A PSO algorithm to solve a Real Anaesthesiology Nurse Scheduling Problem

Leopoldo Altamirano, *Member IEEE* , María-Cristina Riff, *Member IEEE* and Lorraine Trilling

***Abstract*—In this paper we present an approach especially designed to solve the real world problem of Anesthesiology Nurse Scheduling (ANSP) at a French public hospital, where the anesthesiology nurses are one of the most shared resources. We propose a Particle Swarm Optimization algorithm to solve the ANSP. We compare our technique with previous results obtained using Integer Programming and Constraint Programming. The objective is the same for all methods: to maximize fairness of the schedule.**

I. Introduction

PARTICLE Swarm Optimization (PSO) was originally designed as a numerical optimization technique based on swarm intelligence. In the literature, there are a few attempts to exploit its usage in the discrete problem domain [10], which are mostly performed using a binary encoding. However, work on the transformation of the working mechanism of PSO to the permutation problem domain, where the representations are highly constrained, has been relatively limited [8]. This limitation is mainly caused by the lack of a principled generalization of PSO to guide its adaptation to discrete combinatorial problems. In this paper, we aim to design a PSO algorithm for a real world problem with discrete domains without losing the underlining principles of the original PSO.

The Anesthesiology nurse scheduling problem (ANSP) tackled in this paper is a real world problem occurring in a French public hospital. The assigning of shift work to nurses over a period of several days is a hard consuming task. Usually this task is done manually. This group of nurses is more often organized into a team of equally skilled nurses who assume various activities during the inpatient care process. They are also able to perform those activities which overlap some surgical specialties. Those equally skilled nurses can be assigned to different activities from one day to another. The goal is to find a schedule where the distribution of the tasks is balanced throughout the nursing staff.

The contributions of this paper are:

- A new application of the PSO technique in discrete domains
- New scheduling solutions obtained by using a new evaluation function to guide the algorithm. These solutions show a better work distribution than those previously reported using Constraint Programming and Integer Programming for the set of problems tested.

This paper is structured as follows. In the next section we present the related work. Section three briefly describes the real world problem. In section four we introduce our algorithm. Section five presents the evaluation using real data, and finally, section six gives the conclusions of our work.

II. Related Work

Nurse scheduling problems (NSP) or nurse rostering problems (NRP), which involves the creation of individual schedules, have been widely studied over the last decades. Several very recent and complementary bibliographic surveys have been published and give a good overview of the problem modelling and the various approaches [18], [17]. In the literature, two types of scheduling can be found: cyclical and non-cyclical scheduling. A popular approach is to construct cyclical schedules, where the same schedule is repeated as long as the requirements do not change. These kinds of schedules are easy to build but may be very rigid, and may adapt difficultly to changes. In a non-cyclical scheduling process, a new schedule is generated for each scheduling period. This process is more time-consuming but is much more flexible to changes such as the variability of demand [16] Several studies have employed optimization methods to solve the NSP, like linear, integer or mixed integer programming [13], goal programming [3] or constraint programming [19]; [1]. Many of more recent papers tackle the NSP with metaheuristic methods such as genetic algorithms [2], tabu search [3], [16], [2] or simulated annealing [5].

III. ANSP Description

The surgical suite is a complex service, which includes two main parts: the operating rooms (ORs), and the post-anesthesia care unit (PACU). Different teams of nurses working in this service have to be scheduled: the operating room nurses, assisting surgeons during surgery; the anesthesiology nurses, taking care of inpatients during surgery or supervising the PACU; the registered nurses, taking care of inpatients during recovery; and finally, the auxiliary nurses, performing the logistical tasks as well as the cleaning tasks. Anesthesiology nurses work together with anesthesiologists during surgery but also during the recovery time. They can work in a cross section of different surgical specialties and several types of surgery (scheduled cases, ambulatory cases or emergency cases). In hospitals, that includes a surgical emergency service, where some nurses work around the clock (on-call or on stand-by), even in the operating suite, where

This work is partially supported by Fondecyt Project 1080110. Leopoldo Altamirano and María-Cristina Riff, Universidad Técnica Federico Santa María, Computer Science Department, Valparaíso, Chile
e-mail:{Leopoldo.Altamirano, Maria-Cristina.Riff}@inf.utfsm.cl
Lorraine Trilling, Département Génie Industriel, LIESP Laboratory, INSA of Lyon, France, e-mail: Lorraine.Trilling@insa-lyon.fr

Type of shift	start-end time
Day shift(DS)	8:00 - 16:00
Emergency day shift (EDS)	8:00 - 20:00
Emergency night shift (ENS)	20:00 - 8:00
PACU supervision (PS)	9:00 - 17:00; 11:00 - 19:00

Fig. 1. Type of shifts, start and end time

Type of shift	Mon	Tue	Wen	Thur	Fri	Sat	Sun
DS	1	1	1	1	1	0	0
EDS	1.2	1.2	1.2	1.2	1.2	1.4	1.4
ENS	1.4	1.4	1.4	1,4	1.4	1.6	1.6
PS	1.6	1.6	1.6	1.6	1.6	0	0

Fig. 2. Penalty associated to the shifts

surgical teams have to be rapidly ready to welcome a new patient. Nurses can be assigned to either day or night shifts, and each day to one activity. This daily activity can change from one day to another. The problem presented in this paper has been studied in a French hospital, with a surgical suite containing 9 ORs. The part-time and full-time nurses are all equally skilled, and can be assigned to any of the shifts day or night according to the figure 1. The emergency shifts are on stand-by duty and have to be performed on each day of the week (including week-ends), whereas, the other shifts are from Monday to Friday. The shifts involving the supervision of the PACU are considered as equivalent. The anaesthesiology head nurse in a non-cyclical process constructs the anaesthesiology nurse's schedule. First the head nurse gathers nurses'preferences concerning the day they would like to be off and from this, tries to elaborate schedules which satisfy all the constraints listed below. Given that nurses work on an annual schedule, if during one week a nurse has an overload compared to the standard number of working hours per week, then he/she can recover the over-time worked during the following week or later. The constraints that have to be satisfied are the following:

- **C1:** Coverage constraints require a number of nurses for each shift (DS, EDS, ENS, PS) 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 given week
- **C5:** If a nurse works an EDS (respectively ENS) on Saturday, the he/she also works an EDS (respectively ENS) on Sunday, and the next Monday and Tuesday are free.
- **C6:** Succession of activity constraints, which allow minimal rest time between two shifts (equal to 11 hours), have to be satisfied such as:
  - If a nurse works a NDS, the following day is free;
  - If a nurse works an EDS, the following day could be either an ENS or could be free. Given that twice more nurses are required for the EDS than for the ENS during the week, half of the nurses performing an EDS would work an ENS the following day, and the rest would be free.

The constraints **C1** to **C4** are considered as compulsory constraints that have to be respected (hard constraints). Usually, constraints **C5** and **C6**, as well as the constraints of preferences expressed by the nurses, could be optional however an acceptable plan could violate some of them (soft constraints). Since the objective is to find a good plan that maximizes nurses satisfaction, the required solution must ideally satisfy all the constraints from **C1** to **C6**. In order to generate the fairest schedule, the popular and unpopular shifts have to be distributed among the nurses in a balanced way, taking into account the difficulty of each shift, according to the head nurses knowledge. To express the popularity of the shifts, a penalty associated to each type of shift and to each day of the week is defined. The penalties are rational values and are included in the interval $\{1, \ldots, 2\}$. These penalty values are shown in figure 2. This table has been constructed with the information given by the involved nurses. The best schedule would be the one that minimizes total penalty associated to each employee. The following section presents how this problem can be solved using PSO.

IV. A PSO-BASED ALGORITHM FOR ANSP

In this section we will describe all the components of our algorithm. The algorithm works to solve a problem with discrete domains, thus in order to apply PSO to this problem we must be careful about the meaning of both the position and the velocity in our context. In a classical PSO, the vector position indicates the values of the variables belonging to a continuous domain. We define the representation (similar to the vector position idea) as a matrix X of size Nxd, where N is the number of nurses and d represents the day. The values input into the matrix correspond to the type of assignments (DS,EDS,ENS,PS) for each pair (nurse, day).

A. Evaluation Function

In previous research, [14], [15] the authors have defined a fitness function for this problem as $Z = Pmax - Pmin$, where $Pmax$ and $Pmin$ are the maximum and minimum penalty values respectively, among all nurses. The idea was to minimize Z to obtain a fair preference assignment. From our analysis we have decided to change Z to a new evaluation function which does not only regard the extreme penalty values but it takes into account the inequality of the assignments between $Pmax$ and $Pmin$. Before introducing the evaluation function we require the following definitions:

Definition 4.1: Ideal Work distribution: Given the Total-Required, the number N of nurses, we define the function to measure the Ideal Work distribution IW as,

$$IW = \sum_{i=1}^{N}(C * log(C)) \tag{1}$$

where C is computed as

$$C = \frac{\text{Total-Required}}{N} \tag{2}$$

The function IW represents the lower bound without considering any constraint in its computation.

Definition 4.2: Current Work distribution: Given the matrix X, we define the function to measure the Current Work distribution CW as,

$$CW = \sum_{i=1}^{N}(c_i * log(c_i)) \quad (3)$$

where c_i is the penalty of the nurse i in the current candidate solution.

All the candidate solutions satisfy the hard constraints of the problem, thus usually the CW value is higher than the IW one. Finally, the evaluation function is computed as

Definition 4.3: Given the Ideal Work distribution IW and the Current Work distribution CW values, we define the evaluation function Fair Work distribution FW by:

$$FW = \alpha * (\frac{CW}{IW} - 1) \quad (4)$$

with α a constant equal to 10000. We have defined this value in order to amplify the differences between very similar solutions.

B. Initial Population

The initial population searches to obtain particles that satisfy all the hard constraints. It corresponds to a greedy procedure. The procedure begins by assigning the weekend shifts, which are the most conflicting ones. The procedure randomly selects a feasible nurse and assigns her to a weekend shift. The weekend assignment procedure continues until satisfying all the weekend requirements. Once weekend assignments are completed the procedure continues to the other week days. It randomly selects an allowed pair (nurse, day) and when it is possible, assigns a permitted shift into the matrix. In case it is not possible, the procedure tries another assignment. The iterative process is repeated for a maximum number of 2000 tries. This procedure is not expensive, it takes less than 3 seconds per particle and allows for the generation of only feasible particles.

C. Moves

We have considered two moves in our algorithm for evolving the particles:Swap-shifts and Swap-best. Swap-shifts is applied inside a particle, and the Swap-best is applied between particles. The accepted criterion for all moves is the improvement of the solution (like a hill-climbing procedure).

1) Swap-shifts: : Given a particle, the procedure randomly chooses both two nurses and a day. It swaps the assignment of the day between the two nurses. This change is accepted only when the evaluation function of the particle improves. When it is accepted, both the velocity and the local best of the particle are updated.

2) Swap-best: : Given a current particle and a *reference particle*, the procedure randomly selects a nurse. It identifies the penalty value for the nurse selected from the *reference particle*. It randomly selects a day and tries to obtain the same nurse penalty of the *reference particle* in the current particle by swapping between the nurses on this day. Once a swap allows the same penalty value to occur in the current particle the procedure stops. The current particle changes only when the move allows the same penalty value of the *reference particle* to occur. This move tries to guide the current particle to the *reference particle* by using the nurse penalties. At this step, it is important to remark that each particle knows the best solution obtained from its own history, is that which corresponds to the best local solution. The best global solution is the best solution obtained from the history of all the particles. In the algorithm, Swap-best is applied in the following two configurations. As *reference particle* one configuration uses the best local particle, and the other uses the best global particle.

D. $PSO - ANSP$

The algorithm is shown in figure 3. The algorithm begins by creating the initial set of feasible particles and computing their fitness. It uses the new evaluation function described above. The first LocalBest for each particle is itself. The GlobalBest corresponds to the particle with the best value of its evaluation function. The iterative process begins by computing the velocity of each particle using the following equation:

$$v_{ij}(t+1) = wv_{ij}(t) + c_1 r_{1j}(t)|y_{ij}(t) - x_{ij}(t)| + c_2 r_{2j}(t)|\hat{y}_j(t) - x_{ij}(t)|$$

where:

- w: is a parameter related to the last velocity in time t.
- x_{ij}: is the current position for the j dimension of the particle i.
- y_{ij}: is the best local position for the j dimension of the particle i (cognitive factor).
- $\hat{y}_j$: is the position j of the best global position (social factor).
- c_1, c_2 : parameters for the cognitive and social components respectively.
- r_{1j}, r_{2j} : random values from an uniform distribution $U(0, 1)$.

We consider the maximum velocity value equal to 1. The interpretation of the velocity in our algorithm is a probability value to use either the LocalBest or the GlobalBest during the search. The move Swap-shifts is applied to a particle randomly selected in order to improve it. When the particle improves then it updates its LocalBest and its velocity. At the beginning of the algorithm, all velocities are equal to 1. Thus, the particles are inclined to apply a more GlobalBest search than a LocalBest search. In the first interations the algorithm does more exploration and in the last iterations it does more exploitation of the search space.

V. RESULTS AND EVALUATION

The hardware platform for the experiments was a PC Intel Corei7-920 with 4GB RAM under the Linux Mandriva 2010 operating system. The algorithm has been implemented in C++. We have compared our algorithm with Integer Linear Programming (ILP) and Constraint Programming (CP) techniques and evaluated their efficiency by solving some

PSO-ANSP
Begin
Initialise population of X particles (1)
Calculate Fitness (Particles) (2)
For Particle=1 to X
  LocalBest(Particle)= Fitness(Particle) (3)
EndFor
GlobalBest = Min(LocalBest(Particles)) (4)
$i = 0$;
Repeat
  Compute Particles velocity (5)
  p=random(1, X) (6)
  Swap-shifts(p) (7)
  If velocity(p) $<$ random(0,1) **then** (8)
   Swap-best(p, LocalBest) (9)
  Else
   Swap-best(p, GlobalBest) (10)
  Endif
  Verify-best(p) (11)
  $i = i + 1$;
Until i=Max-number-of-iterations
End

Procedure Verify-best(p)
Begin
  If Fitness(p) is better than the LocalBest(p) **then**
   LocalBest(p)=p (12)
  End if
  If Fitness(p) is better than GlobalBest **then**
   GlobalBest=p (13)
  End if
End

Fig. 3. PSO-ANSP

	Nurses	Days	Requirements	DS	EDS	ENS	PS
P1	10	14	Mon to Fri	2	2	1	1
			Sat to Sun	0	1	1	0
P2	20	7	Mon to Fri	7	2	1	2
			Sat to Sun	0	1	1	0
P3	16	7	Mon to Fri	7	2	1	2
			Sat to Sun	0	1	1	0
P4	16	14	Mon to Fri	7	2	1	2
			Sat to Sun	0	1	1	0

Fig. 4. Parameters of the experimented tests

instances of ANSP of different sizes (figure 4). In the samples considered, all the nurses are regarded as full-time nurses. The objective function values for CP and ILP reported in figure 5 are not exactly the optimal values, but the last values obtained when cancelling solvers before completion, i.e. after one hour of a search as it was reported in [14]. The CPU time is when the solver found this solution. In the case of PSO, it is the the best solution obtained after 10 runs for each problem using 3 particles. From the results, PSO has been able to find the optimal solution for problem P1. For the other problems, ILP and PSO have obtained similar results.

A. Analysis of PSO

Using just three particles and ten runs of the algorithm, it has been able to obtain very good results. Figure 6 shows two

	ILP Model			
	Pmax	Pmin	Obj	CPU
P1	8.8	8.4	0.4	14
P2	4.2	3.6	0.6	62
P3	5.0	4.6	0.4	183
P4	9.6	9.4	0.2	241
	CP Model			
	Pmax	Pmin	Obj	CPU
P1	9	8.2	0.8	2168
P2	5.6	2.0	3.6	25
P3	5.4	4.0	1.4	28
P4	10.6	8.8	1.6	1067
	PSO			
	Pmax	Pmin	Obj	CPU
P1	8.6	8.6	0.0	18
P2	4.2	3.8	0.4	77
P3	5.0	4.6	0.4	57
P4	9.6	9.4	0.2	58

Fig. 5. Experimental Results, CPU in seconds, Obj=Pmax-Pmin

feasible assignments for P3. Both assignments are equivalent in terms of the difference between Pmax and Pmin. However, our algorithm prefers the candidate solution with a FW value equal to $1,77836$. This situation accurately illustrates our initial goal of discriminating among all nurses and not only between the extreme penalty values. Another interesting case is ilustrated for the problem P2 in figure 7. The first figure shows the best solution for P2 in terms of the Pmax-Pmin function, however it has a high FW value. The second figure corresponds to the best solution considering the FW function. We consider that in terms of fairness the second solution is better. The function FW is able to express both this difference and our preference.

In the figures 8, 9, 10 and 11 we report the results obtained by PSO with a different number of particles (3, 5, 10, 15) for each problem. For example, for P2 using 5 particles the algorithm has obtained the same Pmax-Pmin value in all the runs, however these solutions in terms of the FW have different values. The algorithm using more particles obtains a set of alternative solutions with the same fitness value. It is interesting from the point of view of the head nurse, because knowing these alternative solutions she can use a subjective criteria to select one of the assignments.

VI. Conclusions

In this paper, we propose a Particle Swarm Optimization algorithm for solving the ANSP problem. To best guide the search of our algorithm, we have defined a new evaluation function, which searches for minimum unfair assignments among the nurses. Using this function, the algorithm has been able to obtain the optimal value for problem P1. This new evaluation function allows a better discrimination between candidate solutions and is in accordance with the idea of the original function, which searched for assignment fairness. The algorithm has shown a similar behaviour as ILP has

	DAY 1	DAY 2	DAY 3	DAY 4	DAY 5	DAY 6	DAY 7	PENALTY
Nurse 1	DS	DS	DS		PS			4.6
Nurse 2			DS	EDS		EDS	EDS	5.0
Nurse 3	DS	EDS	ENS		DS			4.6
Nurse 4	EDS	ENS		DS	DS			4.6
Nurse 5	ENS		DS	DS	ENS			4.8
Nurse 6	PS		DS	DS	EDS			4.8
Nurse 7		PS				ENS	ENS	4.8
Nurse 8		DS	PS	DS	EDS			4.8
Nurse 9		DS	DS	PS	DS			4.6
Nurse 10	DS	DS	DS	DS	DS			5.0
Nurse 11	PS	DS	EDS		DS			4.8
Nurse 12	EDS		DS	DS	PS			4.8
Nurse 13	DS	DS	PS	EDS				4.8
Nurse 14	DS	DS	EDS	ENS				4.6
Nurse 15	DS	EDS		PS	DS			4.8
Nurse 16	DS	PS		DS	DS			4.6
							PMAX	5
							PMIN	4,6
							FW	2,48228

	DAY 1	DAY 2	DAY 3	DAY 4	DAY 5	DAY 6	DAY 7	
Nurse 1		DS		DS		EDS	EDS	4.8
Nurse 2	DS	DS	DS	DS	DS			5.0
Nurse 3		DS	PS	DS	EDS			4.8
Nurse 4	DS	EDS	ENS		DS			4.6
Nurse 5	PS	DS	DS	EDS				4.8
Nurse 6	DS	EDS		PS	DS			4.8
Nurse 7	ENS		DS	DS	ENS			4.8
Nurse 8	PS		DS	DS	EDS			4.8
Nurse 9	DS	PS	EDS		DS			4.8
Nurse 10	DS	DS	EDS		PS			4.8
Nurse 11	DS	PS	DS	EDS				4.8
Nurse 12		DS	PS	DS	DS			4.6
Nurse 13	EDS		DS	PS	DS			4.8
Nurse 14	EDS	ENS		DS	DS			4.6
Nurse 15	DS	DS	DS		PS			4.6
Nurse 16				ENS		ENS	ENS	4.6
							PMAX	5
							PMIN	4,6
							FW	1,77836

Fig. 6. Two candidate solutions for P3

	DAY 1	DAY 2	DAY 3	DAY 4	DAY 5	DAY 6	DAY 7	
Nurse 1			DS			ENS	ENS	4.2
Nurse 2	DS	DS		DS	DS			4.0
Nurse 3	DS			PS	ENS			4.0
Nurse 4	ENS			DS	PS			4.0
Nurse 5		PS	DS	EDS				3.8
Nurse 6	PS		EDS	ENS				4.2
Nurse 7	EDS		PS	EDS				4.0
Nurse 8	PS	ENS		DS				4.0
Nurse 9		EDS				EDS	EDS	4.0
Nurse 10		DS	DS	DS	DS			4.0
Nurse 11	EDS		DS	PS				3.8
Nurse 12	DS		ENS		PS			4.0
Nurse 13		DS	PS		PS			4.2
Nurse 14	DS	DS	EDS		DS			4.2
Nurse 15	DS	EDS		DS	DS			4.2
Nurse 16		PS	PS		DS			4.2
Nurse 17	DS	DS	DS		DS			4.0
Nurse 18	DS	DS	DS		EDS			4.2
Nurse 19	PS			DS	EDS			3.8
Nurse 20		DS	DS	DS	DS			4.0
							PMAX	4.2
							PMIN	3.8
							FW	1123,810671

	DAY 1	DAY 2	DAY 3	DAY 4	DAY 5	DAY 6	DAY 7	
Nurse 1	DS		PS	EDS				3.8
Nurse 2	DS	DS	DS	DS				4.0
Nurse 3	DS	DS		DS	DS			4.0
Nurse 4	EDS		DS	ENS				3.6
Nurse 5		ENS		DS	EDS			3.6
Nurse 6		EDS		PS	DS			3.8
Nurse 7	PS			DS	EDS			3.8
Nurse 8	ENS		ENS		DS			3.8
Nurse 9	PS	DS		EDS				3.8
Nurse 10	EDS		DS		ENS			3.6
Nurse 11		DS	DS	DS	DS			4.0
Nurse 12	DS					ENS	ENS	4.2
Nurse 13	DS	PS	EDS					3.8
Nurse 14		PS	EDS		DS			3.8
Nurse 15	DS	DS	DS	DS				4.0
Nurse 16		DS		DS	PS			3.6
Nurse 17			DS			EDS	EDS	3.8
Nurse 18	DS	EDS			PS			3.8
Nurse 19		DS	PS		DS			3.6
Nurse 20			DS	PS	DS			3.6
							PMAX	4.2
							PMIN	3.6
							FW	7,213935

Fig. 7. Two optimal solutions for P2

for the other problems, outperforming the results obtained by using Constraint programming techniques. Our algorithm required less time to find a quality solution than it took the other evaluated techniques, especially in comparison to constraint programming techniques. Moreover, our technique has been guided by a non-linear function for searching quality solutions. It is a promising research area and a new application of PSO to solve a real-world problem with discrete domains.

REFERENCES

[1] Adbennadher, S. and H. Schlenker (1999). Nurse scheduling using constraint logic programming. Proceedings of AAAI/IAAI.

[2] Aickelin, U. and K. A. Dowsland (2004). An indirect Genetic Algorithm for a nurse-scheduling problem. Computers and Operations Research, 31, 761778.

[3] Berrada, I., J. A. Ferland and P. Michelon (1996). A multi-objective approach to nurse scheduling with both hard and soft constraints. Socio-Economic Planning Sciences, 30, 183-193.

[4] Blöchliger, I. (2004). Modeling staff scheduling problem. A tutorial. European Journal of Operational Research, 533-542.

[5] Brusco, M. J. and L. W. Jacobs (1995). Cost analysis of alternative formulations for personnel scheduling in continuously operating organisations. European Journal of Operational Research, 86, 249261.

[6] Burke, E. K., P. D. Causmaecker, G. V. Berghe and H. V. Landeghem (2004). The state of the art of nurse rostering. Journal of scheduling, 7, 441-499.

[7] Cheang, B., H. Li, A. Lim and B. Rodrigues (2003). Nurse Rostering Problems - A Bibliographic Survey. European Journal of Operational Research, 151, 447-460.

[8] Clerc, M. Particle Swarm Optimization. ISTE Publishing Company, February 2006.

[9] Ernst, A. T., H. Jiang, M. Krishnamoorthy and D. Sier (2004). Staff scheduling and rostering : A review of applications, methods and models. European Journal of Operational Research, 153, 3-27.

[10] Gong, T., Tuson A.(2008). Particle Swarm Optimization for Quadratic Assignment Problems-A Forma Analysis Approach. International Journal of Computational Intelligence Research, Vol.4, N. 2, 177-185.

[11] Guinet, A. (1995). Scheduling independent jobs on uniform parallel machines to minimize tardiness criteria. Journal of Intelligent Manufacturing, 6, 95-103.

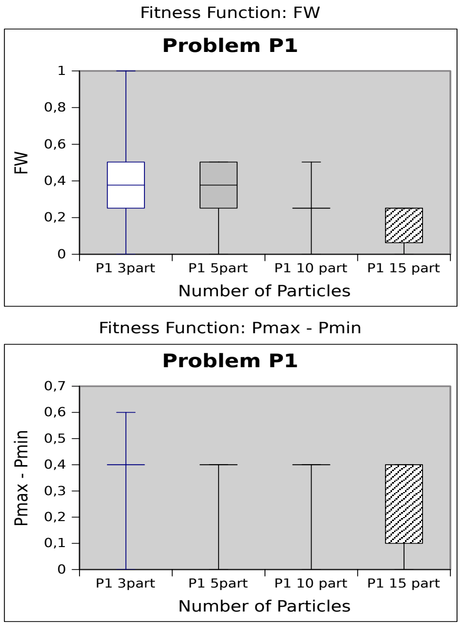

Fig. 8. Boxplots for P1

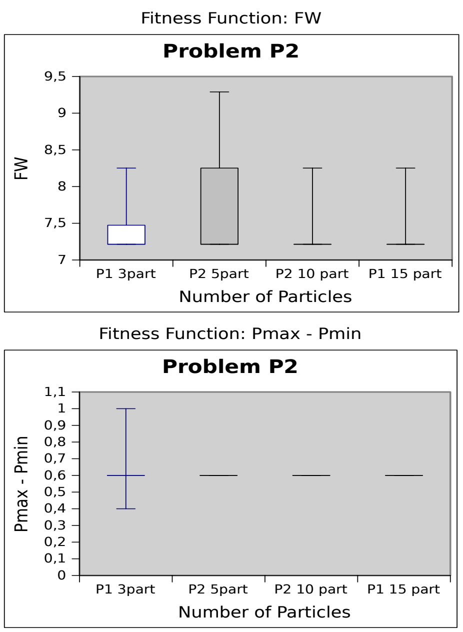

Fig. 9. Boxplots for P2

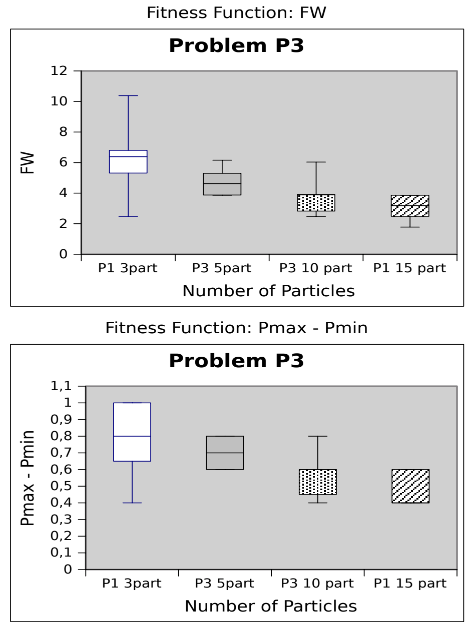

Fig. 10. Boxplots for P3

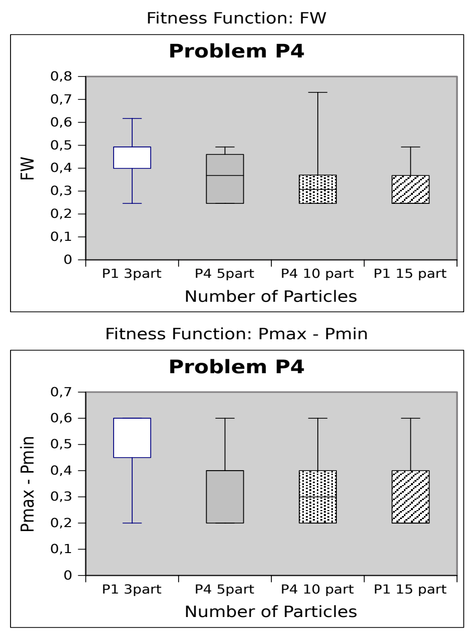

Fig. 11. Boxplots for P4

[12] ILOG (1998). ILOG Solver 4.4 Reference Manual, Gentilly, France.
[13] Jaumard, B., F. Semet and T. Vovor (1998). A generalised linear programming model for nurse scheduling. European Journal of Operational Research, 107, 1-18.
[14] Trilling L., Guinet A, and Le Magny D. (2007) Journal of Logistique et Management, Volume 15, Issue 1, 5-16.
[15] Trilling L., Guinet A. and Le Magny D. (2006). Nurse Scheduling using Integer Linear Programming and Constraint Programming. Proceedings of the 12th IFAC International Symposium, Elsevier , Volume 3, Saint-Etienne, France, 651-656.
[16] Valouxis, C. and Housos E.(2000). Hybrid optimization techniques for the workshift and rest assignment of nursing personnel. Artificial Intelligence in Medicine, 20, 155-175.
[17] Vanhoucke, M. and Maenhout, B. (2007), NSPLib–a tool to evaluate (meta-)heuristic procedures, Operational research for health policy: making better decisions, proceedings of the 31st meeting of the European working group on operational research applied to health services, 151-165.
[18] Vanhoucke, M. and Maenhout B. (2009). On the characterization and generation of nurse scheduling problem instances. European Journal of Operational Research, vol. 196, N.2, 457 - 467.
[19] Weil, G., K. Heus, P. Franois and M. Poujade (1995). Constraint Programming for Nurse Scheduling. IEEE Engeneering in Medicine and Biology, 14, 417-422.