstances; the social preferences are symmetric (if doctor *a* has particular preference for doctor *b*, doctor *b* must have the same preference for doctor *a*);
- time preferences – for each day of the month that plan is made for one of five possible time preferences is defined (as described above);
- independence flag – shows if a physician can perform duties single-handed;
- if some (or possibly all) duties need heterogeneous staff (doctors of different specialties) to be assigned, physician's specialty is kept.

For each day of the month desired number of physicians on duty, along with their specialties (if applicable) is defined. The most and the least desirable time and social preferences are considered hard constraints (so they must be satisfied in order the timetable to be feasible) and the others are soft. The dataset is verified for consistency before timetabling process starts to avoid situations where e.g. no one wants to have duty together with particular person, on particular day or at least one physician in a pair which is bound together with "absolutely" preference is able to perform duties single-handed.

Specimen (genotype) is represented directly – each time period (day) is represented by list of doctors who have a duty assigned on that day. Genotype length is constant for particular timetable (e.g. if duties are planned for May and every day there are two physicians on duty, genotype has a length of 62).

As it was shown in [7] and [16] genotype initialization strategies which produce feasible timetables give better effects than random or non-feasible ones. This is understandable – it's much easier to optimize function of soft constraints than to seek feasible solution first and then optimize it (or seek and optimize simultaneously). Hence the heuristic genotype initialization procedure has been worked out. First, an "empty" genotype (without any doctors assigned) is generated. Then the physicians with the strongest time preferences ("absolutely") are assigned appropriate timeslots, along with their "absolutely" preferred partners. The remaining free duties are assigned random physicians in such way that no physician has two duties in a row, no hard time or social preferences are violated and physicians have appropriate specialties (if applicable). The probability of assigning duty to a doctor is proportional to the strength of his preferences concerning particular day and/or partners, so doctors with more positive preferences are more probable to be chosen than those with neutral or negative ones. This procedure does not take into consideration hard constraint concerning having more than two duties in a week.

It has to be emphasized that the procedure described above does not guarantee producing a solution. If no physicians to assign a particular duty without breaking some hard constraints can be found, backtracking is performed. A random number of randomly chosen duties are cleaned of any assigned doctors and the day before the time when procedure failed to find any appropriate physician is cleaned completely (unless strong time preferences are involved). However, in the experiments conducted on real data there was no need to use this backtracking method.

Penalty-based evaluation function was used. Penalty for genotype *g* amounts

$$f_g = \sum_{i=0}^{i<d} \sum_{j=0}^{j<pd_i} spr_{ij} + tpr_{ij} + wp_{ij} + dp_{ij} \tag{1}$$

where *d* is the number of days in the month which timetable is being made for, pd_i is the number of duties on day *i*, *spr* is a penalty/reward for a physician on duty *j* having negative/positive preference for all other physicians in day *i*, *tpr* is a penalty/reward connected with time preferences (as above), *wp* is a penalty for physician having more than two duties in a week (this penalty is not in effect, if physician has “absolutely” time preference for day *i*; in this case penalty is imposed on previous or next doctor’s duty on particular week) and *dp* is distribution penalty (applicable only to Thursdays, Fridays and weekends – this penalty is imposed if doctor has more or less duties than average on particular day of the week; this penalty grows exponentially – for each duty more/less than average penalty is to times greater).

Value of the evaluation function for solution *g* is calculated by dividing the lowest penalty value in the population by penalty value for *g*.

$$F_g = \frac{f_{\min}}{f_g} \tag{2}$$

After generating the initial population the evolutionary algorithm begins to operate. Creation of population in subsequent generations (iterations) is archived by means of classical genetic roulette, as described in [13], but 20% of the population is always preserved from previous generation. 10% consists of best solutions (in terms of evaluation function described above). The remaining 10% are the solutions that are most distant from the rest of the population, in order to preserve population diversity. The distance between two timetables is measured in pairs of physicians that have been assigned duty on the same day in both timetables, as described in [3]. The higher is the score, the smaller is the distance between timetables. This method is favoured because it allows to represent diversity as a single value average and did not have the drawback of method where absolute positions of the events in timetables are considered (physicians that have duties on the same day in both timetables are counted).

Once every iteration of the algorithm a mutation operator is applied to each solution. The operator changes physician assigned a particular duty to another one in such way that all hard constraints remain satisfied. The duty to mutate is chosen at random but the higher the penalty imposed on duty, the higher the probability this duty is chosen. Computation ends after a set number of iterations or if there is no improvement in the evaluation function of the best solution in three subsequent generations. If penalty imposed on any solution reaches 0 or less that solution is saved and removed from the population.

4. IMPLEMENTATION AND EXPERIMENTAL RESULTS

The approach described above has been implemented using Microsoft Visual C++ 7.0, and the experiments have been conducted on Athlon 2100+ class machine with 512 MB of RAM. The population of 500 solutions have been used. The following values of penalties and rewards have been assumed:

- penalty for negative time or social preference – 10 ;
- reward for positive time or social preference – -5;
- penalty for having more than two duties in a week – 15;
- base of distribution penalty – 15 (so if doctor has 1 duty more or less than doctors have on average the penalty is 15; if 2 more/less – 30, 3 – 60 and so on).

The experiments were conducted on nine datasets. First was a real dataset from Department of Neurosurgery of University of Medicine in Wroclaw and contained data on 10 neurosurgeons (3 of them can’t work alone). No time preferences and only two social preferences have been defined (as “no”). There were two doctors on duty on odd days of the month and one doctor on even ones. The rest of datasets was created artificially, adding two random social preferences for each doctor and 5 random time preferences to the dataset with real data (datasets 2-5) or five random social preferences and 10 random time preferences for each doctor in real dataset (datasets 6-9).

The results are presented in table 1. The numbers presented in the table are average of ten measurements. Second column show number of iterations after first solution with 0 or less penalty has been found. The third column holds average penalty of the entire population after finishing the computation and the fourth column contains average distance between solutions with 0 or less penalty after finishing the computation. Number of 0 or less penalty solution after algorithm stopped is in the last column.

Table 1. Experimental results

Dataset	Average iterations	Average penalty	Average distance	0 or less solutions
1 (real data)	160	47.8	7.6	63
2	217	69.3	9.2	31
3	267	77.1	4.1	53
4	232	43.2	4.3	56
5	199	81.8	9.2	23
6	631	95.3	12.2	9
7	939	57.3	3.8	33
8	715	73.4	4.4	21
9	688	104.2	11	11

The problems solved were rather simple so the algorithm was able to find solutions with 0 or less penalty quite fast. However, it has to be emphasized that less constrained problem often means larger search space and to determine what part of search space has been visited further study is required. Final operational acceptability and subjective quality of the solutions with 0 or less penalty had to be determined by human. Only results from the real dataset were taken into consideration, because it was hard to determine quality of the solutions produced for artificially altered datasets. The overall quality of the solutions has been considered good and the system is ready to operate in real hospital environment. Average algorithm run time was about fifteen minutes, so computational efficiency of the method is acceptable. The scalability of the approach is yet to be determined in future research.

5. CONCLUSIONS AND FUTURE WORK

The automatic duty scheduling system described above is currently being introduced in real hospital environment. The results are quite appealing and the method used shows good promise for the future. Although it has been implemented and tested for one particular problem variation, the framework used is universal and can be used to solve other similar problems. Future work should concentrate on testing some functions to measure distance between solutions other than described in the paper. That should allow finding a way to improve diversity of final population and coverage of the search space. Using other functions than presented (e.g. Fibonacci sequence) in evaluating distribution penalty also seem interesting along with adding more sophisticated genetic operators that use some form of local search. Finally, more test data sets have to be obtained in order to prove generality of the method.

BIBLIOGRAPHY

[1] BURKE E.K., MACCARTHY B., PETROVIC S., QU R., Structured cases in case-based reasoning – re-using and adapting cases for time-tabling problems. Knowledge-Based Systems 13, 2000
[2] BURKE E. K., NEWALL J. P., WEARE R. F., A Simple Heuristically Guided Search for the Timetable Problem, Proceedings of the International ICSC Symposium on Engineering of Intelligent Systems, ICSC Academic Press, Nottingham, 1998
[3] BURKE E.K., PETROVIC S., Recent research directions in automated timetabling, European Journal of Operational Research 140, 2002
[4] CHEANG B., LI H., LIM A., RODRIGUES B., Nurse rostering problems – a bibliographic survey, European Journal of Operational Research 151, 2003
[5] COLORNI A., DORIGO M., MANIEZZO V., Genetic Algorithms and Highly Constrained Problems: the Time-Table Case, Proceedings of the First International Workshop on Parallel Problem Solving from Nature, Lecture Notes in Computer Science 496, 1990
[6] COLORNI A., DORIGO M., MANIEZZO V., Genetic Algorithms: a New Approach to the Time-Table Problem, Lecture Notes in Computer Science – NATO ASI Series, Vol. F 82, Combinatorial Optimization, 1990
[7] CORNE D., ROSS P., Puckish Initialization Strategies for Evolutionary Timetabling, Proceedings of the First International Conference on the Theory and Practice of Automated Timetabling, Napier University, Edinburgh 1995
[8] DO M.B., KAMBHAMPATI S., Planning as constraint satisfaction: Solving the planning graph by compiling it into CSP, Artificial Intelligence 132, 2001
[9] ERNST A.T., JIANG H., KRISHNAMOORTHY M., SIER D., Staff scheduling and rostering: A review of applications, methods and models, European Journal of Operational Research 153, 2004
[10] FOULDS L.R., JOHNSON D.G., SlotManager: a microcomputer-based decision support system for university timetabling, Decision Support Systems 27, 2000
[11] LEE S.-J., WU C.-H., CLXPERT: A Rule-Based Scheduling System, Expert Systems With Applications, Vol. 9, No. 2, 1995
[12] MARINAGI C.C., SPYROPOULOS C.D., PAPATHEODOROU C., KOKKOTOS S., Continual planning and scheduling for managing patient tests in hospital laboratories, Artificial Intelligence in Medicine 20, 2000
[13] MICHALEWICZ Z., Genetic Algorithms + Data Structures = Evolution Programs, Springer Verlag, 1996
[14] MYSZKOWSKI P., NORBERCIAK M., Evolutionary Algorithms for Timetable Problems, Annales UMCS, Sectio Informatica, vol. I, Lublin 2003
[15] NEWALL J. P., Hybrid Methods for Automated Timetabling, PhD Thesis, Department of Computer Science, University of Nottingham 1999
[16] NORBERCIAK M., Feasible genotype initialization for evolutionary timetabling, Proceedings of 9th International Conference on Soft Computing MENDEL 2003, Brno 2003
[17] ROSS P., CORNE D., Comparing GA, SA and Stochastic Hillclimbing on Timetabling Problems. Evolutionary Computing; AISB Workshop, Sheffield 1995, Selected Papers, ed. T. Fogarty, Springer-Verlag Lecture Notes in Computer Science 993, 1995
[18] SPYROPOULOS C.D., AI planning and scheduling in the medical hospital environment, Artificial Intelligence in Medicine 20, 2000
[19] VALOUXIS C., HOUSOS E., Hybrid optimization techniques for the workshift and rest assignment of nursing personnel, Artificial Intelligence in Medicine 20, 2000
[20] YAKHNO T., TEKIN E.: Application of Constraint Hierarchy to Timetabling Problems, Proceedings of EurAsia-ICT 002, Springer-Verlag, 2002

The author would like to thank Dr. Artur Weiser from Department of Neurosurgery of University of Medicine in Wroclaw for his constructive comments and extensive help provided.