ELSEVIER

European Journal of Operational Research 96 (1997) 504–513

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

Theory and Methodology

The combinatorics of timetabling

D. de Werra

Ecole Polytechnique Fédérale de Lausanne, Département de Mathématiques, Chaire de Recherche Opérationnelle, CH-1015 Lausanne, Switzerland

Received 6 December 1995; revised 22 March 1996

Abstract

Various formulations of timetabling problems are given in terms of coloring problems in graphs. We consider a collection of simple class–teacher timetabling problems and review complexity issues for these formulations. This tutorial presentation (which is not claimed to be an exhaustive review nor a research contribution) includes a brief sketch of a tabu search procedure which handles many specific requirements and provides an efficient heuristic technique.

Keywords: Chromatic scheduling; Graph coloring; Timetabling; Course scheduling; Hypergraph; Latin squares

1. Introduction

One reason for the interest of O.R. specialists and mathematicians in timetabling is due to the combinatorial nature of the various problems arising in this context.

Mathematical approaches have been attempted by numerous authors and have led to some formulations and solution procedures in some particular instances.

On the other hand, the occurrence of timetabling problems has stimulated the development of some fields of combinatorial optimisation (for example extensions of coloring models).

We think that it may be useful for non specialists in the domain of timetabling to have a tutorial presentation of the basic combinatorial models which may be used to tackle these problems. It is by no means meant to be an exhaustive survey of the numerous approaches to be found in the literature, nor is it a research contribution dedicated to a highly specialized audience.

For the above reasons we have chosen to present some models starting from the simplest cases and to include progressively additional requirements while dealing in parallel with the combinatorial concepts which are needed. Small numerical examples are given to help the reader in grasping the essence of these models.

We give some results on complexity in order to help understanding how quickly we cross the border between easy and difficult problems.

Finally we do not intend to discuss computational and numerical results; they would simply lead us too far and in another direction.

Timetabling problems are numerous: they differ from each other not only by the types of constraints which are to be taken into account, but also by the density (or the scarcity) of the constraints; two problems of the same "size", with the same types of constraints may be very different from each other if one has many tight constraints and the other has just a few. The solution methods may be quite different, so the problems should be considered as different.

Here we shall start from the simplest problems and introduce requirements consecutively while discussing the variations of the corresponding combina-

PII S0377-2217(96)00111-7

torial models. Since they are more structured, we shall put the emphasis on classteacher timetabling.

In fact, we shall essentially deal with timetabling problems as chromatic scheduling problems, i.e., we will concentrate on graph coloring models. After having presented node coloring in graphs or hypergraphs and edge coloring in graphs, we will briefly discuss a tabu search approach which may handle many types of requirements. Such a method is now generally considered as a very efficient technique for solving many hard combinatorial optimization problems (including coloring problems); it is for this reason appropriate to mention it in this presentation of combinatorial models for timetabling.

We insist that our purpose is not to give a general survey of the various models which have been proposed but rather to focus our attention on combinatorial models and at the same time to show how extensions of classical models have been worked out to formalize and solve some type of timetabling problems. We will also present whenever it will be possible the limits between easy and difficult problems.

To keep the paper within a reasonable length, we shall not give standard definitions of graphs, they can be found in [1]. Similarly for concepts related to complexity, the reader is referred to [6]. This paper is an extended version of [20].

R		t_1	t_2	t_3
	c_1	2	2	1
	c_2		1	3

A requirement matrix

β,γ	δ,ε	α
	β	γ,δ,ε

A corresponding Generalized Latin Square

2	2	1
1	1	3
2	2	1

A regularized requirement matrix

β,γ	δ,ε	α
α	β	γ,δ,ε
δ,ε	α,γ	β

A corresponding square Generalized Latin Square

Fig. 1. Generalized Latin Squares.

2. The basic model: classteacher timetabling problem

We shall start with the so called class–teacher timetabling problem. A class c_i will consist of a set of students who follow exactly the same program. $\mathscr{C} = \{c_1, \ldots, c_m\}$ will be the set of classes while $\mathscr{T} = \{t_1, \ldots, t_n\}$ is the set of teachers. A requirement matrix $R = (r_{ij})$ gives the number r_{ij} of (one hour) lectures involving c_i and t_j for all i, j.

The problem consists in constructing (when possible) a timetable in p periods. Each class (resp. teacher) is involved in at most one lecture at a time.

It is well known that a solution exists if and only if $\Sigma_i r_{ij} \leq p$ for all j and $\Sigma_j r_{ij} \leq p$ for all i.

This is the famous theorem of König (see [1,15]). The problem may be viewed as constructing a generalized Latin square (GLS); a $(p \times p)$ Latin square (LS) is an array in which each cell (i, j) contains one of the p symbols $\alpha, \beta, \ldots, \varphi$ and no symbol appears more than once in any line (row or column). If each symbol is associated with a period, we may consider that the LS corresponds to a requirement matrix R where $r_{ij} = 1$ for each class c_i and each teacher t_j: cell (i, j) contains δ if c_i and t_j meet at period δ. Given an arbitrary R, a timetable corresponds to a GLS: cell (i, j) contains r_{ij} symbols in $\{\alpha, \beta, \ldots, \varphi\}$; they are the periods where the meetings of c_i and t_j are scheduled; each symbol occurs at most once in each row and in each column. (Notice that a GLS need in fact not be square!) Fig. 1 shows a requirement matrix R, an associated GLS and an extended GLS which is square and contains the initial one. It is always possible to transform a rectangle GLS into a larger square GLS which contains the first one, as can be seen easily.

Another formulation would be to associate to R a hypergraph $H(R) = (X, \mathscr{E})$ constructed as follows: each unit in R is a node x in the node set X; for each line (row or column) of R we introduce an edge containing all nodes associated to units in this line. An example is given in Fig. 2.

A timetable in p periods is a node p-coloring of H: each node gets one of the p colors in $\{\alpha, \beta, \ldots,$

$\varphi\}$ in such a way that no two nodes in the same edge get the same color.

We may at this stage consider with the same data R another problem which may be called the weekly problem (WP): instead of assigning the lectures at periods (of one hour on a single day) as before, we may simply assign them to days: a_i (resp. b_j) will be the maximum number of meetings for c_i (resp. t_j) in each day; K_{ij} is the maximum number of meetings c_i–t_j in each day.

A weekly schedule in p days will be associated with a generalized node p-coloring of $H(R)$: in each edge c_i (resp. t_j) there must be at most a_i (resp. b_j) nodes of the same color and in each intersection $c_i \cap t_j$ there will be at most K_{ij} nodes of the same color; this last requirement amounts to introduce new edges $c_i \cap t_j$ in H for each pair c_i, t_j with $c_i \cap t_j \neq \emptyset$.

An example of a weekly schedule is given in Fig. 3.

It is well known that the following holds [14,15].

Proposition 2.1. *A weekly timetable in p days exists if and only if*

(a) $\sum_j r_{ij} \leq pa_i$ *for each* c_i,

(b) $\sum_i r_{ij} \leq pb_j$ *for each* t_j,

(c) $r_{ij} \leq pK_{ij}$ *for each* c_i *and each* t_j.

Observe that we get back to the daily problem by setting $a_i = b_j = 1$; $K_{ij} = \infty$ for all i, j.

Finally it is also easy to observe that the above problems may be viewed as edge coloring problems in bipartite multigraphs $G = (\mathscr{C}, \mathscr{T}, E)$ associated

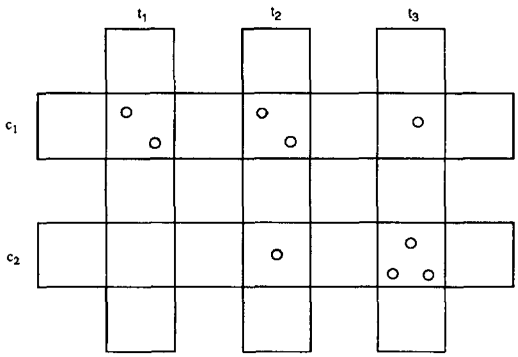

Fig. 2. The Hypergraph $H(R)$ associated to the requirement matrix R of Fig. 1.

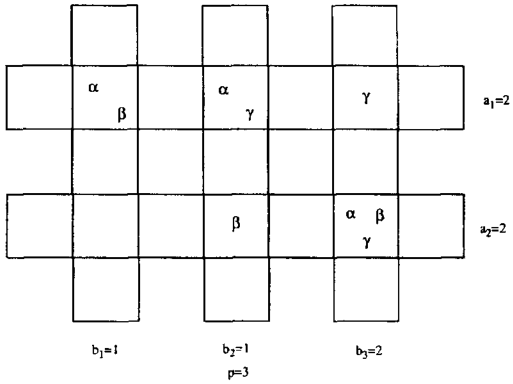

Fig. 3. A weekly schedule in $p = 3$ days for the example of Fig. 1.

to R: each c_i is a node, each t_j is a node and there are r_{ij} edges between nodes c_i and t_j for all i, j. The daily problem is a classical edge p-coloring problem in G which can be solved in polynomial time [15]. The weekly problem is a generalized edge p-coloring problem and is also easy to solve. Related results with a combinatorial flavour are given in [11].

In the next sections we shall review some additional requirements which make the problem more realistic and more difficult.

3. Preassignments and unavailabilities

In almost all real timetabling problems we have to take into account so called unavailability constraints. These can be formulated as follows in a class–teacher timetable:

Teacher t_j is available only at a subset T_j of the p periods for each j and similarly class c_i is available for meetings only at a subset C_i of the p periods. Sometimes also meetings c_i–t_j may be scheduled only in a subset M_{ij} of the p periods.

Such requirements occur in particular if a timetable is already partially constructed: some meetings have been preassigned and should not be changed later. The consequence is the presence of forbidden periods for teachers and classes: they cannot be involved in other meetings at these periods.

As can be expected these requirements make the problem more difficult; it is indeed NP-complete [4].

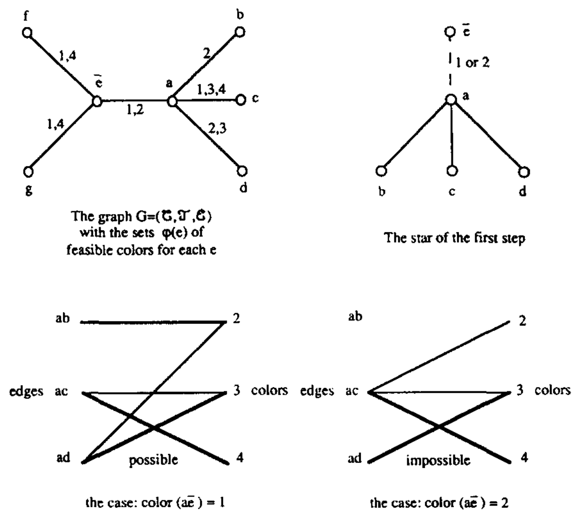

Fig. 4. Unavailability constraints.

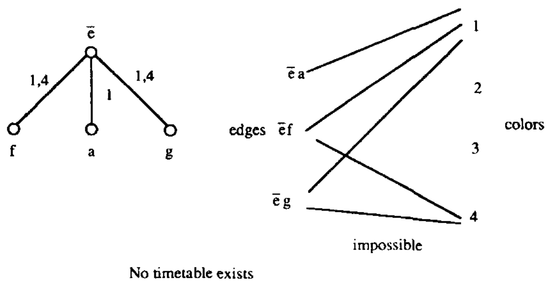

Fig. 5. Unavailability constraints (continued).

Among the known cases which may be solved in polynomial time we mention the following cases:

1. there is one class only in the problem and all teachers have arbitrary unavailabilities (solution by network flow [15])
2. only one class has unavailabilities, the remaining classes and all teachers have no unavailabilities (trivial by edge coloring)
3. for each teacher t_j we have $|T_j| \leq 2$; the classes have no unavailabilities (solution of a boolean quadratic equation by restricted enumeration, see [5,15]).

We shall now mention another situation which gives rise to a problem which may be solved easily; observe first that when taking into account all the unavailabilities of teachers and classes we may simply consider that for each meeting e represented by an edge of the bipartite multigraph $G = (\mathscr{C}, \mathscr{T}, E)$ there is a set $\varphi(e)$ of feasible periods; let (G, φ) be the resulting problem. We can now state (see [18])

Proposition 3.1. *For the class–teacher timetabling problem (G, φ) with unavailabilities, we have the following: there exists either a timetable or a proof of nonexistence if and only if G is a tree.*

We sketch the solution procedure on the example given in Fig. 4. We start by considering a star formed with pendent edges, for instance ab, ac, ad. Edge aē can have color 1 or color 2; we consider all these cases and try to color ab, ac, ad with feasible colors; in each case it is a maximum matching problem in a bipartite graph. We observe that the only possible color for aē is 1.

We go up the tree and consider the star formed with edges ēf, ēa, ēg (see Fig. 5) (if ēf and ēg had not been pendent, we would have had to deal with the corresponding subtrees first).

Then we notice from the associated edge-color graph that no assignment can be found for this star. Since the whole of G has now been considered, there is no solution. The proof of non existence of a timetable follows from a simple argument: a set Γ

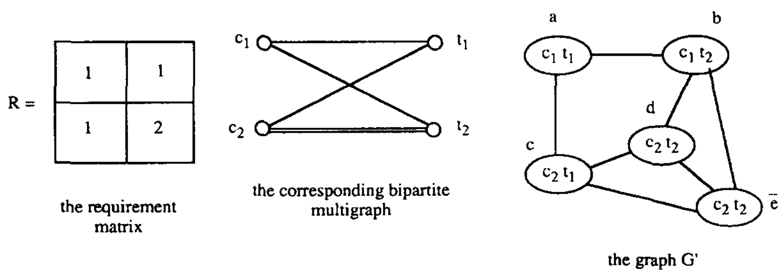

Fig. 6. Formulation in terms of node coloring in a graph G'.

of pairs (v, c) (where v is a node and c a color) can be found together with a collection $E(\Gamma)$ of edges $[v, w]$ such that for each $c \in \varphi([v, w])$ at least one of (v, c), (w, c) is in Γ; furthermore $|\Gamma| < |E(\Gamma)|$ (see [18]). Here $\Gamma = \{(\bar{e}, 1), (\bar{e}, 4), (a, 2)\}$ and $E(\Gamma) = \{\bar{e}f, \bar{e}a, \bar{e}g, ab\}$. Consider for example edge $\bar{e}f$. Coloring $\bar{e}f$ with a feasible color will set color 1 around node $\bar{e}$ (i.e. it will give $(\bar{e}, 1)$) or color 4 around $\bar{e}$ (it will give $(\bar{e}, 4)$). No two edges in $E(\Gamma)$ should give the same (v, c) in Γ. So we see that since $|\Gamma| = 3 < |E(\Gamma)| = 4$, no coloring exists. The above algorithm can be refined to produce Γ and $E(\Gamma)$ when there is no coloring (see [18]).

Another way of dealing with unavailabilities is to consider a node coloring formulation for the problem: as in the hypergraph formulation, each meeting is a node of a graph G' and we link two nodes if the meetings cannot be simultaneous (i.e., if they involve the same class or the same teacher). In other words we link two nodes if they belong to a same edge of the hypergraph. An example is given in Fig. 6.

It is also known that there exists a node p-coloring of $G' = (X, E)$ if and only if there exists an independent set of $|X|$ nodes in a graph G^* obtained from G' by taking p copies $G^1, \ldots, G^p$ of G' (each G^c has a node (v, c) for each v of G') and linking all nodes (v, c), (v, d) associated to the same v for each v and for all pairs c, d. An illustration is given in Fig. 7 for the graph G' of Fig. 6 with $p = 3$.

If (v, c) is in an independent set of G^*, it may be interpreted by saying that node v gets color c. So if we have now for each node v a set $\varphi(v)$ of feasible colors, we remove from G^* all nodes (v, c) such that $c \notin \varphi(v)$. Let $\tilde{G}$ be the remaining graph. So a node p-coloring exists for (G', φ) (the node coloring

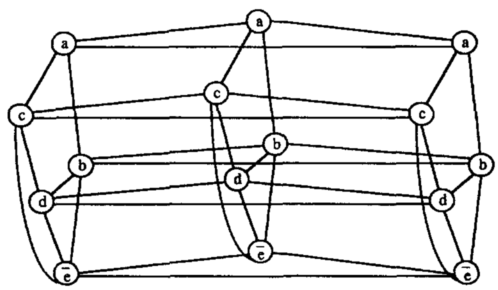

Fig. 7. The graph G^* associated to G' in Fig. 6 with $p = 3$.

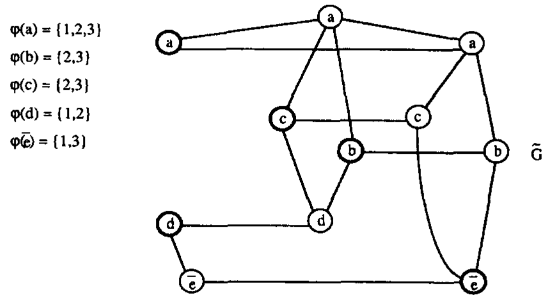

Fig. 8. The graph $\tilde{G}$ corresponding to the graph G' in Fig. 6 with sets of feasible colors for the nodes.

problem in G' with sets $\varphi(v)$ of feasible colors for each node v) if and only if an independent set of $|X|$ nodes exists in $\tilde{G}$. An illustration of $\tilde{G}$ is given in Fig. 8: thick nodes correspond to the independent set of $|X| = 5$ nodes: a and d get color 1, b and c color 2 while $\bar{e}$ gets color 3.

The proof of noncolorability may now be formulated as follows: let Γ be a set of pairs (K, c) where c is a color and K a clique of G. A node v is *covered* by Γ if for each color $c \in \varphi(v)$, there is in G at least one clique $K \ni v$ such that $(K, c) \in \Gamma$. Let $V(\Gamma)$ be the set of nodes of G covered by Γ.

Notice that if $K = \{u, v, \ldots, z\}$, we may replace (K, c) by $K_c = \{(u, c), (v, c), \ldots, (z, c)\}$ which is a clique in G^c (notice that nodes $v \in K$ such that $c \notin \varphi(v)$ do not generate a node (v, c) in K_c).

Consider a subset W of nodes of G and let $\tilde{W}$ be the set of all nodes (v, c) in $\tilde{G}$ such that $c \in \varphi(v)$, $v \in W$. $\tilde{W}$ generates a subgraph H of $\tilde{G}$. Suppose there is in H a collection Γ of cliques of the form (K, c) where K is a clique of G and which covers all nodes in H. The set $V(\Gamma)$ of nodes of G covered by Γ is clearly W.

Notice also that a feasible coloring for the nodes in W exists if and only if there is an independent set of $|W|$ nodes in H.

If $|\Gamma| < |V(\Gamma)| = |W|$ there is no feasible coloring for nodes in W, because the stability number $\alpha(H)$ of H and the clique covering number $\theta(H)$ of H (i.e., the minimum number of cliques needed to cover the nodes of H) satisfy $\alpha(H) \leq \theta(H) \leq |\Gamma| < |V(\Gamma)| = |W|$.

So there is no independent set in H with $|W|$ nodes. We may now reformulate Proposition 3.1; if T was a tree whose edges had to be colored, we may now consider a graph G which is the line graph of T; the nodes of G have to be colored.

If W is a subset of nodes of G, we denote by $H(W)$ the subgraph of $\tilde{G}$ generated by all nodes in $\tilde{W}$ (i.e., all nodes (v, c) in $\tilde{G}$ with $c \in \varphi(v)$, $v \in W$.

Proposition 3.1*. *Let G be the line graph of a tree; there exists a feasible node coloring for (G, φ) if and only if for every subset W of nodes of G*

$$\theta(H(\tilde{W})) \geq |W|.$$

Let us now examine when one can find in polynomial time a maximum independent set in a graph of the form of $\tilde{G}$. It will be the case in particular if $\tilde{G}$ is *perfect* (see [1]).

A graph G is *perfect* if for every induced subgraph $\hat{G}$, the chromatic number $\chi(\hat{G})$ (minimum number of colors needed to color the nodes of $\hat{G}$) is equal to the maximum size $w(\hat{G})$ of a clique in $\hat{G}$. Here G^* (and hence $\tilde{G}$) will be perfect if G' is a *block graph* , i.e., a graph where each block (maximal two-connected component) is a clique. So we can state:

Proposition 3.2 (see [8]). *If G' is a block graph, the timetabling problem with unavailabilities can be solved in polynomial time.*

A simple method based on dynamic programming (generalizing the one given after Proposition 3.1) can be devised.

All these solvable cases are very special and hence do not allow us to solve most of the real problems. However more solvable cases should be discovered by exploring other avenues.

One should at this stage mention some partial results about special unavailability requirements, i.e., the preassignment constraints.

It is known that the classteacher timetabling problem with preassignments is NP-complete even if the number p of periods is 3 [5].

The problem may be formulated in terms of edge coloring (in a bipartite multigraph): given some precolored edges in G, can one extend the precoloring to an edge p-coloring of G? Some cases of precoloring extensions are discussed in [2,13].

As mentioned above this problem is generally difficult. However we may consider special sets of precolored edges. Let us assume that the edges which are precolored form a bicolored cycle C (we may assume that its edges have been colored with colors 1 and 2).

When can one extend this to an edge p-coloring of G where p is the maximum degree $\Delta(G)$ of G? A *mouth* in G consists of three chains which all link two nodes x and y and have all distinct intermediate nodes. A *mouth* is *even* if the three chains have even length.

It has been shown that in a bipartite multigraph G containing no even mouth as a partial subgraph the bicoloring of any cycle can be extended to an edge $\Delta(G)$-coloring of G. The property holds whatever number of parallel edges are introduced between the nodes of G if and only if G has no even mouth [17].

In fact instead of graphs we consider sets of entries in the requirement matrix R; we call *configuration* a subset of entries in R; it corresponds to all bipartite multigraphs obtained by inserting any number of parallel edges in the chosen cells. So we may ask what are the configurations in R such that any preassignment (in the form of a bicolored cycle) can be extended to a complete timetable in p periods (where p is the maximum degree $\Delta(G)$ of G i.e. the maximum line sum in R, where a line designates a row or a column of R).

Proposition 3.3 (see [17]). *A configuration in R has the property that any cyclic preassignment can be extended to a complete timetable in p periods if and only if the corresponding graph contains no even mouth as a partial subgraph.*

4. Compactness requirements

In this section we shall briefly discuss another type of requirement where partial results are also formulated in terms of configurations. Quite often classes and teachers prefer to have so called *compact* timetables: they prefer to have their meetings grouped together as much as possible.

Consider the schedule of a teacher (or a class) in p periods; at each period the teacher is either active (i.e. involved in a meeting) or inactive (there is an idle period). A *block* is a maximal sequence of

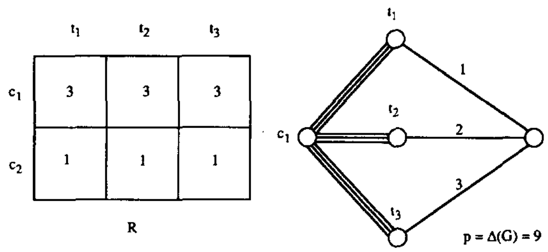

Fig. 9. A timetabling problem without compact solution.

active (or of inactive) periods in the timetable of a teacher (or of a class).

An individual timetable (of a teacher or of a class) will be called *compact* if it consists of at most three blocks. A timetable with one block occurs when the class (or the teacher) is always active (A) or always inactive (I).

Observe that a timetable with AIA is by definition compact; in such a case we consider that the timetable is repeated day after day (or week after week), so the set $\{1, \dots, p\}$ of periods may be considered as a cyclically ordered set and the timetable now may really look compact in the sense that the active periods are cyclically consecutive. We may observe that timetables which are compact for all classes and all teachers may not always exist. Fig. 9 shows a case where no compact timetable exists (edge coloring in G where at each node v the edges adjacent to v have colors which are cyclically consecutive): without loss of generality we may assume that c_2 meets t_i at period i ($i = 1, 2, 3$). For compactness reasons color 7 can only be given to an edge $[c_1, t_1]$ and the same holds for color 6 to some edge $[c_1, t_3]$. This forces colors 8, 9 for the remaining two edges $[c_1, t_1]$ and colors 4, 5 for the remaining two edges $[c_1, t_3]$.

So we have colors 1, 2, 3 for the three edges $[c_1, t_2]$ and this is impossible.

This means that the configuration consisting of the six entries of R does not have the property that it has a compact timetable in p periods whatever positive values r_{ij} are given to the entries (where $p = \Delta(G)$ is the maximum line sum in R). Along this line we may state:

Proposition 4.1 (see [12]). *For a configuration in R there exists a compact timetable in p periods whatever values r_{ij} are given to the entries chosen (where p is the maximum line sum in R) if and only if the associated bipartite graph has no mouth as a partial subgraph.*

The graphs in the above proposition are precisely the bipartite outerplanar graphs; they can be recognized and their edges can be "compactly" colored in polynomial time (see [12]).

5. Chromatic scheduling

Timetabling problems may be regarded as special cases of chromatic scheduling problems; these are scheduling problems where graph coloring models may provide solutions.

In general we have a collection of items K_i (lectures) to be scheduled. A set $\mathscr{R}$ of (renewable) resources $R_1, R_2, \dots, R_u$ are given together with the amount r_{it} of resource R_i available during period t.

For each item K_i we are given the subset $\mathscr{R}_i \subseteq \mathscr{R}$ of resources which are needed (we assume that one unit of each one of the resources in $\mathscr{R}_i$ is needed to process K_i). All these items K_i have a unit processing time.

An item (lecture) K_i may be scheduled at period t if at least one unit of each resource in $\mathscr{R}_i$ is available (and not used by other items at the same time).

As an example each teacher v and each class w is a resource; we have $r_{vt} \leq 1$, $r_{wt} \leq 1$ for each t. Classrooms (or classroom types) may also be resources. A model of hypergraph H may be constructed as follows: each item (lecture) is a node; for each resource R_i we introduce an edge R_i containing all nodes using this resource.

A schedule in p periods corresponds to a generalized node p-coloring of H. Each node gets some color in $\{1, \dots, p\}$; in each edge R_i there are at most r_{it} nodes of color t; this model generalizes the node coloring model in a graph given in Section 3 for unavailability constraints as can be seen easily.

In the special case where each item (lecture) requires one teacher t_j, one class c_i and a classroom (all classrooms are identical and r_t classrooms are available at period t), we may view the problem as a node p-coloring problem in a hypergraph with the additional requirement that at most r_t nodes have color t for each $t \leq p$.

In spite of the special structure of H (which makes as before the node coloring problem identical to an edge coloring problem in a bipartite multigraph), the problem is NP-complete [3].

A polynomially solvable case is obtained when the values $r_1, \ldots, r_p$ (ordered in such a way that $r_1 \geq r_2 \geq \ldots \geq r_p$) satisfy for some u $(2 \leq u \leq p)$ $r_1 - r_u \leq 1$, $r_{u+1} - r_p \leq 1$.

Other solvable cases are discussed in [3].

In general one has to take into account simultaneously unavailability constraints and classroom requirements. The problem may be formulated as a node p-coloring problem in a graph G where each node v has a set $\varphi(v)$ of feasible colors and it is required that at most r_t nodes in G have color t for each $t \leq p$. The problem is NP-complete in general as can be expected. However it is shown in [21] that when G is a collection of node disjoint cliques then the problem can be solved by network flow techniques.

In case the graph G is not a collection of node disjoint cliques, we may consider the simple situation where it is a chain. The problem may be interpreted in terms of scheduling as follows:

we are given a collection of n tasks v arranged in a row; they all have the same processing time, say one time unit. For each v, we are given the set $\varphi(v)$ of periods where v could be processed. For processing the tasks we have k robots and it is required that robots never process simultaneously adjacent tasks. Is there a schedule? This amounts to deciding whether there exists for this G a feasible node coloring (tasks are nodes) such that there are at most k nodes of each color.

Another related situation is when in a class–teacher timetabling problem classrooms have to be rented. We are given a cost k_t for each classroom used in period t. The problem is to find a timetable (i.e. an edge p-coloring $(M_1, \ldots, M_p)$ where each M_i is a matching in the bipartite multigraph $G = (\mathscr{C}, \mathscr{T}, E)$) such that its cost $\sum_{t=1}^{p} k_t |M_t|$ is minimum.

This problem is NP-complete in general even if $p = 3$; however we can state

Proposition 5.1 (see [19]). *If $G = (\mathscr{C}, \mathscr{T}, E)$ is a tree, one can find in polynomial time a timetable (corresponding to an edge p-coloring $(M_1, \ldots, M_p)$ with minimum cost $\sum_{t=1}^{p} k_t |M_t|$.*

The method consists in solving a sequence of assignment problems in a dynamic programming framework.

Again in this case, many more nontrivial solvable cases should be discovered; at this stage, all solvable cases are still far from covering a huge subdomain of the main timetabling problems occurring in practice.

The search for more solvable cases is however encouraged first because this may give ideas of original methods for approaching — most likely in a heuristic way — the variety of problems which are present in real schools. Other reasons will be given later.

6. Conclusions

As a conclusion, we shall briefly sketch a general procedure which has been successful in tackling many types of combinatorial optimization problems; its performance for general timetabling has also been extremely encouraging; it is the now well known tabu search procedure.

In the last years various authors have tried to adapt the tabu search procedure (TS) to timetabling (see [4,9,10]); in opposition to the previous models, this approach is able to handle all kinds of special requirements in a unified framework. We shall base the presentation on the method COSTA (Computing an Operational Schedule with a Tabu Algorithm) described in [4] and successfully applied to several schools in Switzerland.

Essentially the procedure consists in dividing the collection of requirements in *essential* and *relaxed* requirements. A timetable $\mathbb{T}$ is *feasible* if it satisfies the essential requirements; we introduce for each feasible timetable $\mathbb{T}$ in X (the set of feasible timetables) a measure $f(\mathbb{T})$ of violation of the relaxed constraints: the problem now consists in finding a timetable $\mathbb{T}$ in X for which $f(\mathbb{T})$ is minimum. Since $|X| < \infty$, this is a combinatorial optimization problem and the idea is natural to try an approach via TS.

Table 1
General tabu search

```
generate a feasible solution s ∈ X
s* := s (best solution so far)
nbiter := 0
bestiter := 0 (iteration where s* was found)
T := ∅ (tabu list)
Initialize A (aspiration function)
while (f(s) > f_min) and (nbiter − bestiter < nimax) do
        nbiter := nbiter + 1
        generate set V* ⊂ N(s) of solutions s_i = s ⊕ m_i
        with either m_i ∉ T or f(s_i) < A(F(s))
        choose best s' in V*
        update T and A
        if f(s') < f(s*) then s* := s'
                          bestiter := nbiter
        s := s'
```

A general description of TS can be found in [7,16]; the algorithm is sketched in Table 1. The process essentially moves from a solution (timetable) s to a best neighbour s' found in the neighborhood $N(s)$ of s; some moves from s to s' are forbidden (tabu) to prevent cycling; they are included in a tabu list which is periodically updated. However some tabu moves may be performed in spite of their tabu status if they are in some sense promising; this is measured by an aspiration function A.

The exploration process continues until a stopping criterion is met. The efficiency of TS depends highly upon the adjustment of some parameters but also upon the modeling of the problem. In our case the choice of essential and relaxed requirements is crucial. In the COSTA procedure, the essential constraints were the preassignments of lectures, the unavailabilities of teachers and classes, the spreading of lectures of the same course throughout the week, the consecutivity of periods for double lectures, the forbidden sequences of topics and finally the lunch break requirements.

All remaining requirements were considered as relaxed constraints and introduced into the objective function. These included the availability of rooms of special types, the feasibility of periods for specific topics, the overlap of lectures for a teacher (or a class), the geographical requirements (due to the time to move from a building to another one), the compactness for teachers, the preparation time for special rooms and the lunch balancing requirements (one half of the school has lunch at 12, the other half at 1 PM).

Applications to two real schools have shown that finding a feasible schedule is easy; most of the computation time is spent in trying to satisfy the relaxed constraints.

More details about the applications of TS to timetabling can be found in [4,10]. It is likely that such approaches will remain for some time the most efficient approach for handling problems including a wide collection of requirements. Combinations with other techniques like genetic procedures are also promising as far as one can judge from the first experiments.

However the solution of combinatorial problems related to timetabling is important and the need for solutions of more intricate combinatorial models is still essential. The reason is that these solution procedures will be repeatedly used in general methods (as TS or genetic algorithms) to help finding feasible solutions at each step or to show that none exists.

So much more research in combinatorial structures occurring in timetabling is required for identifying a large class of problems which may be solvable in polynomial time.

References

[1] C. Berge, *Graphes*, Gauthier-Villars, Paris (1983)
[2] M. Biró, M. Hujter, Zs. Tuza, Precoloring extension. I. Interval graphs, *Discrete Mathematics* 100 (1992) 267–279
[3] J. Blazewicz, D. de Werra, Some preemptive open shop scheduling problems with a renewable or a non renewable resource, *Discrete Applied Mathematics* 35 (1992) 205–219
[4] D. Costa, A tabu search algorithm for computing an operational timetable, *European Journal of Operational Research* 76 (1994) 98–110
[5] S. Even, A. Itai, A. Shamir, On the complexity of timetable and multicommodity flow problems, *SIAM Journal on Computing* 5 (1976) 691–703
[6] M.R. Garey, D.S. Johnson, *Computers and Intractability: a Guide to the Theory of NP-Completeness* (Freeman, New York, 1979)
[7] F. Glover, Tabu search, Part I, *ORSA Journal on Computing* 1 (1989) 190–206
[8] H. Gröflin, Feasible graph coloring and a generalization of perfect graphs, IAOR Report 199 (1992) University of Fribourg

[9] A. Hertz, V. Robert, Constructing a course schedule by solving a series of assignment type problems, ORWP 94/10, EPFL, November 1994
[10] A. Hertz, Tabu search for large scale timetabling problems, *European Journal of Operational Research* (1991) 39–47
[11] A.J.W. Hilton, School timetables, *Annals of Discrete Mathematics* 11 (1981) 177–188
[12] N.V.R. Mahadev, Ph. Solot, D. de Werra, The cyclic compact open shop scheduling problem, *Discrete Mathematics* 111 (1993) 361–366
[13] O. Marcotte, P.D. Seymour, Extending an edge-coloring, *J. of Graph Theory* 14 (1990) 565–573
[14] D. de Werra, Graphs, hypergraphs and timetabling, *Methods of Operations Research* 49 (1985) 201–213
[15] D. de Werra, An introduction to timetabling, *European Journal of Operational Research* 19 (1985) 151–162
[16] D. de Werra, A. Hertz, Tabu search techniques: A tutorial and an application to neural networks, *OR Spektrum* 11 (1989) 131–141
[17] D. de Werra, N.V.R. Mahadev, U.N. Peled, Edge chromatic scheduling with simultaneity constraints, *SIAM Journal on Discrete Mathematics* 6 (1993) 631–641
[18] D. de Werra, A.J. Hoffman, N.V.R. Mahadev, U.N. Peled, Restrictions and preassignments in preemptive open shop scheduling, *Discrete Applied Mathematics* 68 (1996) 169–188.
[19] D. de Werra, N.V.R. Mahadev, Preassignment requirements in chromatic scheduling, to appear in *Discrete Applied Mathematics* (1997).
[20] D. de Werra, Some combinatorial models for course scheduling, ORWP 95/08, Swiss Federal Institute of Technology, Lausanne (1995)
[21] D. de Werra, Restricted coloring models for timetabling, ORWP 95/09, Swiss Federal Institute of Technology, Lausanne (1995) (submitted for publication)