Optimization of Shifts and On-Call Coverage of Cardiologists Working in a Hospital Complex Structure by using Free Software

Eugenio Cervesato[1], Giovanni Righini[2], Gian L Rellini[1], Matteo Cassin[1], Rita Piazza[1], Gian L Nicolosi[1]

[1]Cardiology Dept.-ARC , 'Santa Maria degli Angeli' Hospital, Pordenone, Italy
[2]University of Milan, Department of Computer Science, Crema, Italy

Abstract

The organization of shifts and on-call coverage of Cardiologists working in a hospital complex structure must ensure the availability 24h, full year, complying with laws and contractual, social and professional obligations. It faces a 'staff scheduling' problem.

The team consists of 16 Cardiologists to ensure daytime and nighttime guards, a nighttime on-call service, on-call service for full day on weekends and public holidays and a 24h hemodynamic on-call. Special situations and exceptions are taken into account.

An original software was developed in Java to input data, generate the mathematical model and read back the solution, relying on a open SQL database. The solution is obtained by GUSEK, an open tool to the linear programming solver GLPK minimizing the sum of the deviations from individual quotas.

The system generates the shifts monthly. Typical computation time is few minutes to ensure that a solution exists and few hours to obtain a highly optimized solution. The system is in use for 22 months. It provides equal assignment of Cardiologists to each type of shift with uniform like distribution, avoiding immediate repetition of the same shift. The software is open source available for download at: www.arc.fvg.it.

1. Introduction

The organization of shifts and on-call coverage of Cardiologists working in a hospital complex structure must ensure the availability 24h, full year, complying with laws and contractual, social and professional obligations. Staff scheduling problems have been largely studied since decades, such as nurse scheduling and recently for medical doctors [1]. Manual planning is very difficult, time consuming and error-prone process, and its results are likely to be put under discussion by the personnel. To overcome it, an automation project was started through an Hospital / University collaboration that led to the development of an automatic planner for scheduling Cardiologists in an Italian hospital.

2. Materials and methods

The team consists of 16 Cardiologists to ensure daytime and nighttime guards, a nighttime on-call service, an on-call service for full day on weekends and public holidays and a 24h hemodynamic on-call coverage (this assigned to the same specialist weekly) (Table 1). For weekends and public holidays, it was agreed to combine guards and on-call coverage in a way to reduce the number of Cardiologists involved. Three dummy shifts had been later added to guarantee attendance at work of some area specialists (Table 1).

Table 1. Definition of shifts and on-call availability

DIU	daytime guard 8-20 – only working days
NOT	nighttime guard 20-8 – only working days
REPD	daytime on-call service 8-20 – only weekends and public holidays
REPN	nighttime on-call service 20-8 – only working days
WE1	daytime on-call service 8-20 Saturday + nighttime guard 20-8 Saturday
WE2	daytime guard 8-20 Saturday
WE3	daytime on-call service 8-20 + nighttime guard 20-8 Sunday or public holidays
WE4	nighttime on-call service 20-8 Saturday + daytime guard 8-20 Sunday + nighttime guard 20-8 Sunday
WE5	daytime guard 8-20 public holidays + nighttime on-call service 20-8
RE1	hemodynamic on-call service (weekly from Saturday through Friday)
RE2	hemodynamic shift – only working days
HIS	electrophysiologic shift – only working days
ECO	echocardiographic shift – only working days

Special situations and exceptions are taken into account: one Cardiologist performs 30 daytime guards per year (on Tuesdays or Thursdays); one Cardiologist performs 60%; the four hemodynamists contribute at the cardiology guard and on-call coverage pooled as one. At the moment two Cardiologists are absent for pregnancy.

An original software was developed in Java to collect data, generate the data model and read back the solution, relying on a free SQL database (Firebird 2.5 [2]) connected by standard JDBC driver. The graphic user interface was designed by NetBeans IDE 7.0.1 [3] taking advantage of the QuickTable component, a high performance database grid control javabean [4]. The open linear programming solver GLPK tool [5], invoked by means of the GUSEK interface[6], is integrated into the project and acts as the 'engine' for obtaining an effective solution to the problem.

2.1. The design

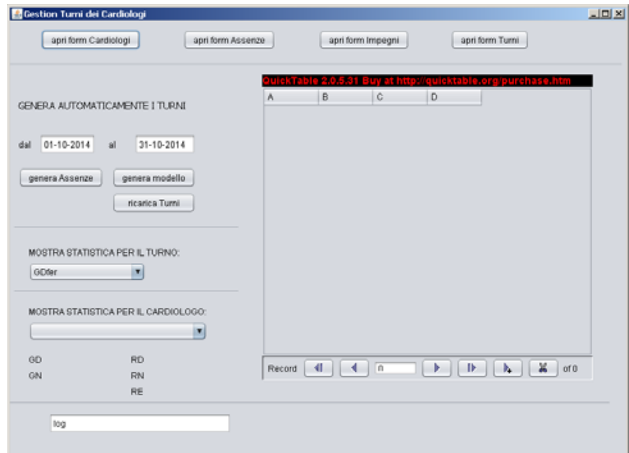
Data and constraints are defined in four tables stored in the SQL server.

Table CARDIOLO-GISTS	Table RESTS	Table HOLIDAYS	Table SHIFTS
ID	DATE	DATE	DATE
NAME	C1	C1	GD
SPECIAL	C2	C2	GN
PERCENTAGE	C3	C3	RD
COUNT_DIU	C4	C4	RN
COUNT_NOT	C5	C5	RE
COUNT_RN	C6	C6	
COUNT_WE1	C7	C7	
COUNT_WE2	C8	C8	
COUNT_WE3	C9	C9	
COUNT_WE4	C10	C10	
COUNT_WE5	C11	C11	
COUNT_RE		C12	
EMO		C13	
HIS		C14	
ECO		C15	
		C16	
		C17	
		C18	
		C19	
		C20	

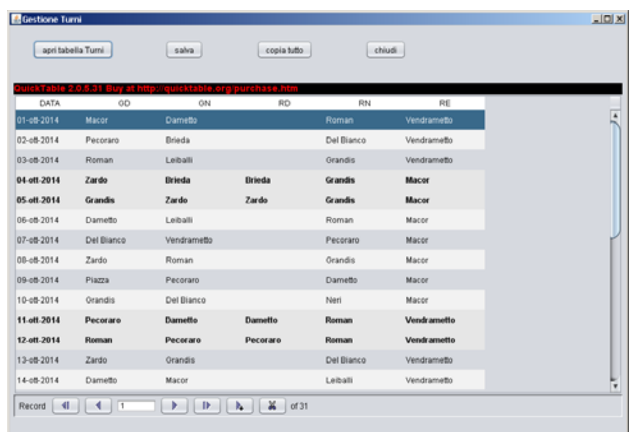
The first three tables store input data (Cardiologist quotas and special/group flags, planned rest days and previous shifts on public holidays respectively); the last one stores the solution loaded from the engine.

2.2. The user interface

The user interface was developed at first to help the user to define the problem characteristics. It allows to add, remove and change Cardiologist data, such as individual quotas and percentages, grant special conditions and group them by specialty. The top-left three buttons open data grids to update the first three tables of the database respectively.



Next are specified the start-end period and the buttons to generate the data for the GLPK model and to read back the solution generated by GUSEK, then accessed by the fourth top button, that opens the grid in read/write mode allowing the operator to make some adjustments by hand if needed.



Finally, the user interface provides total counts for each shift and for each Cardiologist to help the user to update input data to start the following period.

2.3. The model

The problem is expressed as a Mixed Integer Linear Programming (MILP) problem because variables are required to be integer or even binary. The model is developed using the default GLPK model language, the

GMPL (GNU MathProg Language) [7,8].

It uses assignment variables: each binary variable x_{gtc} takes value 1 if Cardiologist *c* is assigned a shift of type *t* day *g* and 0 otherwise.

The constraint set includes primary constraints that cannot be changed or deactivated, secondary constraints that are not strictly necessary to define a feasible plan, and soft constraints that can be violated. The objective function is a weighted sum of the violations of soft constraints. The three sets of constraints are detailed here below.

Primary constraints.

Primary constraints must be always satisfied by any feasible plan.

- Assignment: each slot must be assigned to a Cardiologist and each Cardiologist can be assigned either 0 or 1 work shift in each day.
- Pairings: some work shift assignments go in pairs: for instance, work shifts WE2 on Saturday and WE3 on Sunday must be assigned to the same Cardiologist in each week-end.
- Incompatible assignments: these constraints state that some pairs of assignments are infeasible. For instance:
  - work shift RE1 is incompatible with any work shift in the previous Friday;
  - a planned period of absence of a Cardiologist is incompatible with the assignment of night work shifts in the previous Friday (absence periods are always planned from a Saturday to a Sunday).
- Work shifts concentration: these constraints forbid the assignment of several work shifts to a same Cardiologist in time window of given width. The model includes concentration constraints related to work shifts of the same type, work shifts of any type, different time windows for working days and week-ends as well as particular concentration constraints for the subset of Cardiologists in charge of hemodynamics.
- Special assignments: these constraints correspond to particular individual needs of some Cardiologists.

Secondary constraints.

The mathematical programming model also includes several sets of secondary constraints that are not strictly necessary to define a feasible plan but are imposed to make the plan equitable and well-accepted by the Cardiologists. If their inclusion yields an infeasible instance, they can be relaxed or deactivated. However, in our experience this is not likely to happen.

- Work shifts repetitions: it is preferred that no Cardiologist is on duty on Friday night more than a given number of times in the planning period (e.g., at most once in a month). The same applies to work shifts in public holidays and week-ends.
- Undesired pairings: these constraints are similar to those on incompatible assignments. For instance:
  - it is desirable that Cardiologists who are assigned extra work shifts (RE1, RE2, HIS, ECO) are not assigned a work shift the night before;
  - similarly, when a Cardiologist is assigned work shift WE3 or WE4 on a Sunday, it is preferred he is not assigned any work shift on the next Thursday or Friday respectively to have a rest.

Soft constraints.

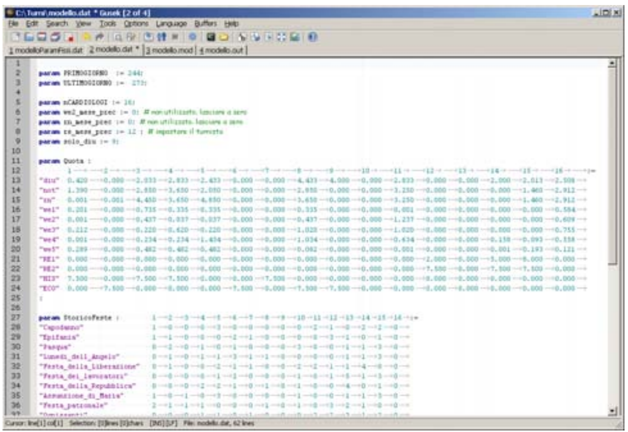
The objective function to be minimized represents a measure of the defects of the plan. It is a weighted sum of nine terms: three of them are *repetitions* and six are *quota violations.*

- Repetitions: requirements on repetitions are used to penalize multiple assignments of some work shifts to a same Cardiologist in a given time window. The number of assignments in excess with respect to the preferred value is measured by a suitable variable, whose value turns into a penalty in the objective function.
  - When a Cardiologist is assigned a work shift in a week-end or a public holiday, it is preferred he is not assigned any other work shift in the next week-end or public holiday in a prescribed time window (set to 10 days). Extra work shifts are not considered for this purpose.
  - When a Cardiologist is assigned a night work shift, it is preferred he is not assigned any other night work shift in a prescribed time window (set to 8 days) before being assigned a different type of work shift.
  - A third penalty is due to repeated assignments of work shifts in public holidays within four consecutive years to a same Cardiologist. For this purpose we record the number of times a work shift has been assigned to a Cardiologist in each public holiday in the last three years. Night shifts just before a public holiday are also counted as if they were public holiday work shifts.
- Quota violations: relying upon historical data, each Cardiologist is assigned a *quota* for each type of work shift. Quotas represent the ideal number of assignments that would produce a perfectly balanced distribution of work shifts to Cardiologists at the end of the planning period. Values of quotas are not integer in general and hence a perfectly balanced distribution cannot be achieved. However, quotas drive the planning process by keeping the amount of unbalance small at any point in time. Deviations in excess or in default with respect to quotas are penalized; they are represented by continuous and non-negative variables for each type of work shift and each Cardiologist. All types of non-extra work shifts are taken into account and work shifts RE1 also are. However WE1, WE2, WE3, WE4 and WE5 are considered in an aggregate way. Hence five variables are used for each Cardiologist. A sixth penalty

is computed for the violation of the overall quota, i.e. for the unbalance of the overall assigned workload. Extra work shifts are not considered for this purpose.

2.4. The solver

GUSEK [6] provides an open source IDE for Win32, linked to the GLPK standalone solver [5]. It takes in input three text files: the model itself, the year-based fixed data and the data dynamically generated by the user interface (see next figure) for the period. A set of parameters is passed to GLPK, i.e. '--cover --clique --gomory --mir' to force cuts to the search tree and optionally '--mipgap' or '--tmlim' to limit the optimization level or the computation time respectively.



The output of GUSEK is a new text file in which row by row are reported variables and constraints for each slot, including the binary x_{gtc} solutions. The user interface performs then a 'reverse engineering' on that file to extract values and store them in the table SHIFTS of the database.

3. Results

The system generates the shifts monthly. To harmonize contiguous months, usually the operator sets absent in the early days people working the last days of the previous period. The operator has the freedom to change some of the data generated by the user interface or even to change the model, by activating and deactivating constraints and modifying the weights in the objective function.

The solver can be used in two different ways by setting suitable parameters: in one case the search is directed to feasibility, to quickly check whether a feasible solution exists; in the other case the search is directed towards optimality, to produce an optimal solution. In our experience a few minutes of computing time are required to ensure that a solution exists (if not, the constraints are changed) and a few hours are required to obtain a highly optimal plan for one month.

The output is finally pasted into a worksheet for publication.

4. Conclusions

We consider the problem of equitably assigning different types of duties to Cardiologists in a hospital ward. The problem has been formulated as an integer linear programming problem with many types of hard and soft constraints, solved by the optimization module.

The mathematical model was produced 'pro bono' and the whole software package is freely available for download at: www.arc.fvg.it, together with a complete technical report [9].

The system has been in use for 22 months, continuously improved to handle specific situations. It provides equitable assignments of work shifts to Cardiologists and it is well-accepted by the personnel. Holidays and days off are guaranteed.

The system is currently used to generate the plans monthly. An improvement could be to test it in a rolling horizon planning procedure, i.e. generating two months, taking into account also the needs arising in month $m+1$ when the final plan for month m is decided. This implies longer computing time and therefore a stronger optimization process.

References

[1] Puente, Gmes, Fernandez, Priore. Medical doctor rostering problem in a hospital emergency department by means of genetic algorithms. Comp. and Industrial Engineering 2009;56: 1232-1242.

[2] Firebird 2.5 Quick Start Guide. IBPhoenix Editors. Firebird Project members. 26 September 2011, document version 4.4. http://www.firebirdsql.org/en/documentation/

[3] NetBeans IDE. https://netbeans.org/

[4] QuickTable. http://quicktable.org

[5] GLPK. http://www.gnu.org/software/glpk/.

[6] GUSEK. http://gusek.sourceforge.net/gusek

[7] GNU Linear Programming Kit: Modeling Language GNU MathProg, Draft edition, Jan 2004. Version 4.4.

[8] GNU Linear Programming Kit: Reference Manual, Draft edition, Jan 2004. Version 4.4.

[9] Note del Polo - Ricerca, n. 143, agosto 2014. Università degli Studi di Milano, Dipartimento di Informatica, sede di Crema.

Address for correspondence.

Eugenio Cervesato
Cardiology Dept.-ARC , 'Santa Maria degli Angeli' Hospital
Via Montereale, 24
33170 Pordenone
Italy
E-mail eugenio.cervesato@aopn.sanita.fvg.it