Computer-Aided School and University Timetabling: The New Wave

Victor A. Bardadym

International Renaissance Foundation
PO Box 358/3
Kiev-30 252030 Ukraine
bardadym@issep.freenet.kiev.ua

Abstract

During the last five years a peak of interest has been observed in the problems related to computer-aided timetabling. The most recent works in this area are based on the application of modern information technologies. Here the main directions of modern research and design are reviewed. A classification is proposed for academic timetabling problems, requirements for the timetables, mathematical models, solution methods, data representation, and interface design. Modern problem solution and software design approaches are represented in connection with the theoretical background and world experience of 35 years. The integration of several algorithmic and interactive tools is discussed. Criticisms of timetabling computer support practicability and related myths are also reviewed. Related topics, like sport scheduling and scheduling sport, are represented. Some unsolved matters are outlined.

1 Introduction

Various scheduling problems in education including construction of lectures' and exams' timetables, and course and classroom scheduling are among the most difficult in educational planning. The difficulties of a typical scheduling problem arise from its large scale, great number of contradictory requirements, constraints and criteria of assignments' quality. The sufficient difficulty for the scheduler is the large amount of routine work, namely finding acceptable alternatives, requirements' fulfilment analysis, looking for mistakes, and filling out several report forms (tables of classes, teachers, and classrooms).

The preparation of a class-teacher timetable for secondary school may need up to two weeks, and the mistakes found during the teaching term may cause sufficient changes in a timetable. Carter [15], Samofalov and Simonenko [64] mentioned that the manual preparation of university timetables may take up to one month.

The interest of school and university administrations in timetabling automatisation has caused numerous challenges to this problem, and problem statements of automated school/university timetable design are well known. But school and university timetabling problems remain a reality for Operations Research and Computer Science in spite of numerous attempts to solve them.

The timetabling problems can be sufficiently varied in different universities depending on their specific requirements and conditions. Due to this, timetabling systems usually are applied only in the institutions where they were designed.

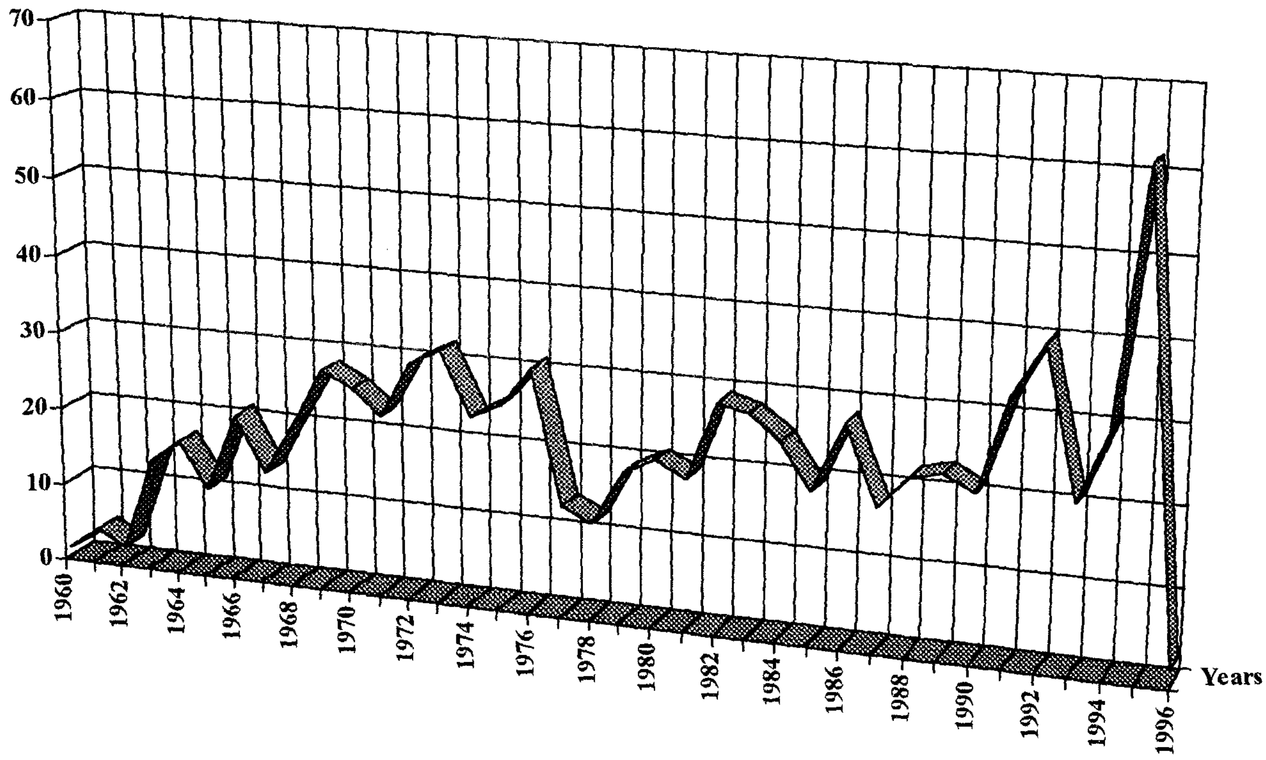

Fig. 1. The distribution of publications related with timetabling by years

Nevertheless, some timetabling systems have been distributed in numerous institutions, mostly in secondary schools [43].

The first attempts at computer-aided timetabling were made in the early 60's when universities obtained computers. Since that time many researchers and programmers have challenged this problem. Author knows more than 700 published writings dealing with these problems, including monographs, results of theoretical analyses and computational experiments, conceptual projects of software tools, articles describing experience in design and usage, surveys and bibliographies, theses for degrees, discussional papers, technical documentation, commercial catalogues, advertising, and private communications. Naturally, all of them cannot be mentioned here, and it is mostly pioneering works, new approaches, or original points of view are quoted in this paper.

Research in school timetabling has brought about many significant results in combinatorics, discrete optimisation and heuristic approaches, as well as technical solutions of its implementation. These results have proved useful in several theoretical and applied fields, and the problem of academic timetabling can be considered a model for the more general problems of decision making, scheduling, etc.

At present, many timetabling researchers world-wide are continuing their studies and software designs at a new level of quality using recent successes in artificial intelligence, decision making, discrete optimisation, and information technologies, as well as the power of modern computers and software. During the last five years a peak in publications related to timetabling and based on new approaches has been observed (See Fig. 1).

The aims of this paper are:

(i) to classify the problem statements, timetable requirements, formalisation approaches, and solution methods;

(ii) to study the logic of 35 years of timetabling software evolution, reasons and consequences of some decisions;

(iii) to find the emerging analogies and impacts between problems and solution approaches having quite different roots;

(iv) to formulate the concept of "new wave" timetabling software tools; and

(v) to outline some problems for future studies.

2 Timetabling Problems Classification

The common educational timetabling problems can be classified as follows:

1) *Faculty Timetabling.* A set of instructors, a set of lessons, and availabilities of instructors to teach several courses are defined. The problem is to distribute courses between the instructors under specified conditions.
2) *Class-Teacher Timetabling.* This problem considers a set of classes, and a set of teachers. Each class is a set of students having a common curriculum and studying together. For several pairs *<class, teacher>* the required lessons and their numbers are defined. Each single lesson must be assigned to the time period

in such a way that no teacher and no class should have two or more simultaneous lessons.

3) *Course Scheduling.* A set of students is given; students are not grouped into classes having fixed curricula. For each student the set of lectures is defined. The lectures must be scheduled to time periods in such a way that no student can have two or more simultaneous lectures.
4) *Examination Timetabling.* The set of students (classes) and the sets of examinations for each student (class) are given. Each examination must be assigned to a time period so that no students (classes) should have two or more exams simultaneously.
5) *Classroom Assignment.* When the class-teacher lessons or student-professor lectures are scheduled, they must be assigned to classrooms according to given availabilities. No classrooms should be used simultaneously by different classes.

Sometimes these problems are considered together or in combination with other educational planning problems, like curricula development or student advising. This leads to the inclusion of corresponding solvers or support tools into integrated computer-aided management systems assuming preliminary scheduling as well as real-time dispatching.

3 Main Timetable Requirements

In addition to above conditions, different general and specific requirements must be satisfied. The main requirements of the ultimate schedule may be classified as follows (See Fig. 2):

(i) *Completeness,* i.e. :

1) complete correspondence of timetable to curriculum. The curriculum is considered here as the set of lessons to be scheduled with specified multiplicities per whole planning period (usually a term, fortnight, or week). There are many ways to define a single curriculum item. The most common items are:

   <class, subject, multiplicity>,
   <class, teacher, multiplicity>,
   <class, subject, teacher, multiplicity>,
   <class, subject, teacher, classroom, multiplicity>,
   <student, subject, multiplicity>,
   <student, subject, professor, multiplicity>;

2) preassignments of some lessons to specified time periods according to the requirements determined for teachers, classes, subjects, or classrooms;

(ii) *Non-contradictoriness,* i.e. absence of any types of conflicts, namely:

1) simultaneous lessons in the same students' group or in the several groups having the same students,
2) simultaneous lessons with the same teacher,

3) simultaneous lessons in the same classroom,
4) lessons for some teachers or classes when they are unavailable, and
5) assignment of lessons to classrooms unacceptable (improperly equipped) for the corresponding subject or having insufficient workspace;

(iii) *Proper Sequencing,* including:

1) continuity of the lessons for all students meaning the absence of gaps. The gap (also spoken of as "window") is a time interval when a class or teacher has no lessons during some period, and has lessons before and after this period on this day. Several special cases (for example, when a lunch break must be scheduled) can be considered as lessons on fictive subjects, but not as gaps. The absence of gaps in teachers' timetables is often desirable also.
2) sequencing requirements reflecting logical structure of courses and defined by the curriculum for lessons on different subjects, or lessons of different types (for example, labs must follow the lectures on this subject),
3) minimisation of students' and teachers' travels between different buildings (usually taking into account whether pairs of buildings are near or far to each other);

(iv) *Uniform Distribution:*

1) the number of lessons per day for each teacher, each student or class, or each pair <*class, teacher*> cannot exceed specified limit,
2) the same lessons in the same classes must be distributed uniformly over the week,
3) the disciplines must be distributed by days of the week and time intervals according to pedagogical, psychological and ergonomic requirements. For example, it might be recommended that the lessons on Mathematics be assigned to Tuesday, Wednesday, or Thursday between the second and fourth lessons; physical training and workshops be assigned to the last lessons in the day; and that easy and hard lessons would be alternated;
4) the time interval between two successive exams cannot exceed the specified minimal interval;

(v) *Others*, for example, reliability meaning maximal opportunities to change a teacher when necessary.

From another point of view, the requirements can be divided into necessary and desirable ones. This subdivision can be varied in different institutions. When the timetabling problem is formulated as one of Mathematical Programming problems, the necessary conditions determine the set of feasible solutions, and desirable conditions determine the optimality criteria.

By types					By sufficiency
Completeness	Non-contradictoriness	Sequencing	Uniform Distribution	Others	
1) correspondence to curriculum 2) preassignments	1) no student clashes 2) no teacher clashes 3) no classroom clashes 4) availability of teachers 5) availability of students 6) acceptable assignments of classrooms	1) no gaps 2) logical course sequencing 3) minimisation of travels	1) daily limits 2) uniformity within a week 3) pedagogical, psychological and ergonomic requirements	1) reliability ... etc.	1) necessary 2) desirable

Fig. 2. Classification of timetable requirements

4 Data Representation And Mathematical Models

4.1 Sources: Bookkeeping vs Mathematical Models

Due to hardware and system software poor abilities to support interface, early timetabling software tools assumed automatic timetable generation: the responsible person prepared the input information including lists of teachers, classes or students, and classrooms, curriculum, preassignments and availabilities, and special data. The software unit read this information, tried to generate a timetable, and printed out the timetable or the message about generation fault (possibly with some diagnostics concerned with input data incompatibility, or exhaustion of hardware resources or time). The reason for failure often could not be diagnosed.

The orientation to completely automatic solutions caused the development of appropriate data representation and mathematical models. Early software was usually based on one of two main approaches: simulation of timetabling by hand vs reduction of the timetable problem to some mathematical problem. Data representation for handmade timetables usually was straightforward and often was oriented to specific features of universities or schools.

4.2 Mathematical Programming and Combinatorial Approaches

Graph Colouring. The most timetabling software is based on graph representation. The curriculum and other initial information are represented by graphs, multigraphs, hypergraphs, or set of graphs. Teachers, students or classes, and classrooms are usually represented by graph nodes; lessons to be scheduled. Availabilities or possible conflicts are represented by edges, multiedges or hyperedges. The problem is to assign time periods to edges representing lessons to satisfy the requirements. For instance,

(I) class-teacher is directly reduced to colouring single edges of curriculum bipartite multigraph (possibly, with additional constraints to colouring) [27],
(ii) the timetable for a single period must be an assignment in this graph;
(iii)classroom assignment problem is reduced to assignment finding in availability bipartite graph; and
(iv)course scheduling is directly reduced to graph nodes' colouring [22, 53, 70].

Network Flows And Transportation Problems. Graph representation is not enough to take into account more complicated requirements, and graph models can be generalised. To make the lessons' distribution more uniform by subperiods (weeks in a term, or days in a week), network flows optimisation [17, 22, 57] and transportation problems [5] are used. The obtained solutions can define natural decomposition of the problem to subperiods. Schniederjans and Kim [67] consider the faculty timetabling problem to be a multicriteria assignment problem.

Other Combinatorial Structures. The first representation of this kind was proposed by Gotlieb [38]. It assumed hypergraph simultaneously representing curriculum and availabilities by means of 3-dimensional array; the timetabling was in fact reduced to a 3-dimensional assignment problem (the requirements for the subjects were not taken into account). Many later studies used the representation of Gotlieb and its generalisations as background for precise and heuristic solutions. Barraclough [10] also studied the classes' divisions into subgroups and merging into streams. Kitagawa and Ikeda [46] used 11-dimensional tuples of sets and functions to consider similar divisions and merges, classrooms' requirements and different lessons' lengths. Junginger [43] reduced timetabling to 3-dimensional transportation problem.

Some approaches are based on matrix representation, for instance, choice matrices [26, 64], or double stochastic matrices [66], but search of distinct representatives' sets is closely connected with the search of matchings in graphs. The use of Latin squares [41] appears similar to network flow methods reviewed by de Werra [22]. Kostyuk et al. [47] and Lukyanets [54] represented the timetables construction by operations of set theory.

Integer Mathematical Programming. In order to take into account the most difficult requirements, timetabling is often reduced to some general mathematical programming problem. Zarubitskaya and Samoilenko [77] formulated the integer linear programming problem to take into account the division of classes to subgroups and their merging to streams. Classroom assignment problem is also reduced to integer linear programming in [35, 37]. In more complicated cases the quadratic assignment problem can appear [4].

General Scheduling Theory. Network planning [39, 60] and calendar scheduling representations [9, 48, 74] were used when lesson sequencing was sufficient, or when travels between classrooms in different buildings were to be minimised, or when lessons were assumed to have different lengths. These methods are similar to scheduling methods used in computer-aided manufacturing and often referred to assignment or network flow problems as subproblems.

4.3 The Features of New Wave

Mathematical Logic appeared in some early attempts to solve timetabling problems but had no significant success due to restricted abilities of computers and high complexity of general problems. In recent years the successes in expert systems have attracted researchers to use these representations with new approaches to problems' solution [36, 44, 49].

Database Representation. At present general purpose database management systems and computer-aided software engineering tools (like FoxPro) are widely available, and many designers apply standard database structures (usually a set of relational database tables, forms, reports and utilities) with simple 'bookkeeping' heuristics simulating handmade timetable construction. This approach is often used for timetabling modules in general management information systems of universities and schools. Database representation is sometimes also combined with the above mathematical models [6].

Bookkeeping	Graph Theory	Combinatorics	Mathematical Programming	Mathematical Logics	Databases
	1) edges colouring 2) nodes colouring 3) assignments 4) 3-dimensional assignments	1) n-dimensional arrays 2) n-dimensional tuples 3) Latin squares	1) network flows 2) transportation problems 3) 3-dimensional transportation problems 4) integer linear programming 5) quadratic programming 6) scheduling theory		

Fig. 3. Classification of mathematical models

5 Theoretical Problems in Timetabling

Existence Conditions. Timetable existence conditions were usually studied through matrix representation [46, 50, 58, 64]. Simple necessary or sufficient conditions of feasible timetable existence are useful in backtracking algorithms to reject dead end search branches [6].

Analysis of Algorithmic Complexity. Timetabling problems proved to be hard practically as well as theoretically. Although the edges' colouring of a bipartite multigraph can be obtained by O (E log E) operations, where E is the total number of single multigraph edges [19], the other sufficient requirements lead the extended problems to be *NP*-complete or *NP*-hard. Even et al. [27] proved the *NP*-completeness when some single periods are unavailable for several teachers. De Werra [22] mentioned that this problem is mutually reducible with the problem assuming lessons' preassignments. Also in this paper the course scheduling problem was considered as well known *NP*-complete graph nodes colouring problem. Bardadym [8] studied the problems combining class-teacher and course scheduling. In these problems the classes were considered as divisible into subgroups and able to be combined into lecture streams. The timetabling problems appeared to be *NP*-complete both for the whole planning period and for single periods. Cater and Tovey [13] studied the complexity of several classroom assignment problems; the single-period optimal assignment problem was shown to be polynomially solvable, and the noninterval feasible and interval satisfice problems were shown to be *NP*-complete.

Many of the above-mentioned works also reduce timetabling to problems well known as *NP*-complete or *NP*-hard (integer linear programming, 0-1 integer linear programming, three-dimensional transportation problem, quadratic assignment problem, travelling salesman problem, graph vertices colouring) (Garey and Johnson, 1982).

Estimation of Timetable Quality. The great number of contradictory requirements may necessitate the omission of some constraints, and treatment of conflicts as unavoidable [25, 56]. When the number of conflicts is to be minimised, the problem of comparison of the influence of different requirements appears. Statistical analogies are most often used for this comparison. Estrada [26], Samofalov and Simonenko [64] studied the function estimating the timetable quality via mean square deviation between proposed and desired timetables. Smith [71] defined the timetable quality by the entropy depending on the number of satisfied requirements.

6 Solution Methods

6.1 Precise Methods

The timetabling problem was investigated simultaneously with numerous theoretical and applied problems, and this caused the researchers to use general solution methods for mathematical programming problems, including backtracking, branch-and-bound, Gomory algorithms, dynamic programming, successive variants' analysis, etc. However, precise methods were found to be too inefficient to solve real scheduling problems. Only few-dimensional problems or problems in which certain sufficient requirements are discarded usually can be solved precisely [27].

Due to this reason a great majority of timetabling software units have been based on heuristic problem solvers.

6.2 Heuristics

Simulation of Handmade Ttimetables' Construction. The first heuristic approaches in timetabling simulated the work of a person making handmade timetables, and usually were based on common sense. These algorithms may be classified as follows:

1) greedy algorithms selecting each assignment by some local optimality rule and making no changes in lessons assigned previously. These algorithms cannot return from dead ends when no lessons can be assigned;

2) choosing each assignment with some local optimality rule, and changing earlier assignments when no lessons can be assigned for the period at hand. This approach can be treated as incomplete backtracking with restrictions on local variants' analysis;

3) algorithms finding the possibility of each given assignment simultaneously with the reassignment of conflicting lessons assigned earlier [10]. If the elementary reassignment proves impossible, some algorithms try to find alternating paths to reassign a set of lessons and to assign more lessons. This is called "pushing and pulling" by schools administrators, and in fact this heuristic is a restricted variant of known augmenting paths methods [61]; and

4) decomposition, i.e. reduction of the given problem to similar problems with smaller cardinality, and applying to these subproblems the same decomposition (if possible) or other methods [2]. The most applicable decomposition is natural decomposition by time periods (like weeks or days).

These heuristics, with numerous modifications and combinations, have been applied in many timetabling programs. The variety of heuristics is defined by some decision rules or preference rules choosing the local search directions.

Decision Rules and Preference Rules. The simplest rules to define the local search step choose the first available assignment or a randomly defined assignment. More common are decision rules [3, 11] or rules optimising some preference function [62]. The order can be defined statically, i.e. in the form of pre-processing without possibility of further changes; or dynamically, i.e. taking into account the lessons assigned earlier at the time of each assignment. The recursive rules may be used to prognosticate the influence of a choice on the ability to choose the next assignments properly. Instead of deterministic rules, randomised choice can be applied. It can be based on the Monte Carlo approach, or can use analogies with mixed strategies in games theory [62] where the probability of search direction choice is proportional to some preference function.

Simultaneously with the suggestions for obtaining feasible solutions, decision rules and preference functions can take into account the psychological, pedagogical and ergonomic requirements of the timetable (difficulty, sequencing, and uniformity). These rules and functions should be defined by experts.

A description of the most common decision rules for graph colouring was made by Carter [15]. In order to summarise the discussion of this kind of heuristics, let us note only the classification criteria:

1) largest degree first vs smallest degree last;
2) depth first vs width first;
3) static vs dynamic rules;
4) recursive prognosis vs no prognosis;
5) deterministic vs stochastic choice;
6) feasible solution search vs optimisation.

In addition to the above approaches, the heuristic algorithms for the general mathematical programming problems mentioned above, like the falling vector method for integer linear programming proposed by Ivan Sergienko and implemented in timetabling by Zarubitskaya and Samoilenko [77]; and artificial intelligence heuristic tools were applied. Loskutov [52] described the heuristic approach in terms of simulation.

Restricted versions of special methods were also used. The solution of a problem with the restricted model without guarantee of some conditions' satisfaction can also be considered as heuristic. On the other hand, the above heuristic approaches are also used to make local modifications of more general methods.

6.3 Metaheuristics: a Feature of New Wave

The principal difficulty of combinatorial problems like timetabling is the existence of numerous feasible solutions disjoint from each other and numerous local extrema. The heuristics described above often have no tools to leave local extremum, to jump over the barrier, and to reach better solutions.

In recent years new approaches have been actively studied and applied to timetabling and other problems having great practical use. They tend to find better solutions than already defined local ones, and may be subdivided in:

(i) *Mathematical Programming*

1) tabu search, assuming transfers to the locally best feasible solution with possible local change for worse to jump over barriers, and maintaining tabu list to avoid cycling [40]. Tabu search can be considered as a modification of the above mentioned falling vector method, although they were proposed independently;
2) exchange procedures [30];
3) Lagrangean relaxation [14, 74]. The connections and analogies among the above three approaches were discussed by Ferland and Lavoie [30];

(ii) *Artificial Intelligence Tools*

1) resolution rules implemented most commonly by logical programming based on PROLOG [28, 44];

2) timetabling expert systems and students advising expert systems with timetabling components [36, 49]; and

(iii) *Simulation*

1) simulated annealing, using a thermodynamic analogy to treat a feasible solution as a 'substance' (a set of 'molecules'). When the substance is heated, molecules can jump over potential barriers, and when cooled, they fill positions with minimal potential [1, 25];
2) genetic algorithms, making analogies with an evolution of a 'population' of solutions as sequence of solutions' combinations (crossovers), mutations, and selections of sets of the best solutions [12].

6.4 Interactive Timetabling: a Feature of New Wave

Database Corrections. Numerous heuristics do not guarantee that the schedule will be built even if it exists. Moreover, optimal solution of mathematical programming problem may be unacceptable due to incomplete mathematical models. These considerations and the possible necessity of changing the schedule during a term [45] have led to the development of interactive tools to correcting the input data or the timetable. In early stages these tools appeared in several systems; initially they assumed batch command mode to modify data.

Check of Handmade Timetables. A specific type of timetabling software that appeared in the first stage of the PC epoch consisted of utilities that checked the handmade timetables and formed lists of diagnostics [20, 63]. A person must analyse these diagnostics, correct the timetable with general purpose editing tools, and check the corrected timetable once more. This procedure was repeated until an acceptable timetable appeared.

Use of Universal Spreadsheets and Database Management Systems. The appearance of PC's with powerful interactive general-purpose information tools, like spreadsheets, database management systems, and integrated systems allowed new abilities to be used for interactive timetabling. General purpose text processors and spreadsheets proved useful to input, access and print timetables [16], but these common software tools did almost nothing to check timetables' correctness.

Special Timetabling Editors. When hardware and software allowed interactive work with computers, timetabling systems were completed with facilities allowing interactive choice of alternatives [31], command interface [17, 43] or prompt-response interface [75]. These facilities allowed input and correction of initial data and timetables, and also influence on the automatic problem solution by choice of alternatives, changing of critical assignments, etc.

More developed interactive tools supply the timetabler with the ability to input initial data and timetable using form dialogue boxes and spreadsheets. These tools check the data correctness, prevent the input of incorrect or conflicting data, or warn about emerging mistakes or conflicts. Common methods of database integrity, completeness, and non-contradictoriness are used for this purpose. Several types of menu, like single-choice, multiple-choice, list boxes or combo boxes (editable menu) with location by hot keys or automatic entry expand are also used to make the input easy and correct. The data structures' representation is extremely important for minimising the response time.

These systems can also be used to retrieve information necessary to design a timetable (free time intervals for classes and teachers, free classrooms, unassigned lessons, etc.). Timetabling systems of this type are often integrated and unified with other applications of management information systems and joined to the same databases.

Sometimes there are no automatic problem solvers supplied in systems of this type. As observed by Mathaisel and Comm [55], in many cases users work only with the interactive facilities and reject automatic problem solvers.

Object-Based Interface. The next generation of interactive software is connected with object-oriented programming and event-driven programming. Drag-and-drop data handling was implemented by Mathaisel and Comm [55] to simulate a method of handmade timetable composition using a desktop sheet with cells and a deck of cards representing lessons to be assigned. The cards must be put to the cells of the sheet. In computer implementation these 'cards' (windows with slots filled by lesson attributes) may be picked up, moved and placed by mouse.

When the interface is designed in this manner, the correctness of the timetable is checked after each action; in case of conflict or requirement failure, the system warns the user. Another way to support correct data handling is to lock and to hide impossible lesson assignments. Also certain total values are updated and displayed after each action. Navigation tools for the spreadsheet and decks for scrolling, search and direct access are also assumed.

6.5 Integrating of Interactive and Algorithmic Tools in Decision Support Systems

This section conceptually describes a modern timetabling software system based on the concept of decision support systems. The system integrates:
1) internal data structures and/or access to external databases;
2) object-based user interface with scrolled windows, special spreadsheets, editable two-dimensional menu, forms, drag-and-drop facilities; and
3) problem solvers for automatic generation of timetables based on mathematical programming, artificial intelligence, or simulation metaheuristics.

The person working with the DSS is able to enter and change/edit initial data and parametric settings. The input/editing operations get data from menus or tables when it is possible rather than input alphanumeric data from the keyboard. This manner takes less time and involves fewer misspells. The possibility of simultaneous location of menu items by their string identifiers is assumed.

The timetable is constructed/edited within table interface (spreadsheet) where, for instance, students groups are ranged horizontally, and days/lessons vertically, and each cell is a dialogue box consisting, for example, of a teacher's name, subject, classroom and the quality estimating number (weight function value) of a single assignment. One can construct a handmade timetable from the very beginning, or edit or complete a timetable used previously or generated automatically by picking data up from decks of 'cards' or menus and putting data to table form.

The satisfaction of requirements can be managed for the different timetabling stages as follows. Any requirement and constraint check may be switched to *On/Warning/Off* mode. If switched *On*, the check prohibits and hides choices conflicting with the requirement or contradicting with previously chosen items. In *Warning* mode all cells involved in a conflict receive "hot" status and become visually highlighted. The warnings appear when hot cells exist on the sheet or new hot cells have appeared. The hot cells may be located and operated upon. At any time automatic generation methods are available to propose variants of completing or continuing the construction. The initial data and timetables are stored in external memory, and the system obviously may be used for information retrieval and dispatching (free classroom search for an event, looking for changes when a teacher is ill, etc.).

The problem solver can assume batch and interactive mode with requests to choose alternatives, solution methods, or search directions, or to adjust parameters.

Precise	Heuristics		Metaheuristics	Interactive	
	by algorithm	by local rules		by facilities	by organisation
1) backtracking 2) branch-and-bound 3) Gomory algorithms 4) dynamic programming 5) successive analysis	1) greedy algorithms 2) decision rules 3) preference functions 4) incomplete backtracking 5) incomplete augmenting paths 6) decomposition	1) largest degree first vs smallest degree last 2) depth first vs width first 3) static vs dynamic rules 4) recursive prognosis vs no prognosis 5) deterministic vs stochastic choice 6) feasible solution search vs optimisation	1) mathematical programming • tabu search • exchange procedures • Lagrangean relaxation 2) artificial intelligence • resolution rules 3) simulation • simulated annealing, • genetic algorithms	1) database corrections 2) universal spreadsheets 3) special editors	1) choice of alternatives 2) command interface 3) prompt-response interface 4) popup/pulldown menu, lict boxes and combo boxes 5) object-based interface, drag-and-drop
Combined (expert systems, decision support systems)					

Fig. 4. Classification of solution methods

7 Survey of Surveys

As stated above, timetabling has attracted hundreds of researchers, and it is natural that surveys and bibliographies have followed.

The best-known published representative timetabling bibliography is [66]. The bibliography contains more than 200 references followed by references to most popular review magazines. An annotation classifies problems and solution approaches.

A survey of several timetabling problems' statements, corresponding mathematical models based on graph nodes and edges colouring, and network flows with detailed descriptions and theoretical results has been written by de Werra [22].

Carter [15] considered examination timetabling, including analysis of several practical applications. It also considered wide range of heuristics applicable also to other timetabling problems, and to general graph colouring problems. Some recommendations on approach selecting were stated.

Bardadym [7] reviewed the formalisation and solution approaches, compared and studied impacts of the precise, heuristic and interactive methods, observed relations between different timetabling problems and other applications, and formulated the concept of a modern timetabling decision support system.

Some surveys described the experience of design and use of timetabling software in several countries: Germany and Austria [43], the Netherlands [20], East Europe [65], the USSR [42, 59].

8 Related Topics

8.1 Curricula Development and Planning Components in CAI/CAL

Numerous works are concerned with other problems of automated teaching planning. Works on automated curricula development mostly deal with the sequencing of items to be studied (topics, concepts or problems) according to the relations between them. The subject is represented by an oriented graph or network scheme; loops are diagnosed and/or removed, and then the order of items studying is defined. For example, a concept must be introduced only after introduction the concepts used in its definition. This sequencing is known as topological sorting [76], and it can be found by $O(N+E)$ operations, where N and E denote the numbers of graph nodes and edges correspondingly. More developed models assume optimisation on the set of topological sortings, where optimality criterion takes into account the connections between items, their studying lengths, and forgetting functions depending on time. These models are equivalent or close to the NP-complete travelling salesman problem. These approaches were reviewed in [51, 60]. Similar tools are implemented in computer-aided learning software to control the real-time learning process [72].

The application of simulation and linear programming for curricula development is described by Fedotov [29]. Petri nets were used by Ferraris et al. [33]. In recent years expert systems have been designed to develop individual curricula [24].

8.2 Similar Problems in Other Applications

Many scheduling and planning problems have mathematical models similar to timetabling: vehicle routing, resource sharing, computer-aided manufacturing, multiprocessor systems' scheduling, meetings scheduling [23], terminals scheduling [69], sport scheduling [21, 32, 68].

8.3 Scheduling Sports

Software designers' and users' competitions have been popular in recent years. School timetabling was also involved in this process. In 1993 the Russian Ministry of Education organised an open competition of timetabling software. The authors of programs had to design the class-teacher-classroom timetables with initial data from real secondary schools (usually about 50 classes x 100 teachers x 5-6 days per week x 6 lessons per day). The timetabling process had to be completed within a limited time period. Successfully completed timetables were estimated by experts - vice-principals of Moscow secondary schools responsible for timetable construction in their schools. First prize was awarded to program ASSR 2.5 by Igor Beender and Leonid Chernenko (Odessa, Ukraine).

8.4 Criticisms and Myths

35 years of computer-aided timetabling have provided no ultimate solution. This fact has led to many critical and mythical opinions. Some of them are contrary to each other. The most popular are:

1a) 'The computer is unable to make timetables' vs 1b) 'My students can design a timetabling program within a month'.

2a) 'Computer-aided timetable construction is not economical (for small schools)'; 2b) 'A handmade timetable is always better than one produced by a computer' vs 2c) 'When I have a computer, I'll have no problem with timetabling'.

3a) 'Interactive timetabling is useless without an automatic timetable generator' vs 3b) 'Interactive facilities are useful for timetabling but problem solvers are not'.

4a) 'You have bad timetable due to the computer, not to me', a vice-principal says; 4b) 'If a computer made the timetable I would be dismissed'; 4c) 'If a computer made the timetable I would lose my influence on teachers'.

The specialists need not comment. Nevertheless, let us explain our reaction to opponents' criticism:

1a), 2b), 3a): Even if you make handmade timetables using only interactive editors, you will save a lot of time through avoiding misspellings and routine work on filling paper table cells with repetition of teachers' names, timetable checks, and report making.

1b): Let them try.

2a): As derived from answer to 1a), automated timetabling is not economical only if computer and software are too expensive for the school.

2c): Be ready to have more problems than usual the first time. If you have good software, you will be completely overjoyed once you have:

- overcome the psychological barrier against the computer,
- studied how to handle data correctly,
- completed filling all necessary databases, and finally
- made your own methodology of work with your software.

Moreover, you will consider timetabling as attractive as puzzle like Patience or a Rubik cube 50x50x100.

3b): This point of view is extreme but reasonable, because solvers often leave incomplete timetables in deep search dead ends that are hard to improve manually or automatically.

4a)-4c): A computer is only a tool. It makes only things specified by man. If 'the computer' makes a gap in a teacher's timetable, the reason may be that the vice-principal 'forgot' to prohibit gaps for this teacher.

9 Unresolved Matters

The further development of scheduling DSS's and extension of their application fields will lead to the design of special software tools of specific DSS applications implementation. These tools must be based on research in the following matters of computer science:

1) design of simple language and/or interactive environment to express relationships between different items in the informational structure in order to support specific (auxiliary) requirements and constraints;
2) maintenance and visualisation of hierarchical structures allowing decomposition of interactive problem solutions according to time intervals (for example, terms, weeks, days) and object hierarchies (for example, departments, streams, groups, subgroups); and
3) research on the algebraic properties of interactive operations over informational structures; for example, research on invariants of some operations in order to preserve the data integrity including integrity of partial schedules constructed earlier.

Acknowledgements

I greatly appreciate persons who helped me to find the information about automated timetabling. Their kind help enforced the design of collection involving numerous pearls of human intelligence applied to this really intractable but necessary problem. The most sufficient contributions were made by Michael Carter (Canada), Jacques Ferland (Canada), Alexander Leibovich (Russia), Michael Pleskach (Ukraine), Hana Prochazkova (Czechia), and Valery Simonenko (Ukraine).

I would like to thank Jaime Taber for her help with preparation of this paper.

References

1. Abramson D. (1991) "Constructing School Timetables using Simulated Annealing", *Management Science*, **37**, 98-113.
2. Almond M. (1969) "A University Faculty Timetable", *Computer Journal*, **12**, 215-217.
3. Anisimov B.V. and Vlasov V.P. (1971) "Analysis of Decision Rules Applied to Lessons Timetabling", *Mathematical and Information Problems of Science Prognostication and Management*, Kiev, V.Glushkov Inst. for Cybernetics, 113-122 (Russian).
4. Aubin J. and Ferland J.A. (1989) "A Large Scale Timetabling Problem", *Computers and Operational Research*, **16**, 67-77.
5. Babich V.I. and Raetski V.I. (1991) "Models and Results of Automated Academic Timetabling", *Methods and Systems of Technical Diagnostics*, Saratov State University, 99-102 (Russian).
6. Bardadym V.A. and Vasilyeva-Sklyarova T.V. (1994) "A Decision Support System For Computer-Aided School Timetabling", *Proceedings of East-West International Conference on Computer Technologies in Education*, Crimea, Ukraine, Part 2, 33-35.
7. Bardadym V.A. (1991a) "Computer-Aided Lessons Timetables Construction. A Survey" *USIM (Management Systems and Computers)*, **#8**, 119-126 (Russian).
8. Bardadym V.A. (1991b) "The Complexity Estimate for Two Timetable Construction Problems", in: A.M.Dovgiallo (Ed.), *Computer Assisted Learning Techniques*, Kiev, Glushkov Institute for Cybernetics, 72-78 (Russian).
9. Barkan S.A. and Tanaev V.S. (1970) "About the Construction of Lesson Timetables", *Vesti Akademii Nauk BSSR (Communications of Byelorussian Academy of Sciences), Physical and Mathematical Serie*, **#1**, 76-82 (Russian).
10. Barraclough E.D. (1965) "The Application of a Digital Computer to the Construction of Timetables", *The Computer Journal*, **8**, 136-146.
11. Belova N.B. and Krasner N.Ya. (1984) "Timetabling Models and Methods", *Models and Methods for Rational Planning in the University*, Voronezh State University, 43-55 (Russian).
12. Burke E.K., Elliman D.G. and Weare R.F. (1994) "A Genetic Algorithm Based University Timetabling System", *Proceedings of East-West International Conference on Computer Technologies in Education*, Crimea, Ukraine, part 1, 35-39.
13. Carter M.W. and Tovey C.A. (1992) "When is the Classroom Assignment Problem Hard?", *Operations Research*, **40**, 28-39.
14. Carter M.W. (1989) "A Lagrangian Relaxation Approach to the Classroom Assignment Problem", *INFOR*, **27**, 230-246.
15. Carter M.W. (1986) "A Survey of Practical Applications of Examination Timetabling Algorithms" *Operations Research*, **34**, 193-202.
16. Chan S.K. and Lam C.Y. (1990) "The Electronic Spreadsheet as a Tool for Course Co-ordination in a School of Engineering", *Computers and Education*, **14**, 231-238.

17. Chahal N. and de Werra D. (1989) "An Interactive System for Constructing Timetables on a PC", *European Journal of Operational Research*, **40**, 32-37.
18. Chow K.O., Loo W.S., Kwan C.M., Martinsons M.G. (1991) "An Object-oriented Expert System in Academic Timetabling", *Proc. of National Conference on Information Technology*, Malaysia, 1991, 90-96.
19. Cole R. and Hopcroft J. (1982) "On Edge Coloring Bipartite Graphs", *SIAM Journal on Computing*, **11**, 540-546.
20. De Gans O. (1981), "A Computer Timetabling System for Secondary Schools in the Netherlands", *European Journal of Operational Research*, 7, 175-182.
21. De Werra D., Jacot-Descombes L. and Masson P. (1990) "A Constrained Sports Scheduling Problem", *Discrete Applied Mathematics*, **26**, 41-49.
22. De Werra D. (1985) "An Introduction to Timetabling", *European Journal of Operational Research*, **19**, 151-162.
23. De Werra D. (1975) "On a Particular Conference Scheduling Problem", *INFOR*, **13**, 308-315.
24. Dovgiallo A.M., Mehtiev H.B. and Petrushin V.A. (1989) "An Expert System for Curricula Development", *Methods and Systems of Technical Diagnostics,* Saratov State University, 98-101 (Russian).
25. Dowsland K.A. (1990) "A Timetabling Problem in which Clashes are inevitable", *Journal of Operations Research*, **41**, 907-918.
26. Estrada Senti V. (1986) "Structures, Algorithms and Software for University Planning in the Republic of Cuba", PhD Thesis, Kiev Polytechnical Institute (Russian).
27. Even S., Itai A. and Shamir A. (1975) "On the Complexity of Timetable and Multicommodity Flow Problems", *SIAM Journal of Computing*, **5**, 691-703.
28. Fahrion R. and Dollanski G. (1992) "Construction of University Faculty Timetables using Logic Programming Techniques", *Discrete Applied Mathematics*, **35**, 221-236.
29. Fedotov A.V. (1985) " Simulation in University Management", Leningrad State University (Russian).
30. Ferland J.A. and Lavoie A. (1992) "Exchange Procedures for Timetable Problems", *Discrete Applied Mathematics*, **35**, 237-253.
31. Ferland J.A. and Fleurent C. (1991a) "SAPHIR: A Decision Support System for Course Scheduling", Publication **#764**, Universite de Montreal.
32. Ferland J.A. and Fleurent C. (1991b) "Computer Aided Scheduling for a Sport League", *INFOR*, **29**, 14-25.
33. Ferraris M., Midoro V. and Olimpo G. (1984) "Petri Nets as a Modelling Tool in the Development of CAL Software", *Computers and Education*, **8**, 41-49.
34. Garey M. and Johnson D. (1982) "Computers and Intractability", Moscow, Mir Publ (Russian edition).
35. German E.I., Pak L.V. and Chudinov V.N. (1978) "Lecture Room Distribution in the University", *Automated Management Systems in the University*, Novosibirsk State University, 138-142 (Russian).

36. Golumbic M.C., Markovich M., Tsur S. and Schild U.I. (1986) "A Knowledge-based Expert System for Student Advising", *IEEE Transactions on Education*, **E-29**, 120-123.
37. Gosselin K. and Truchon M. (1986) "Allocation of Classrooms by Linear Programming", *Journal of Operations Research*, **37**, 561-569.
38. Gotlieb C.C. (1962) "The Construction of Class-Teacher Time-Tables", in Popplewell C.M. (ed.): *Information Processing 1962. Proc. IFIP Congress 62,* Amsterdam, North-Holland, 73-77.
39. Grigorishin I.A. (1984) "Instrumental Support of Teaching Using Computer Courses", PhD Thesis, Kiev, V.Glushkov Institute for Cybernetics (Russian).
40. Hertz A. (1992), "Finding a Feasible Course Schedule using Tabu Search", *Discrete Applied Mathematics*, **35**, 225-270.
41. Hilton A.J.W. (1981) "School Timetables", *Annals of Discrete Mathematics*, **11**, 177-188.
42. Indushkin V.L., Lysenko V.I. and Tarasov S.B. (1977) "Principles of Automated Management System Design in Higher Education", Moscow, NIIVSh (Res. Inst. for Higher Education) (Russian).
43. Junginger W. (1986) "Timetabling in Germany - a Survey", *Interfaces,* **16**(4), 66-74.
44. Kang L., Von Schoenberg G.H. and White G.M. (1991) "Computer University Timetabling Using Logic", *Computers and Education*, **17**, 145-153.
45. Kirilichev A.M., Kochetkov G.M. and Stepin Yu.P. (1976) "The Principles of Construction of Calendar Planning Subsystem for University Automated Management System", *Design and Implementation of University Automated Management System*, Proceedings of Moscow Institute for Oli and Gas Industry, **128**, 46-51 (Russian).
46. Kitagawa F. and Ikeda H. (1988) "An Existential Problem of a Weight-Controlled Subset and its Application to School Timetable Construction", *Discrete Mathematics*, **72**, 195-211.
47. Kostyuk V.I., Martines J.O. and Zorin V.V. (1976) "The Use of Successive Execution Algorithms in Timetabling", *The Problems of Automated University Management System Implementation*, Moscow, NIIVSh (Res. Inst. for Higher Education), 3-5 (Russian).
48. Knyazeva T.A. and Portugal V.M. (1969) "Computer-aided University Department Timetable Construction", *Computer-Aided Manufacturing*, Kiev, Institute for Cybernetics (Russian).
49. Kowalski K. and Ealy D. (1991) "Schedule Advisement Expert System", *Computers and Education*, **17**, 259-265.
50. Lazak D. (1968) "Die Beziehungen des Kollisionsindexverfahrens zum GOTLIEB-Verfahren, Die Ableitung eines Kriteriums fur die Existenz von Losungen bei einer Dienstplanmatrix", *Elektronische Datenverarbeitung*, **10**, 26-37, 58-61.
51. Logvinov I.I. (1980) "Teaching Processes Simulation", Moscow, Pedagogica Publ (Russian).

52. Loskutov S.I. (1983) "About Educational Timetabling", *Mathematical Methods of Management and Information Processing*, Moscow Inst. for Physics and Technology, 151-157 (Russian).
53. Lotfi V. and Sarin S. (1986) "A Graph-Coloring Algorithm for Large-Scale Scheduling Problems", *Computers and Operations Research*, **13**, 27-32.
54. Lukyanets N.B. (1980) "School Timetable Construction and Editing with the Aid of the Computer", *The Problems of Computer-Aided Management in Education*, Moscow, Res. Inst. for Pedagogics, 70-82 (Russian).
55. Mathaisel D.F.X. and Comm C.L. (1991) "Course and Classroom Scheduling - an Interactive Computer-Graphics Approach", *Journal of Systems and Software*, **15**, 149-157.
56. Mehta N.K. (1981) "The Application of a Graph Coloring Method to an Examination Scheduling Problem", *Interfaces*, #**11**, 57-64.
57. Mulvey J.M. (1982) "A Classroom/Time Assignment Model", *European Journal of Operational Research*, **9**, 64-70.
58. Neufeld G.A. and Tartar J. (1974) "Graph Coloring Conditions for the Existence of Solutions to the Timetable Problem", *Communications of the ACM*, **17**, 450-453.
59. Orlov V.N. (1976) "An Analysis of Automated Lessons Timetabling Systems", *Automated Management Systens and Higher Education*, Cheboksary State University, 68-74 (Russian).
60. Ovchinnikov A.A., Puginski A.A. and Petrov G.F. (1972) "The Network Methods for Education Planning and Organisation", Moscow, Vysshaya Shkola (Russian).
61. Papadimitriou C. and Steiglits K. (1985) "Combinatorial Optimisation", Moscow, Mir Publ (Russian edition).
62. Potashnik V.Ya. and Pichko S.P. (1977) "A Randomized Scheme for Timetabling in a University", *The Efficiency of Automated Management Systems in Education*, Kiev, Znanie Publ., 23-27 (Russian).
63. Romero B.P. (1982) "Examination Scheduling in a Large Engineering School: A Computer-assisted Participative Procedure" *Interfaces*, **12**, #2, 17-24.
64. Samofalov K.G. and Simonenko V.P. (1972) "Automatisation of University Timetabling", Kiev Polytechnical Institute Publ (Russian).
65. Saveliev A.Ya., Zubarev Yu.B., Kovalenko V.E. and Koloskova T.A. (1984) "Automated Management in University", Moscow, Radio I Svyaz (Russian).
66. Schmidt G. and Strohlein T. (1980) "Timetable Construction - an Annotated Bibliography", *Computer Journal*, **23**, 307-316.
67. Schniederjans M.J. and Kim G.C. (1987) "A Goal Programming Model to Optimize Departmental Preference in Course Assignments", *Computers and Operations Research*, **14**, 87-96.
68. Schreuder J.A.M. (1992) "Combinatorial Aspects of Construction of Competition Dutch Professional Football Leagues", *Discrete Applied Mathematics*, **35**, 301-312.
69. Sebo A. (1991) "A Particular Timetabling Problem - Terminal Scheduling", *Computers and Mathematics with Applications*, **21**, 137-156.

70. Selim S.M. (1988) "Split Vertices in Vertex Colouring and Their Application in Developing a Solution to the Faculty Timetable Problem", *Computer Journal*, **31**, 76-82.
71. Smith W.E. (1984) "Efficiency of a University Timetable - an Application of Entropy of Choice", *Bulletin of the Australian Mathematical Society*, **30**, 19-26.
72. Tretiak V.A. (1990) "The Design and Implementation of Problem-Oriented Language Tools to Design CAI/CAL Software", PhD Thesis, Kiev, V.Glushkov Inst. for Cybernetics (Russian).
73. Tripathy A. (1984) "School Timetabling - A Case in Large Binary Integer Linear Programming", *Management Science*, **30**,1473-1489.
74. Vishnevskij L.D. (1976) "About a Problem of Calendar Planning", *Kibernetika*, **#3**, 115-118 (Russian).
75. White G.M. and Wong S.K.S. (1988) "Interactive Timetabling in Universities", *Computers and Education*, **12**, 521-529.
76. Wirth N., (1989) "Algorithms and Data Structures", Moscow, Mir Publ (Russian edition).
77. Zarubitskaya T.F. and Samoilenko A.T. (1976) "On the Automated Timetabling Subsystem", *Experience of Automated Management System Design and Use in Kiev State University*, Moscow, NIIVSh (Research Inst. for Higher Education), 35-40 (Russian).