NURSE SCHEDULING PROBLEM

Erjon Duka MSc, Ing.
University of Durres FASTIP Albania

Abstract

In this paper, what i have been discussed, is analyzing penalties and cost shifts based on several elements for nurse scheduling problem (NSP). NSP’s issue is to assign nurses to different tasks based on constraints. The problem is known to be NP-hard, in other words it does not have a solution or needs years to be solved. In this work we try to solve the problem by satisfying the constraints set, and we also include the nurse’s preference and try to balance the difficulty level of all the involved nurses. We also analyze the complexity of the problem as a function of parameters such as number of nurses, number of shifts, and optimality of the function. According to the importance in practice, many scientists have developed NSP problems in a satisfactory time limit.

Keywords: NSP(Nurse Scheduling Problem), IP(Integer Programming), LIP (Linear Integer Programming), NIP(Non Linear Integer Programing), SIP(Scheduling in Integer Programming)

Introduction

Scheduling is always defined in the following way: Scheduling concerns in allocation of limit resources to tasks over time. It is a decision making process with the goal of optimizing one or more objectives e.g.: competition time or resource utilization. Most of works focuses in scheduling the time domain. The importance of good scheduling is strongly motivated by the present development of technology [1]. In most literature materials about scheduling problem are focused in two kinds of scheduling problems. One type is allocating resources to a program in order to optimize a given performance measure and the other type is scheduling the machine or other processors to produces a minimal time or cost. If there are tasks which can be performed by several devices, scheduling is needed in other case is not important (different devices performed exclusive tasks). A scheduling problem may not be hard to formulate but solving it is entirely another matter. Most scheduling problems are NP-hard. My problem, nurse scheduling is a NP-hard problem (Non- deterministic Polynomial-Time hard). NP-hard means that the result of the optimal time that we find can be verified in polynomial time. Combinatorial problems constitute an important set of problems in computer science and applied mathematics. Scheduling concerns the allocation of limit resources to tasks over time. A scheduling problem is defined by description of the processors, by description of the task and the measure of performance. Nurse Scheduling Problem is a typically Constraints Satisfaction Problem (CSP) since it consists in assigning a value from a finite domain to each variable of a finite set. In Constraints Programming the constraints are the same as Integer Linear Programming. The nurse's attendance in work is to a certain degree more important than the presence of doctors as they have to be alert all the time. The doctors are needed for an intervention, but the nurses are needed all the time as long as the patients are in the hospital, they are like baby-sitters: they have to be there. At first i will propose my problem, then i will explain the constraints and the code. I will run several experiments with the code in order to see the performance and i will discuss the results.

My code is structured by sets in Lingo 15 Application Software that is part of Lingo Systems. Sets are the foundation of Lingo Modulating Language and simple groups of related objects. A set must be a list of products, employees etch. Each member in the sets must have one or more characteristics associated with it. We call these characteristics attributes. My sets are:

```
Data
        N = 20;
        H = 7;
        T = 38;
        TMAX = 48;
ENDDATA
SETS:
        NURSE / 1..N/: RATE, DIFF, EXE, P;
        DAYS / 1..H/;
        WEEK /1..10/;
        ACTIVITY / REGULAR, DAY-A, DAY-B, NIGHT, SSPI/:
NBHOURS;
        JXK (DAYS, ACTIVITY): NEED, PENALITY;
        IXJ (NURSE, DAYS);
        IXJXK (NURSE, DAYS, ACTIVITY): AFFECT;
        IXK (NURSE, ACTIVITY);
        ENDSETS
```

In the sets we have nurses days and weeks. In the activity path we have the shifts that are regular DAY-A (first day shift), DAY-B(second day shift) we need to shifts in the first period that means double nurses more than night shifts), NIGHT(Night shift) and SSPI (Supervisor Shift). The number of Rows is the 20 as the number of nurses. The number of colums is the number of shifts (5Shifts per day). We have 7 groups of columns (7 days). In a row we must not have more that seven 1-s . For example the first nurse in the first day is not working. In the second day on Tuesday is working on the first shift, on Wednesday is not working too. We have 20 Nurses(N) but before running the code we can change the number of nurses in **N = 20;** The interval of adding/deleting must be [15;25]. The software will not run less than 15 nurses and more that 25 nurses in the period of 10 weeks. When we add a nurse in these rows we must add the binary numbers 1 and 0 in the code in the row of RATE and 0 in the row of EXE. The numbers after the exclamation mark are considered as comments.

RATE = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ; !.8 .8 .8 .8 .8 .8 .8 .8 .8 .8 .8 .7 .7 .5 .5 .5 .5 .3;

EXE =0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0; !0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;

Ps : we must have 20 binary 1-s and 0-s. RATE and EXE rows. We have 7 Days (N) in a week and 5 shifts (k). When we see these attributes IXJ (i=nurse, j=day, k=shift). Affect must be 0 and 1 that means it works or not. We have other activities Affect (Nurse, Day and Shift).

The values of the output are either 0 or 1, which means that a nurse does not work in a certain shift if the result is '0', and works if the result is '1' . Now we have the table for the Affect.

P is Penallty the value of Penallty. Diff is the difference between how many hours works per week and time minimum. *For example a nurse is working =8*3+1*12=36.*

*The number of variables is:7 days*5shifts/day*20nurse=700 (integer variables). From the definition of X_{ijk}, whre X can have values of '0' and '1'.*

i=1:20 (20 nurses), j=1:7 (7 days of the week), k=1:5 (5 shifts),

In the code there is a variable called Diff that calculates the difference between the working hours and the minimum time a nurse can work. The minimum time a nurse can work is 38 hours. For example: a nurse that works three 'regular' shift (8 hour/shift) and one 'Day_B' shift (12 hour/shift) then the total working hours are 8*3+1*12=36. Since the minimum working hours are 38, the DIFF-value for this nurse is 38-36=2.We can generate different type of constraints. If we want to arrange the schedule so that the 7th nurse never meets the 11th nurse, we can set a constraints of the type as below:

$$\mathbf{X_{7jk}+X_{11jk}<2}$$

Based on our output of the code, and based on the hours that each nurse works we can always minimize the number of nurses in order to reduce the costs of the hospital. In one run of the cod with 5 shifts, 20 nurses that was executed for 1 minute we notice that some of the nurses work 14 hours less than the minimum working hours which is 38 hours/week.

Diff = T_{min}-$T_{working}$=38-36=2

14	*3.6*
5.45	*3.6*
6	*3.8*
10	*3.8*
5.45	*3.6*
14	*3.6*
10	*3.8*
5.45	*3.6*
10	*3.8*
2	*4.2*
6	*4*
14	*3.6*
2	*4.2*
10	*3.8*
6	*4*
14	*3.6*
14	*3.6*
14	*3.6*
4.9	*4.2*
0.9	*4*
0	*0*
0	*0*
0	*0*
0	*0*
0	*0*
0	*0*

Total Diff and Penalties :

8.79375	**3.675**
14	**4.2**
5.45	**3.6**

Table 1 Diff and Penalties (N=20, 5 shifts, 10 weeks)

NSP Model Statistics:

```
Vars=     742  ( all are linear)
  Integer vars=     700 Binary vars=     700
  Nonzeros=     6711 Const nonz=     6240(     3400 are +- 1) Density= 0.014
  Smallest and largest elements in abs value=    1.00000        48.0000
No. < :      440 No. =:195 No. > :       20 No. posd :        0, Obj= MIN, GUBs
<
Single cols=      40
```

700 integer variables = 20 nurses * 7 days * 5 shifts. Binary Variables that means the value is zero or one. Nonzero variables means that we have no result equal to zero. Smallest and largest elements in absolute value are 1 and

48. We have also **NBHOURS = 8 12 12 12.55 8 8**. How many hours works a nurse in a shift.

Shifts	Hours
Regular	8
Day A	12
Day B	12
Night	12.55
SSPI	8
Auxiliary Shift	8

Table 2: NBHOURS

An attribute that is in the code is NEED that mean how many hours a nurse must need to work. T_{max} is an other attribute that is calculated **Tmax=Affect* NBhours.** For a nurse the Diff is the multiplication of penalty and NB-hours for a nurse

The excel file is updated it we change the number of nurses the tables will always change. If we have 18 nurses the table will have 2 rows less. In a hospital the different assigned tasks require the introduction of a new “shift” at the same time. So in this work an extra shift may define a completely different task at the same time at the same section of the hospital. An anasteolog may be needed, a nurse that is specialized in bones may be need etc. When we solve these problems, as the number of shifts increases the complexity of the problem is increased. A question until now, is that the weekly total penalty is 38. But in the outputs we get 76. And on average we get a penalty (difficulty score) of 3.8 (76/20=3.8). The total ‘penalty’ that is needed per week is 76. Because we have a need of 7 nurses for the ‘regular’ shift (7 nurses*5days*1),

$$\mathbf{7 \times 1 \times 5 + 1 \times 1.2 \times 5 + 1 \times 1.4 \times 5 + 1 \times 1.4 \times 2 + 1 \times 1.4 \times 5 + 1 \times 1.6 \times 2 + 2 \times 1.6 \times 5 = 76}$$

We executed the code for 4 seconds at least 3 times and we did not see any difference in the assigned jobs, the penalty values and the difference between the hours worked and the t_min (t_min=38 hours per week.) The only difference was in the ‘solver steps’ and the ‘solver iterations’. We observe a penalty of 3.8+/-0.3. When we executed the code for a longer time (60 seconds) we observed a smaller penalty value (P_{max}-P_{min}=0.6). The penalty is 3.8+/-0.2. We have a smaller standard deviation of the penalty value distribution. We also noticed that there are 54 different job assignments, i.e. a different of 7.7% in the job distribution. 2 subsequent executions of 1 minute (58.37) showed no difference at all for the assigned tasks. The Diff and penalties are exactly the same. We executed the code for 3 min and we noticed no change from the the 1 minute executed case. Everything was the same, the Pmax-Pmin (still 0.6), Diff, and the individual penalties. We can comment on the ‘Diff’ and the penalty for each nurse. For all the runs/execution of 4 sec and 60 seconds the values of Diff are the same

(Diff_ave=8.4075). The difference is in the distribution of these less-working hours.

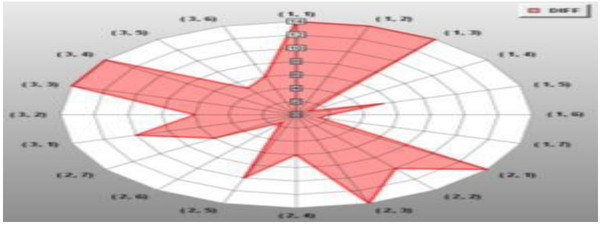

Figure 2 :The Diff for 5-shifts, 20 nurses, executed for 60 seconds

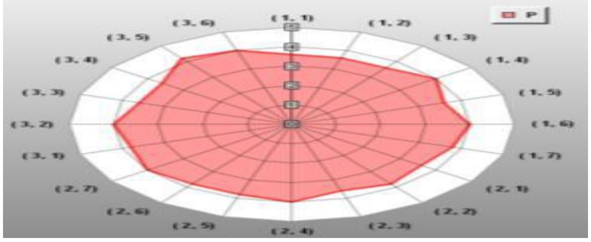

Figure 3 :The penalty for the 20 nurses.
The ideal case would be when we see a ‘perfect’ circle

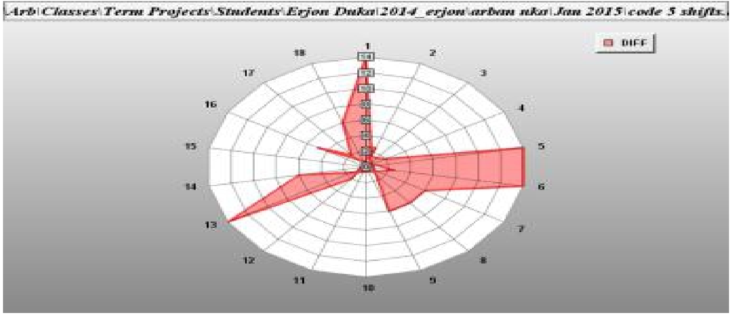

Figure 4 : 18 nurses, 5 shifts 57.85 seconds run

Review the Constraints

We can add different constraints for example Nurse 7 and Nurse 11 must nor work a day together

$$\mathbf{X7jk + X11jk >=1}$$

In the code we have added other constraints. To create and implement the code in Lingo 15 Application we must remember the constraints:

C1: Coverage constraints require a number of nurses for each shift (DS, EDS, ENS and S) and each day.

C2: Working hours must not exceed 12 hours per day

C3: Working hours must be close to 38 hours per week, and must not exceed 48 hours per week

C4: A nurse cannot work more than three night shifts during a week.

C5: If a nurse works an EDS (respectively ENS) on Saturday, then he/she also works an EDS on Sunday and than next Monday and Tuesday is free.

C6: This allows minimal rest time between 2(two) shifts. If a nurse works a Night Day Shift the following day is free. If a nurse works an EDS the following day is free.

In the graph is shown that nurse 5 works first night and second night but nut the third night. The first thing that we do in the code is after declaring the sets, attributes and constraints is minimize the gap between nurses with the highest rate of strain and the nurse having the low estate.

MIN= PMAX - PMIN;

The*@FOR* function is used to generate constraints across members of a set.

A binary integer variable also called a 0/1 variable is a special case of an integer variable that is required to be either zero or one. It's often used as a switch to model Yes/No decisions. In our case @BIN is associated to the Affect that we have told before is 0 and 1.

As we said before a nurse canbe assigned to onlyone activity per day.

```
@FOR (nurse(i):
@FOR (days (j): [unicity]
@SUM (ACTIVITY (k): AFFECT ( i, j, k))
<= 1.
```

Testing for 5, 6 and 7 shifts

After we have done a lot of test of NSP we create a table for the results. When we see carefully the table we detect that when we run the code several times without changing any data we see that objective value (our case to solve) change from 0.1 to 0.6 When we change nothing but also we add time the objective value becomes minimum from 0.8 to 0.6 and sometimes if we are lucky we can see it 0.1. The maximum time for running the code is 3 days from Friday to Monday in my work PC’lab in Fastip, University of Durres. The objective value was 0.12. Our scope is to make it 0.1 or the ideal value that is zero. In this table we have write in the columns : number of nurses, number of weeks, ESS (Extended Solver Steps), TSI (Total Solver Iterations) ERS(Elapsed Runtime Seconds that is time in the table) , Total Variables, Integer Variables, Total Constraints, Non-linear Constraints, Total Nonzero Non-linear Nonzero . The search have been stopped after a reasonable time for different reasons :

- Solvers , especially the ILP Solver have found a very good solution almost reaching the asymptotic value after a short time
- Even after 12 works of search the best value found was the one given after one hour
- A good schedule could be acceptable even if it has not exactly the minimal difference between Pmin and Pmax.

One interesting issue that I would like to underline in this thesis is that when i minimize(15 nurses) the number of nurses the Generator Memory Used of Lingo Program reduced itself by 1KB. The minimum number of starting the program is 15 Nurses. I cannot solve my application using 10 nurses. The minimum number of weeks would be 5 weeks .In other words program will not run in 10 nurses and 3 weeks.

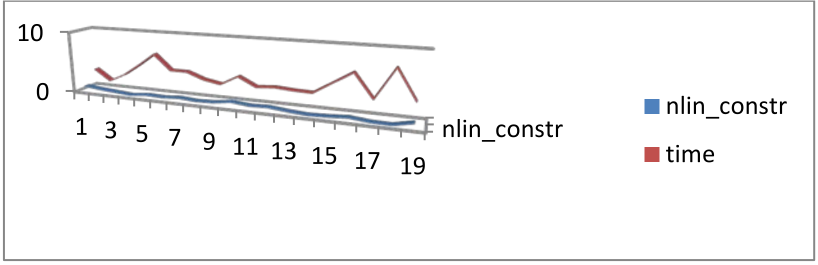

Figure 6 Non-Linear Constraints vs Time

The Objective Value vs. Time 20 Nurses 5 Shifts.

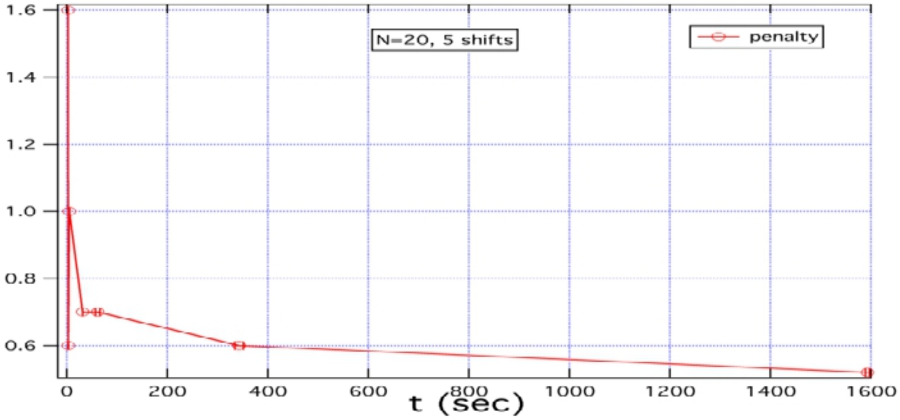

Figure 7 Penalty VS Time 20 Nurses 5 Shifts 10 weeks

In this graph we see that when the time executing is growning the objective value goes closed to Zero that is your scope. The Problem is resolved . All summary' penalties of the shifts for each employer are equal.

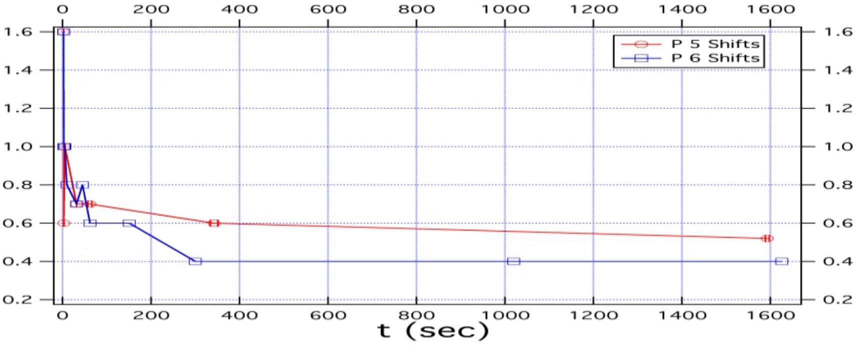

Figure 8. Objective value vs. Time: for 5 shifts and 6 shifts.

Interestingly, we notice a faster convergence (in computer time) for the case of 6 shifts. This seems counterintuitive, as with more shifts that have to be filled, there would be much more possibilities to check. We see that when we increase the number of shift the objective value is decreasing. It si fact that when we add a shift the software must do more iterations but the penalty is decreasing.

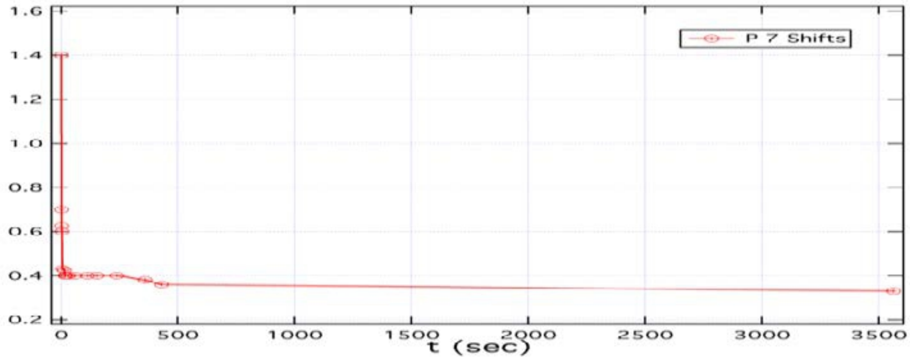

Figure 9. Objective value vs. time for 7 shifts.

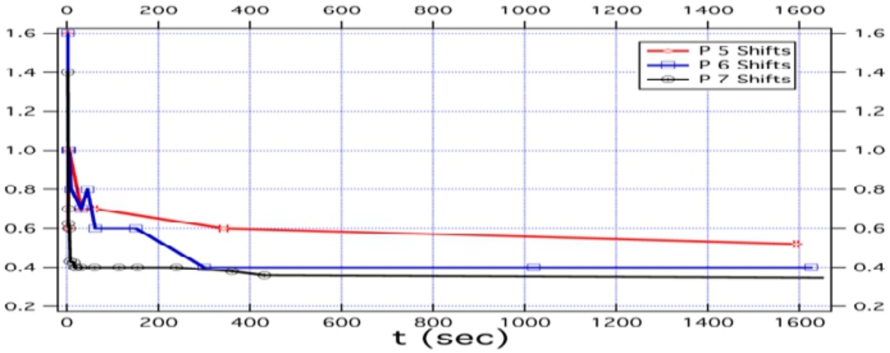

Figure 10. Objective value vs. time for 5, 6 and 7 shifts

In the last graph we notice that the case which has more combinations to check for an optimal solution, it takes less time. During the whole range of the values of time, the case with 7 shifts has always values that are lower than the cases for 5 and 6 shifts. At first look this seems very counterintuitive. One interpretation of these results may be that with more combinations to deal with, the easier it is to keep balance among all the nurses with respect of the penalty assigned to them. It is fact that when we add a shift the software must do more iterations but the penalty is decreasing. In the figure 11 we show the value of the optimal value of the penalty as a function of the total iterations for 6 and 7 shifts. Here we notice that when there are 7 shifts, for the same number of iterations, the objective value is smaller.

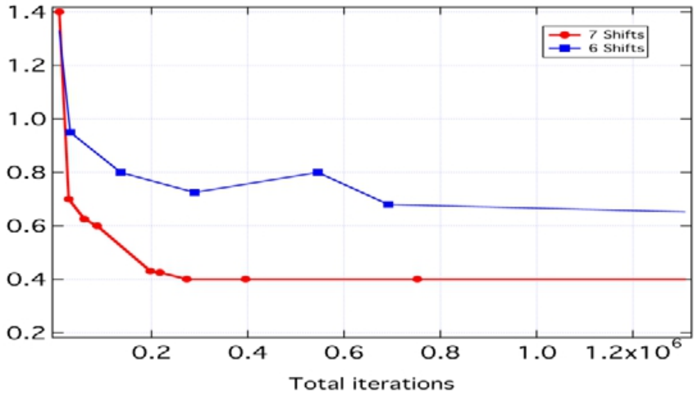

Figure11 The penalty as a function of the total iterations for 6 and 7 shifts

Conclusion

This project aims to apply learnings of operations research and optimizing resources to practical cases. The aim of this problem is to maximize the the fairness of the schedule , while respecting all the constraints . In regards with the results obtained after some tests ILP hase found a very good solution to our problem . Better values of the penalties associated to the shifts could be defined in order to represent the reality more accurately especially by taking into account the length of the shifts.

The models can also be solved by means of optimization software. As shown in this paper, the current schedules can benefit from this work. My problem is NP-hard that it means unsolveable.My objective is to do the objective values (diff =0.1) . The ideal must be 0 but it is impossible.there is shown in the table in 5 shifts , 20 nurses 10 weeks. We see that when we increase the number of shift the objective value is decreasing. It si fact that when we add a shift the software must do more iterations but the penalty is decreasing. We see that with the same number of iterations the penalty is smaller when we add a shift.

References:

[1] Nurse Scheduling Problem Trilling_Guinet_Lemagny_INCOM_2006 Lorraine Trilling University of Lyon 2006.

[2] Evolutionary AlgorithmsforNurseScheduling Problem. Ahmad Jan Masahito Yamamoto Azuma Ohuchi. Graduate School of Engineering, Graduate School of Engineering, Graduate School of Engineering, Hokkaido University, Hokkaido University, Hokkaido University.

[3] An Indirect Genetic Algorithm for a Nurse Scheduling Problem Computers & Operations Research, 31(5), pp 761-778, 2004.Uwe Aickelin School of Computer Science University of Nottingham NG8 1BB UK

[4] Global Journal on Technology Issue 6 (2014) 21-31 Selected Paper of

Global Conference on Computer Science, Software, Networks and Engineering (COMENG-2013) Nurse Scheduling Problem

[5] Formulating SPStochastic Programming Scenario Planning Models in What'sBest! www.lindo.com December 2011