Application of a Genetic Algorithm to a Scheduling Assignement Problem

Amândio Marques[a] and Francisco Morgado[b]
[a]CISUC - Center of Informatics and Systems of University of Coimbra, 3030 Coimbra, Portugal
[b]Polytechnic Institute of Viseu, 3500 Viseu, Portugal
email: amandio@dei.uc.pt, Tel: 239790057

Abstract

A set of heuristics are used successfully in a schedulling problem within the framework of healthcare medical services. Emphasis is given to the genetic algorithm which looks for the best schedule problem solution.

1 Introduction

The main objective of this work is the development of an intelligent system based on genetic algorithms to assist the planning of shifts scheduling in a local Hospital. The system will ease the current scheduling edition acting as an advisory to prevent uncorrected distributions assignment which lead to not enough resting periods of the health professionals and to a lack of parity concerning time and type of service. Clearly, these reasons cause inappropriate medical service care.

The specific objectives of the developed application are:

1. To take into account the number of working hours in excess or missing of the healthcare professional;
2. To allow a generic specification of the health care service requirements;
3. To visualize all the healthcare professionals assigned to a specific day shift;
4. To propose shift schedules sought for proper parity and balanced distribution for mid and long term;
5. To allow the adjustment of the proposed shift schedule.

2 Problem Formulation

The following parameters have to be defined:

F	- Set of healthcare professionals;
D	- Number of days of the schedule period;
T	- Number of Shifts;
Nec_{dt}	- Healthcare needs wrt Shift t of day d;
TRD	- Number of Shifts per day;

The problem solution can be formalized by the following variables which express each healthcare professional assignment to the care specific needs:

$$X_{fdt} = \begin{cases} 1 \text{ if professional } f \text{ does Shift } t \text{ of day } d \\ 0 \text{ otherwise} \end{cases}$$

with $f \in F$, $d \in D$, $t \in T$.

2.1 Problem Constraints

The solution admissibility is impose by the following constraints:

$$\sum_{f \in F} X_{fdt} = Nec_{dt} \qquad d \in D, t \in T \tag{1}$$

In each shift day the number of healthcare professionals have to satisfy the service needs.

$$\frac{\sum_{d \in D, f \in F} X_{fdt}}{\#D} = TRD \qquad f \in F \tag{2}$$

Each healthcare professional should fulfill the shifts specified in his working contract.

2.2 Objective Functions

The two objective functions to be minimized are as follows. Equation (3) is the objective function designated hereby Disorder:

$$Z_1 = \frac{\sum_{d \in D, f \in F, t \in T} (X_{fdt} \times penalty(fdt))}{\#F \times \#D \times \#T} \tag{3}$$

which corresponds to the mean of the penalties of the bad assigned shifts. The objective function Unfairness is given by (4):

$$Z_2 = 1 - \frac{\sum_{d \in D, t \in T} \left(Ptd_{td} \frac{1}{1 + mean\ dev(Ttr_{fdt}\ f \in F)} \right)}{\sum_{d \in D, t \in T} Ptd_{dt}} \tag{4}$$

It is calculated by the weighted mean of the inverse of the dispersion resulting from the service distribution per shift and type of day.

3 Heuristics versus Optimization Method

Although we have linear constraints, the problem can be cast in the context of multiobjective nonlinear binary programming, due to the nonlinearity of the objective functions.

Since linear methods are not adequate to solve this type of problems, other approaches seem adequate. Therefore, an heuristic based approach was here successfully applied and it will be described next.

```
For each day ( d )
   Calculates Shift Pattern (d,d+5)
   For all Shifts(t), from n down to 1
        While needs (d,t) are not fullfilled
          Select professional (f) more appropriate
          Assign professional (f) to the needs (d,t)
         End While
   End For
End For
```

4 Finding the Best Solution

We claim that for this problem (i) the performance of the proposed solutions can be properly evaluated, (ii) the problem is complex being NP complete; (iii) it has not been found yet an exact method to determine the best solution; (iv) the problem involves a large number of variables, thus occurring the curse of dimensionality.

For a problem with such characteristics the best approach to be used relies on Genetic Algorithms.

5 Genetic Algorithms

Genetic Algorithms (GAs) perform a stochastic global search method that mimics the metaphor of natural biological evolution. GAs operate iteratively on a population of individuals (solutions). In each iteration all the members are evaluated according to a fitness function. The lowest fitness individuals are eliminated and from the crossover of the remaining ones, a new generation of solutions is created following a mutation which is realized in a small percentage completing thus the cycle. This cycle is repeated until a stop condition is reached (see Figure 1).

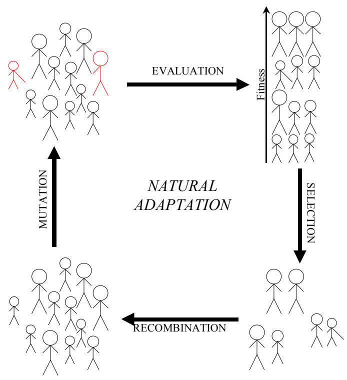

Figure 1: Genetic Algorithm Schema.

The individuals representation assumes an important role in any genetic algorithm approach In this case, the individual, a shift scheduling, is represented by:

$$X_{fdt} \qquad f \in F, d \in D, t \in T$$

The fitness of a specific solution is given the weighted sum of the objective Disorder and Unfairness. Since we have a multiobjective problem two strategies are presented.

In order to find a solution that minimizes a weighted average of the two objective functions, the ranking is obtained by sorting the solutions according to $Z = pZ_1 + (1-p)Z_2, 0 < p < 1$.

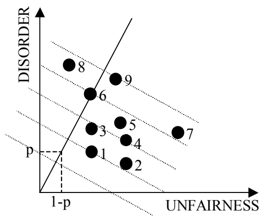

Figure 2: Function F_1.

To find the best solutions, either the Disorder or the Unfairness, the ranking should be done iteratively, capturing the trade-off between these objectives, the so-called Pareto curve (see Figure 2), inserting the found solutions in the ranking and removing them from the initial list. The process is repeated until any other solution can be found.

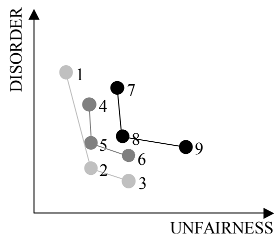

Figure 3: Function F_2.

The crossover operator combines two solutions (progenitors) from the actual generation with identical fitness and it generates two solutions (descendents) recombining portions of both parents. We take two solutions $XP1$ and $XP2$, and from a random $x \in D$ two new solutions $XD1$ and $XD2$ are generated as follows:

$$
\begin{aligned}
XD1_{fdt} &= XP1_{d<x} \cup XP2_{d\geq x} \;\; f \in F, d \in D, t \in T \\
XD2_{fdt} &= XP1_{d\geq x} \cup XP2_{d<x} \;\; f \in F, d \in D, t \in T
\end{aligned}
$$

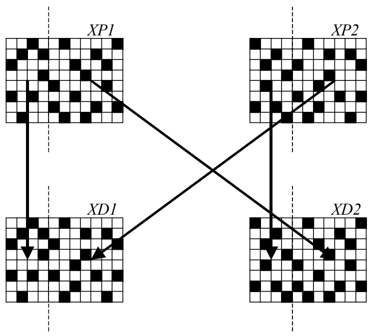

Figure 4: Crossover operation.

The mutation operates randomly on the chromosomes of an individual. To allow the convergence of an algorithm this operator is used with a low frequency. The mutation is achieved choosing randomly two heathcare professionals f_1, $f_2 \in F$, two days $d1, d2 \in D$, and two shifts $t1$,$t2 \in T$ that verify the following condition:

$$X_{f_1 d_1 t_1} = 1 \wedge X_{f_2 d_2 t_2} = 1 \wedge X_{f_1 d_2 t_2} = 0 \wedge X_{f_2 d_1 t_1} = 0$$

and changing their values, respectively.

$$X_{f_1d_1t_1} = 0 \wedge X_{f_2d_2t_2} = 0 \wedge X_{f_1d_2t_2} = 1 \wedge X_{f_2d_1t_1} = 1$$

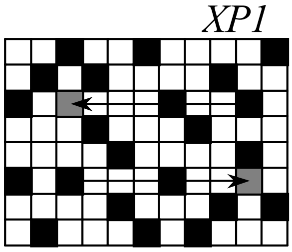

Figure 5: Mutation operation.

The evolution is associated with diversity. To achieve a good diversity on the initial population, based on the heuristic proposed, new heuristics were derived to generate purely random solutions, or random solutions to favoring the decreasing Disorder or to favoring the decreasing Unfairness.

6 Conclusion

The proposed approach solving a schedule shift problem base on a standard genetic algorithm was successfully implemented. The new heuristics generated a valid good shift scheduling. The genetic algorithm optimized the solutions found with the defined heuristics. The results show good agreement with the needs of the medical care service as well as the personal preferences of healthcare professionals.

7 References

1. Melanie Mitchell (1996). *An Introduction to Genetic Algorithms.* MIT Press, Cambridge, Massachusetts.
2. David E. Goldberg (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley Longman.
3. Stroustrup, Bjarne (1995). *The C++ Programing Language.* Second Edition, Addison-Wesley.
4. Guerreiro, Jorge et al.(1985), *Programação Linear*, Vol II, Lisboa, Mcgraw-Hill.
5. Niklaus Wirth (1975) . *Algorithms & Data Structures.* Prentice-Hall.