Multiple-level Models: an application to employee timetabling

Tahar Zemmouri[1,2], Peter Chan[1], Michael Hiroux[1,2], Georges Weil[1,2]

[1] Equitime S.A., 4 Ave. de l'Obiou,
38705 La Tronche, France,
http://www.equitime.fr
[2] Laboratory TIMC-CNRS-IMAG, University Joseph Fourier,
38705 La Tronche, France

Abstract. We present a novel method of modelling problems where several levels of time granularity coexist, and are handled simultaneously. This situation arises in employee timetabling in France, where legal laws restrict work and rest durations per day, per week, per month, per trimester and per year (applicable from Jan. 2001). It is no longer possible to design a system to work at a single level of time aggregation and yet produce legal timetables.
Current State-of-the-Art systems could handle only 2 levels (daily and weekly) with very complex models; we need at least 3 levels to produce legal timetables. Compared to typical solvers with only one model, enormous performance gains are possible because we can instantiate only the appropriate number of solvers at each level.
We investigate the simultaneous use of different models within the Constraint Satisfaction Framework. It allows expressing different classes of constraints and hence produces timetables that exhibit many *innovative* features, from the daily level up to a year or more (such as career planning).

1 Introduction

Employee timetabling is at the heart of every human enterprise. Their success depends on making good and timely use of human resources: the right man at the right place at the right time – *all the time*. Unlike machines, there are many rules and regulations on employees' work and rest. In France, there are legal constraints at the daily, weekly, monthly, trimester and even yearly levels. In this paragraph, we shall give an overview of the problem and review the state-of-the-art. Although the earliest work in this area dates back some 50 years, this problem is far from being solved. § 2 sets the ground work in presenting models at the daily, monthly and yearly levels. Multiple-level models will be discussed in §3 and we compare these two ways of modelling. We conclude in § 4.

1.1 Current research in employee timetabling

Employee timetabling consists of assigning work to employees so as to cover provisional workload over each period of the timetabling horizon. This has been expressed since 1954 by Danzig [12] as a set-covering problem over the daily horizon:

$$\text{Min } z = \sum_{j \in \textbf{Shifts}} C_j * X_j \quad (\mathbf{1})$$

$$\text{Such that } \sum_{j \in \textbf{Shifts}} A_{ij} * * X_j \leq B_i,\ i \in \text{Employees} \quad (\mathbf{2})$$
$$X_j \geq 0, \text{ integer}$$

Where

n=number of periods to be covered in the planning horizon,
m= number of admissible shifts
C_j = cost of an employee affected to the shift j
A_{ij} = 1 if the period i is covered by the shift j, 0 otherwise
B_i = number of employees required for period i
X_j = number of employees affected to the shift j

Since then, there was continuing research work in solving this problem and extended variants using Linear Programming (LP) methods. Explicit shift representations, such as table A_{ij}, gave way to implicit formulations (see [20], [3], [4]), which reduced the complexity with a smaller number of variables. The first implicit model is due to Moondra [20] who also handled extra short breaks. Other extensions are proposed to cover practical cases such as different start times [3] and meal breaks [4].

With such models, current state-of-the-art applications handling periods of 15 minutes achieved horizons of up to one week; the total number of feasible shift models reach in hundreds of thousands and even millions. Using the *starting-time band* method (see [4], [15]), the resulting LP models were able to reduce the number of variables to a few thousand. The problem was then solvable on (powerful) personal computers with state-of-the-art LP solvers.

Multi-skilled employee timetabling such as that addressed in [5] are becoming an important part of the work environment, but the number of variables diverged.

Enterprises' need is to produce schedules that cover workload with as little over capacity as possible. This need is present at all time levels and at all times (daily, weekly, monthly and yearly). Their task is complex because there are laws and regulations that define the maximum work duration per day, per week and per month. Since January 2001 in France, the limit on the maximum work duration *per year* became legal. In addition, there are limits on the minimum rest duration between two work spells (meal break, daily rest, week-end, annual leave).

Indeed, annual timetables have been proposed as early as 1982 (see [10]) in order to achieve flexibility in various sectors (e.g. [11], [14] and [17]). The French law clearly defined the legal limits and spurred further research (see [9], [21], [22]). Another LP proposal [6] handled the weekly horizon by combining seven daily shift-timetabling models in a network flow framework, which handles the weekly requirements. It is more flexible than starting time band constraint as it is applicable to consecutive days of the tour.

[21] used a technique iteratively combining LP techniques for operator timetabling and CP for tour timetabling. The first stage provided a set of daily shifts that cover estimated demand for each day of the horizon. They are used by the CP stage to construct weekly or monthly tours. There is no timetabling-related interaction between the two stages. Frequently to meet the weekly requirements, it is necessary to modify manually the daily shifts or add additional agents.

1.2 Our Proposition and Motivation

Our research is aimed at creating software solutions for building realistic employee timetables that respect **all** legal constraints. Up to now, models have been created using variables of the same time aggregation (or *single-level models*). To cover two levels, extensive modelling efforts were deployed to reduce complexity by exploiting underlying problem properties (see [15, 4]). It is difficult to cover even more levels with such methods and it is not clear how the constraints at different levels can be respected if they are not represented in a common model.

To solve the problem of legal limits at different levels of detail, we propose a novel type of model allowing for *simultaneous* application of several models within the same resolution mechanism. In short, the *multi-level model* uses variables representing daily, monthly and yearly time aggregations. The different levels enabled users to express rules or preferences at the appropriate time granularity. This work generalizes ideas initially proposed by the authors in [8], where we showed how daily cycles could be transformed into weekly cycles so as to be processed by the same mechanism for applying them. The same rules can be translated to different levels.

To illustrate this *modeling concept*, we use the Constraint Programming (CP) framework (e.g. [23]) to allow the different models to work together, propagating deductions at one level to consequences at another. In many practical situations when the constraints are too restrictive and admit no solutions, it is necessary to relax progressively the problem. Practically speaking, such a system would be simpler to build with models using explicit variables on different levels, rather than one using a single implicit model such as that proposed in [4].

In contrast to current LP models, we propose models that consider individual employees, so that their individual skills, preferences and planning history can be taken into account to produce balanced assignments. Although the number of employees handled by our models would be smaller, we expect the quality of schedules to be very high. In addition, the large horizon can take into account varying needs that are seasonal, as well as changing manpower supply (i.e. turnover).

Our approach can be considered as *full – CP*, designed to cover at least one year of detailed planning with three levels where legally binding constraints have to be respected.

We have not explored the possibility of using other *resolution methods,* such as Mixed Integer LP (MILP), to implement the *modeling concept* presented in this paper.

2 Single Level Models (SLM)

We shall present the Constraint Satisfaction Problem and several single-level models separately in comparable terms. In this paper, we discuss in detail the following: daily, weekly/monthly and yearly levels, based on the constraints we want to solve, which are basically the legal restrictions.

Legal constraints in employee timetabling are of two basic categories: upper bounds on the work duration and lower bounds on rest duration between works spells.

2.1 Constraint Satisfaction Problem

Definition: A Constraint Satisfaction Problem (**CSP**, see [18], [19], [23]) is defined by the triplet (X, D, C):

X = $\{x_1, \ldots, x_n\}$ is a finite non-empty set of n variables (so called finite-domain variables)

D = $\{d_1, \ldots, d_n\}$ is a finite non-empty set of domains of values for the variables X, d_i being the domain of variable x_i.

C = $\{c_1, \ldots, c_n\}$ is a finite non-empty set of constraints. Each constraint $c_i(x_{i1}, \ldots, x_{ik})$ defined over the subset of variables $\{x_{i1}, \ldots, x_{ik}\} \subseteq X$, represents in extension, the subset of the Cartesian product $x_{i1} \otimes \ldots \otimes x_{ik}$. Each element of this subset is a set $\{n_{i1}, \ldots, n_{ik}\}$ of values ($x_1 = n_{i1}$) that satisfy the constraint c_i.

The **CSP** is indeed very general and many real-world problems can be expressed in these terms. It was shown to be NP-complete (see [16], [18]).

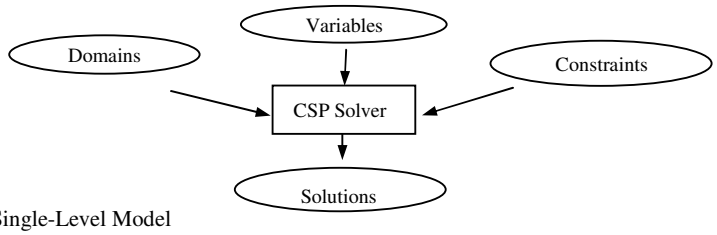

Fig. 1. Single-Level Model

2.2 Day level Model

The decision variable at the daily level is the assignment of an employee during period p of day d to work at a given skill s or to rest. Defined per period typically 15 minutes, the daily workload N_D constraint is covered when

$$\forall e \in \textbf{Employees}, \forall p \in \textbf{Periods}\ \textbf{X}(e, p) \subseteq \textbf{Skills} \cup \{\textbf{Rest}\} \quad (3)$$

$$\forall p \in \textbf{Periods}, \mathbf{N_D}(p, s) \leq |\{\ \textbf{X}(e, p) = s, \forall e \in \textbf{Employees}\ \}| \quad (4)$$

Where **Periods**, **Days**, **Employees, Skills** are given sets of objects, |S| being the cardinality of the set S and $\mathbf{N_D}$ is the need in number of persons per period on the given day and per skill.

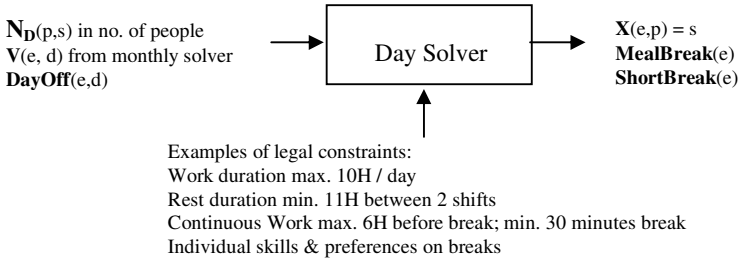

Fig. 2. Day-level Model

The needs per period and per skill in number of employees are an input data obtained by statistics over a given horizon. The solver also has the day-off and the standard shift of each employee, which are the variables of the monthly solver. The daily assignment per period/day per employee is obtained by refining the hours of the employees' standard shift, according to needs of the day, individual preferences and equity in planning.

Definition: A standard-shift is a general set of working hours such as morning shift (8 am to 4 pm), evening shift (4 pm to midnight) and night shift (midnight to 8 am). They are defined when the company's work organization is created.

Definition: A shift is a set of working hours for an employee on one day (e.g. 8.15 am to 12 noon, 12.30 pm to 6 pm). We distinguish the standard shift that is generic (independent of the employee and the day) and can be generated by the daily level model, from the specific shift that is a day-to-day affair for a given employee.

2.3 Month-level Model

The monthly level variables take values in the Cartesian product **Skills** X **Shifts**, the set of standard shifts; i.e. we assign an employee to a skill and a standard shift. The monthly needs N_M are expressed in terms of number of persons per day, skill and standard shift. Employees are assigned shifts based on their skills and availabilities.

$$V(e, d) \subseteq \textbf{Skills} \times \textbf{Shifts} \quad (\mathbf{5})$$

V stands for "vacation", the French word for shift

$$\forall d \in \textbf{Days}, \mathbf{N_M}(d, s, v) \leq |\{ V(e, d) = (s, v), \forall e \in \textbf{Employees} \}| \quad (\mathbf{6})$$

The set contains the couples (Skill, Shift)

This model was implemented by the authors in [7] and [8].

Table 1. Models at different levels, with sample values.

Different levels of time aggregation used in employee timetabling. To build an MLM, you will have to choose the required levels and models. Since the week model is similar to the month model, it will not be further discussed here.

	Day Model	Week Model	Month Model	Year Model
Basic Variables	X(Employee, Period) = Skill or rest, assigned to the employee for that period, on that day. E.g. Casher, Hot-Line, Cooking, etc.	W(Employee, Week) = (Skill x Standard shift) assigned to employee on that week. E.g. (Afternoon, Cooking), (Morning, Casher)	V(Employee, Day) = (Skill x Standard shift) assigned to employee on that day. E.g. (Morning, Counter), (Evening, Hot-Line)	Y(Employee, Skill, Week) = number of hours worked by the employee during that week at that skill.
Horizon	1 to 2 days	1 to 2 weeks	28 to 42 days	14 to 16 months
Period	10 – 15 minutes	1 day	1 day	1 week
Needs	N_D(Period, Skill) in number of employees	N_W (Week, Skill, Std. Shift) in no. of employees	N_M (Day, Skill, Std. Shift) in number of employees	N_A (Week, Skill) in employee-hours
Work duration Constraint				
Max.	Maximum 10 hours per day	Max. 43 hours per week	Max. 20 working days	Max. 1600 hours / year. Max. 38 hours / week over 12 weeks
Min. or Average	Minimum 4 hours per day		Minimum 8 working days per month (e.g. for part-timers)	Average of 35 hours / week over 1 year
Rest duration Constraint				
Long break	Meals (45 minutes – 1 hour), after 3 to 6 hours of work	Two (consecutive) day-off, after 3 to 6 continuous working days	Every 2 weeks, at least 2 consecutive day-off including 1 Sunday	Individual holidays (2 or 3 weeks)
Short break	Pause (10 – 15 min.) after max. 2 hours of duty	One single day-off, after at least 2 days of continuous work	N.A.	Week off

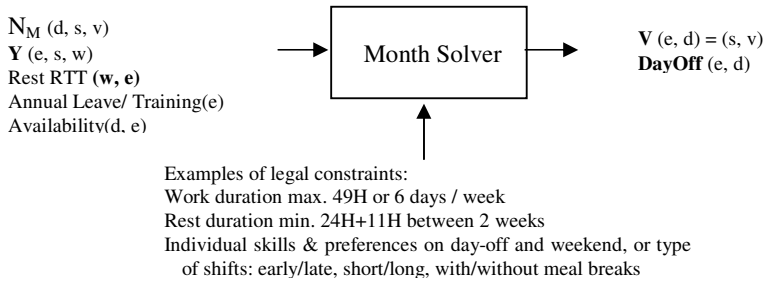

Fig. 3. Month-level Model

In many continuous work institutions, rest constraints may be defined by a minimum of A weekends off out of B weeks.

2.4 Year-level Model

At the yearly level, the variable is the number of hours worked by the employee e, for the week w at a given skill s. They are assigned so as to satisfy annual needs N_A expressed per week and per skill.

$$Y\ (e, s, w) \subseteq \mathbf{N} \tag{7}$$

$$\forall\ w \in \mathbf{Weeks},\quad N_A\ (w, s)\ \leq\ \sum_{e \in \mathbf{Employees}} Y(e, s, w) \tag{8}$$

The single-skill version of this model is fairly common in the distribution sector (for example, see [9]).

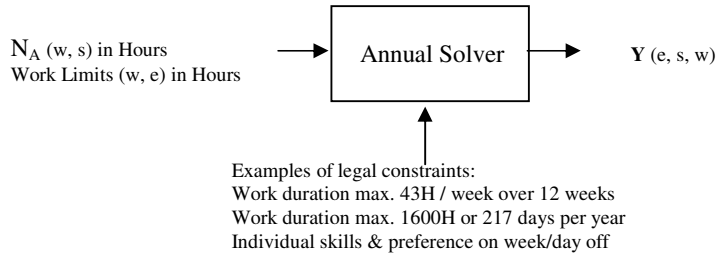

Fig. 4. Annual level Model

At the yearly level, French regulations state a maximum of 1600 hours of work, or an equivalent of 217 days (7,37 hrs per day or 36,68 hrs per week) for certain personnel categories. There is no legal obligation on rest (interpreted as annual leave). In some industries, annual leave of 3 to 4 weeks are allotted during summer and/or winter holidays, with short leave (1 week) around Christmas, All Saints' Day, etc.

In absence of provisional planning, close to the end of the year, key individuals would have reached or exceeded their annual work limit, resulting in heavy bonus being paid in full. On the other hand, individuals having worked below their annual work limit are required to be paid in full, by law. Such financial considerations often justify the use of the annual solver.

2.5 Single Level Solvers

Thus formulated as CSP, each model can be solved using backtracking, coupled with constraint consistency checking algorithms [16]. Although it is not the main contribution of this paper, we include some indications in this paragraph for completeness.

At the daily level, each variable X(e, p) has a finite set of possible values equal to the set of skills to which the employee may be assigned during the period, including rest. As proposed in our paper [24], we confound the different types of rest: pause, meal break, daily rest, weekly day-off, or shift since they all do not contribute towards covering an expressed requirement. A solution is found when all variables are each assigned a value in such a way that all constraints are satisfied at once. Solutions can be found by searching though all possible assignments of values to variables. So as to reduce search, whenever the constraint network is updated various consistency techniques and filtering algorithms have been proposed to remove inconsistent values from these sets of possible values [13].

For example, when the variable for employee e, for period p on day d, is assigned to skill s, and if work involving skill s' cannot follow s (for technical or practical reasons) then s' must be removed from the set of possible values of the neighboring variables. Furthermore, the workload constraint must be triggered to verify if the coverage of the other skills s" of this employee is possible for that period.

The usual heuristic for solving each SLM is to choose the variable with the least remaining possibilities (the "fail-first principle" [23]). At the daily level, we would choose the period p where there is the largest difference between the number of available personnel and the needs, for all skills, say s. This period is the 'most' difficult to solve. Among the candidates with this skill, one would choose the employee e that is single-skilled in s. In the case of ties, the various preferences both from the employee and employers' point of view may be taken into account.

At any one level, timetabling multiple-skilled employees requires new heuristics. It is of course necessary to assign single-skilled employees without hesitation. To ensure that requirements for all skills be covered, it is necessary to pick the right skill for the right employee. A typical heuristic to select an employee to cover a work is to choose the employee whose other skills are least needed during the period. Otherwise, the employee should be assigned to this skill s which is most needed.

At the annual level, the period is one week. Every week, the total number of hours per skill over all employees must match specified needs per skill. A heuristic method can be used to solve the annual SLM, for example by focusing on weeks where the excess available employee-hours over total needs are least.

3. Multiple-Level Models (MLM)

In order to produce legal schedules for managers, we propose using three or more models within a multiple-level solver. The Constraint Satisfaction Framework provides the glue between individual SLM. Decisions made at one level may be propagated or translated into effects on the other levels. Constraints are seen as software triggers or demons, implemented as object-oriented update methods at each level.

As suggested in Table 1., we can build an MLM with any combination of levels. In this paper, we further explore only the day-month-year combination. We assume a simple mechanism to map the periods in each day between the day and month models and to map the days between the month and annual models.

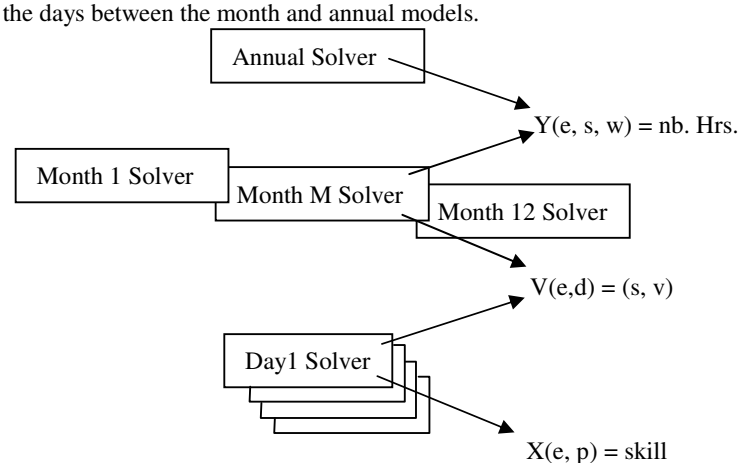

Fig. 5. To produce a schedule for 1 month, involving constraints at the daily, monthly and yearly levels, different models are needed in different numbers. Finite-domain variables establish the link between two different models

3.1 Cross-level propagation

Example: Consider a pizza outlet, open from 8 am to 10 pm daily, where Skills = { Casher, Delivery, Cuisine, Management}. Each of the first five employees e_1, …, e_5 are capable of all skills but Management. Suppose that the employee e_5 is assigned to work at skill Cuisine during period p:

$$X(e_5, p) = \text{Cuisine} \quad (\mathbf{9})$$

At the monthly level, the set of possible shifts for that day are immediately restricted to those shifts that contain period p, including the rest shift.

$$V(e_5, d) = \{ v \in \text{Shifts} \mid A_{pv} = 1 \} \quad (\mathbf{10})$$

There may be secondary deductions if the successive working days approach the limit for the week. There would be deduction at the yearly level if the total work assignment for the week exceeds 38 hours (recall the constraint on the number of weeks exceeding 38 hours over a 12 week horizon).

Inversely, when the set of possible shifts for a day d and employee e_5 is modified, it can be deduced that the non-working periods common to remaining possible shifts must be assigned to rest.

$$\forall\ v \in V(e_5, d), \text{let } P = \{p \mid p \in \text{Periods}, A_{pv} = 0\} \quad \textbf{(11)}$$
$$\text{then } X(e_5, p) = \text{Rest}, \forall\ p \in P$$

In this way, decisions made at higher levels of detail such as yearly/monthly, can be mapped into a great number of decisions at lower levels, e.g. daily/monthly.

Some more examples of cross-level constraint propagation are as follows:

- After having worked a total of 35 hours during several (consecutive or not) days, and if none of the compatible shifts of an employee have duration of 4 hours (39-35), then the rest of the week is assigned day-off. Otherwise the limit of 39 hours per week cannot be attained.
- When the shift of a day is defined, the constraint on 6 continuous working days can be used to deduce that the preceding and/or succeeding days are off.
- When the basic shift variable of the employee e, is assigned to the day-off shift, all the basic variables X during the day are assigned to rest.
- When the number of 'high' consecutive weeks (more than 38 working hours) is exceeded, the possible set of shifts on the remaining days of the week is restricted to those with a small number of work hours.

3.2 Partial Model Instantiation

Figure 5 illustrates the various models to be instantiated in a typical run. So as to take into account previous assignments, the yearly horizon covers essentially the past, but still need to cover the civil year. Consequently, it is necessary to cover typically 16-18 month horizon. However, it is not necessary to instantiate every models to the 15 – minute detail level. Our idea is *partial instantiation*, i.e. instantiate the appropriate level of detail when needed.

In a planning system covering the monthly horizon at 15-minute intervals, the total available search space is astronomical. With the additional time structures offered by each level, it is possible to express employees' preferences in an MLM, whereas the monolithic SLM covering the same horizon is not sufficiently structured to allow such expressions. Consequently, heuristics based on the MLM can take these expressions into account, using them to reduce searching solutions into non-preferable areas. When they lead to conflicts with hard constraints or workload coverage elsewhere over the time horizon, the algorithm will ignore these preferences.

Consider the timetabling of the pizza outlet over one month, with daily, weekly, monthly and yearly constraints. So as to capture yearly, monthly and daily constraints, we would have to create a large monolithic horizon of 365 days * 144 = 52560 peri-

ods of 10 minutes. Each period would correspond to one assignment or domain variable, per employee. With 4 possible values (4 skills) per variable, the theoretical complexity comes to $4^{52560} \sim 10^{31500}$.

Instead, we would have created one yearly model, 12 monthly models and 31 daily models to cover the month concerned. Each daily model contains 144 variables for the 10-minute periods. This corresponds to 31*144 = 4464+12+1 = 4477 domain variables. The corresponding theoretical complexity is $4^{4477}. \sim 10^{2700}$. The monthly model contains 31 variables each taking 8 values (2 shifts x 4 skills) $8^{31} \sim 10^{27}$. The yearly model contains 12 variables each taking values (from 0 to 40) $12^{41} \sim 1.7 \times 10^{44}$. The overall complexity is $10^{2700} \times (10^{27})^{12} \times 1.7 \times 10^{44} \sim 10^{3068}$. This is very much less than that in the corresponding SLM.

3.3 Solving the MLM

From the resolution point of view, an MLM is made up of several SLM, each with its own variables, domains and constraints. There are additional constraints that tie variables from different models, as shown by the cross-level propagation examples in § 3.1. In this paragraph, we propose various methods to solve the MLM, based on Constraint Programming methods.

Global constraints could be used to solve for example cumulative constraints on variables at the same time level. Currently available global constraints are based on the SLM approach and cannot be used directly in the MLM.

3.3.1 Search algorithms

To build the search algorithm for the MLM, one can exploit the new structures and possibilities offered by navigating through the different models. Variable selection requires more diligence because they represent different aggregations. Each high time-level variable should have more priority because it corresponds to a large number of lower time-level variables.

The simplest solver to solve the MLM would be a single backtracking algorithm, applying the "fail-first principle" successively through the various levels. The annual solver would choose the most constrained week where there are the most needs compared to availabilities, for example. An assignment of weekly working hours for an employee with the most rare skill would trigger the monthly solver to choose the most constrained day and deduce shifts. Finally the daily solver for that day would choose the most constrained period to satisfy. The daily solver would continue to instantiate assignments for that employee already present, to periods with the largest needs, the possible shifts for that day would be reduced further until the variable V(employee, day) gets instantiated. In turn, this would trigger other monthly solver constraints on neighboring shifts. In the process, daily level variables can become very highly restricted (e.g. < 3 possible values). They would then be selected for enumeration.

3.3.2 Heuristics

It is clear that the MLM has a huge search space. In order to produce good schedules, the MLM can take into consideration *employer or employee preferences* that could not fit into the limited horizon or detail of typical monolithic models.

A general *employer preference* is to produce timetables that are balanced with regards to unpopular assignments such as assignments on public holidays (if the center is open for business), early morning work following day-off (including weekend, public holidays and annual leave), or late evening work preceding the same day-off. Of course, such preferences will again have to be forgone if hard constraints on total work assignment or rest are not respected in another part of the horizon. Another management preference would be to encourage tutor-student relationships, having sufficient numbers of experienced staff at certain days or times.

In fact, the large horizon of the annual solver would allow the system to model slow-changing factors such as seasonal needs and availabilities due to employee training and turnover.

Employee preferences at the day level include:

- Some part time employees would prefer to work in the evenings, while others in the morning
- Some employees would prefer to work long days (and lesser number of days per week). Others prefer shorter shifts to attend night classes, or to fetch kids from kindergarten.
- Some would prefer to work without lunch breaks; others would prefer long breaks (to allow for shopping or attend health classes, etc.).
- Some would prefer to work regular hours (9 to 5), while others are motivated to work irregular hours with bonus.

At the weekly/monthly level, individual preferences can be expressed for working various days of the week or for preferred off-days. At the yearly level, when union rules make it necessary to define off-weeks, employees may express preferences.

3.4 Comparing SLM and MLM

Proposing new types of models (not just new models) does not change the resolution mechanism *per sec*. It has an interest only when the resolution is made more efficient, such as that achieved by implicit modelling of shifts compared to explicit ones. In a similar situation, a multi-level model can be solved with the same mechanism as a single-level. Current engineering practice suggests precisely defining the horizons and limits of a problem we analyze. Hence we always end up with monolithic models and this is the case today of timetabling models. Our MLM proposition attempts to break out of this mold, and simultaneously use different models (at different time horizons), so as to be able to represent (and solve) legal constraints at different time levels.

The main tangible result is that MLM allows using the smallest number of variables (albeit, of different natures), as compared to the monolithic SLM. MLM can instantiate an appropriate number of models with the right kinds of variables, whereas SLM has to instantiate all its homogeneous variables.

During its resolution, we will see that models will behave in a non-sequential way, allowing collaborative problem solving deduction by the various models. Deductions at higher grain (yearly or monthly) can be mapped into numerous deductions at lower grain (the monthly or daily levels). Hence higher grain deductions should be favored over lower grain ones within the system.

The MLM does have advantages that the resolution mechanism can exploit. When the resulting model is simple, heuristics can be built that exploit the special structures for example when the problem is over constraint and some kind of relaxation is needed to obtain feasible solutions.

As remarked in § 1.2, since this paper concerns *modeling concepts* and not *resolution methods*, we do not report computation times. Indeed, the multi-level *concept* could well be implemented with other *resolution methods* (such as MILP).

4. Conclusions

We have presented a type of model designed to span several levels of detail, where most algorithms handle at most 2 levels. Covering a large yet detailed time horizon, multiple-level models can take into consideration legal constraints, both employee and employer preferences and past assignments that are normally out-of-scope in single-level models. Decisions made at higher levels can be mapped into many decisions at lower levels of detail and hence reduce overall search complexity.

With more circumspection, our approach could have some effects on the practice of designing models. The first usual step in problem solving is to analyze the granularity of the problem at hand. Our paper shows that this step is perhaps erroneous, considering all the constraints of the problem, because we are then limited by the constraints available at this granularity. The real-world problem may have other constraints at other levels of granularity and are not *visible* and cannot be taken into account by our analysis.

Future work may involve additional modelling of slow-changing factors such as seasonal needs and availabilities due to employee turnover. With an extension to four levels, we would model career aspects; involving training that changes employees' grades over a 3-year horizon. With such a rich model, the MLM would be to take into account a large number of parameters and potentially produce timetables as well as human planners and managers.

References

1 T. Aykin, "Optimal Shift Timetabling with Multiple Break Windows", Management Science Vol. 42, pp. 591-602, 1996.

2 C. Azmat, "A Model for Annualized Work Time", 14th Mini Euro Conference on Human Centered Processes, Luxemburg, 2003.

3 S.E. Bechtold, and L.W. Jacobs, "Implicit Modeling of Flexible Break Assignments in Optimal Shift Timetabling", Management Science Vol. 36, pp. 1339-1351, 1990.

4 M.J. Brusco and L.W. Jacobs, “Optimal Models for Meal-Break and Start-Time Flexibility in Continuous Tour Timetabling”, Management Science Vol. 46, pp. 1630-1641, 2000.
5 X. Cai and K.N. Li, “A genetic algorithm for timetabling staff of mixed skills under multi-criteria”, European Journal of Operational Research 125 (2000): 359 – 369.
6 T. Cezik, O. Günlük and H. Luss, “An integer programming model for the weekly tour timetabling problem”, Naval Research Logistics **48** (7): 607-624, October 2001.
7 P. Chan , K. Heus and G. Weil, “Nurse Timetabling with Global Constraints in CHIP: Gymnaste”, International Conference on Practical Applications of Constraint Technology. London UK. 1998.
8 P. Chan and G. Weil, “Cyclical Staff Timetabling using Constraint Logic Programming”, Proceedings of the Third International Conference Practice and Theory of Automated Timetabling, PATAT 2000, also appeared in LNCS 2079, Pp. 159-175.
9 A. Corominas, A. Lusa, and R. Pastor, “Using MILP to plan annualized hours”, Journal of the Operational Research Society, **53** (10): 1101-1108, 2002.
10 D. Clutterbuck, “After flexible hours, now its flexiyears”, International Journal of Management, 37(3): 31-36, 1982.
11 P. Curran, “Annual hours bring productivity to Spicers”, Management Services, July 32-33, 1992.
12 G.B. Dantzig, “A Comment on Edie’s Traffic Delays at Toll Booths”, Operational Research Vol. 2, pp. 339-341, 1954.
13 R.M. Haralick and G.L. Elliot, “Increasing tree search efficiency for constraint satisfaction problems”, Artificial Intelligence Vol. **14**, pp. 263-314, 1980.
14 Hung R., “Scheduling a workforce under annualized hours”, International Journal of Production Research, **35**(11) : 2419 – 2427, 1999.
15 Jarrah A.I.Z., Jonathan F.B. and Anura H.S., “Solving Large-scale Tour Timetabling Problems”, Management Science, **40**(9): 1124 – 1144, 1994.
16 V. Kumar, “Algorithms for Constraint-Satisfaction Problems: A survey”, Artificial Intelligence Magazine **13**(1): 32 – 44, 1992.
17 P. Lynch, “Annual Hours: an idea whose time has come”, Personnel Management, pp. 46-50, Nov. 1995.
18 A.K., Mackworth, “Consistency in Networks of Relations”, Artificial Intelligence Vol. 8, pp. 99-118, 1977.
19 U. Montanari, “Networks of constraints: Fundamental properties and applications to picture processing”, Information Science, **7**(2): 95-132, 1974.
20 S.L. Moondra, “An LP model for work force timetabling in banks”, Journal of Bank Resources, Vol. 7, pp. 299-301, 1976.
21 A. Partouche, “Planification d’horaires de travail”, Thesis, University of Paris-Dauphine, France, 1998.
22 B. Rottembourg et al. “Annualisation du temps de travail en centre d’appels: tomographie, flots et chemins”, Journées Francophone de la Recherche Opérationnelle, Conservatoire Nationale des Arts et des Métiers, Paris, 29 juin 2001.
23 E. Tsang, Foundations of Constraint Satisfaction, Academic Press, London 1995.
24 G. Weil, K. Heus, “Nurse Timetabling: solving a hard problem using Constraint Programming”, In CP’95 Workshop, Studying and Solving Really Hard Problems, Cassis, France, 1995.