International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 15 (2017) pp. 4959-4966

Non-liner Great Deluge Algorithm for Handling Nurse Rostering Problem

Yahya Z. Arajy*, Salwani Abdullah and Saif Kifah

Data Mining and Optimisation Research Group (DMO), Centre for Artificial Intelligence Technology, Universiti Kebangsaan Malaysia, 43600 Bangi Selangor, Malaysia.

Abstract

The optimisation of the nurse rostering problem is chosen in this work due to its importance as an optimisation challenge with an availability to improve the organisation in hospitals duties, also due to its relevance to elevate the health care through an enhancement of the quality in a decision making process. Nurse rostering in the real world is naturally difficult and complex problem. It consists on the large number of demands and requirements that conflicts the hospital workload constraints with the employees work regulations and personal preferences. In this paper, we proposed a modified version of great deluge algorithm (GDA). GDA usually explores neighbouring solutions which are accepted if they are better than the best solution so far or if the detriment in quality is no larger than the current water level. In the original GDA, the water level decreases steadily in a linear technique. In this work, we modified the decay rate of the water level into a non-linear fashion. Computational results were measured with three complexity levels as a total of 30 variant instances based on real-world constraints.

Keywords: nurse rostering problem, great deluge optimisation, real-world problems

INTRODUCTION

Health care is an important factor in real-life practice. Nurse rostering listed as one of the scheduling problems which are often come across a real world modern hospital's activities [1]. Nurse rostering consists in planning a periodical schedule for nurses by assigning daily duties for each nurse with satisfying a number of constraints. In order to obtain feasible schedule (roster) that can practically work, all hospital demands (hard constrains) must be fulfilled (i.e. assign all shifts required for a number of nurses) with trying to satisfy a varied requirements of the nurses (i.e. workload and personal demands) as much as possible. The purpose of solving this problem is to obligate the objectives of effectively utilize limited resources such that enhancing the hospitals' efficiency without losing the well-being and job satisfaction of nurses, thus healthcare will elevate through an enhancement of the quality in a decision making process. Along with the importance of the health care side, NRP recognised as a complex NP-hard problem [1] which it has a variant of requirements and real world constraints, gives the researchers a great scientific challenge to solve. The nurse restring problem is one of the most intensively exploring topics. Further reviewing of the literature can show that over the last 45 years a range of researchers studied the effect of different techniques and approaches on NRP. To understand and efficiently solve the problem, we refer to the comprehensive literate reviews, which a great collection of papers and summaries regarding this rostering problem can be found in [1], [2], and [3].

One of the first methods implemented to solve NRP is integer and mathematical programming. Warner and Prawda [4] presented a mathematical programming model for scheduling of nurses in a hospital. Thornton and Sattar [5] revisited nurse rostering and integer programming. Millar and Kiragu [6] attempted cyclic and non-cyclic scheduling using network programming. Moz and Pato [7, 8] applied an integer multi-commodity flow model, developed mathematical ILP formulations with additional constraints in a multi-level acyclical network, and used CPLEX solver. These techniques were used to guarantee the production of the most stable and desirable solutions. However, these techniques failed when utilized to solve a large number of constraints with a large number of instances. A significant deterioration in the quality of the solutions was observed when a vast search space problem (e.g., nurse rostering) was involved.

Researchers have applied heuristic/metaheuristic algorithms (e.g., evolutionary algorithms, VNS, simulated annealing, scattered search, and tabu search) in their attempts to solve NRP. Brusco and Jacobs [9] generated a cyclic schedule for continuously operating organizations. They combined simulated annealing metaheuristic with basic local search heuristic and applied the model on hospitals and other labor sectors continuously. Burke et al. [10] integrated the tabu search method with commercial techniques to solve NRP in Belgian hospitals. Burke et al. [11] also proposed a hybridization method to improve the quality of solutions. They applied a variable neighbourhood search method to search for new solutions by using heuristic ordering for repairing the schedule along with backtracking. The technique depended on two neighbourhood structures that successively assigned shifts to a free duty nurse and swapped shifts between two assigned nurses. Bellanti et al. [12] introduced a tabu search procedure and an iterated local search to solve NRP. Both hard and soft constraints are violated during the process, and various neighbourhood structures are operated on partial solutions. This procedure assists the search in working intensively to satisfy all constraints with each move; thus, avoiding the generation of infeasible rosters. Burke et al. [13] developed two hybrid tabu search algorithms, with diversification and greedy shuffling heuristics as presented in Bellanti et al. [12], who used the algorithms on a number of neighbourhoods to assign work shifts for a free duty nurse, swap shifts between two assigned nurses, and replace parts of the working schedule of nurses. The diversification method can be used to escape the current search space and allow a wild search in case the solution becomes trapped in local optima.

The present work focuses on the latest NRPs presented in the First International Nurse Rostering Competition (INRC-2010) organized by the International Series of conferences on the Practice and Theory of Automated Timetabling (PATAT-2010). INRC-2010 provided datasets with various complexity levels that incorporated a significant number of common real-world constraints [14].

A number of approaches have been anticipated by the participants of the competition in relation to this challenging and interesting problem. We briefly review the methods proposed by the INRC-2010 competition finalists.

Burke and Curtoiso [15] proposed two algorithms to solve NRP. The first algorithm, a variable depth search, is an ejection chain-based method. The algorithm was tested on short-term instances (sprint; 10-Seconds-timeframe) for the 2010 nurse rostering completion data. The algorithm initially uses a hill climbing strategy and employs different neighbourhoods. Burke and Curtoiso applied a dynamic programming algorithm in the constructive phase to build an individual schedule for each nurse. The second algorithm, a branch and price method, was applied based on the complexity of instances (medium and long). The algorithm embeds the pricing problem as a resource constrained shorted path problem. Similarly, Burke and Curtoiso applied the same technique as the first algorithm and solved NRP using a dynamic programming approach.

Valouxis et al. [16] attempted to solve NRP by applying a systematic two-phase approach. They partitioned the problem into sub-problems of manageable computational sizes. Integer mathematical programming was used to solve each sub-problem individually. Valouxis et al. systematically separated the daily workload assignment for each nurse in the first phase and scheduled specific shifts for the assigned nurse in the second phase. They also incorporated additional local optimisation search techniques to search across combinations of partial schedules for nurses.

Nonobe [17] used a metaheuristic algorithm by employing a general-purpose problem solver based on a tabu search approach for constraint problems (COP). In the present work, we only reformulated NRP as COP and implemented user-defined constraints. The constraint weights and tabu tenure was dynamically controlled during the search to enhance performance. The approach was applied on three tracks of the competition.

Bilgin et al. [18] developed a single approach to tackle three sets of NRP compaction tracks. They presented a hyper-heuristic approach combined with a greedy shuffle heuristic. NRP was modelled as a 0–1 matrix with random nurse assignments in such a way that all solutions were feasible. The proposal depended on the heuristic selection method and the move acceptance criterion (applied with simulated annealing). Bilgin et al. initially attempted to solve NRP with integer linear programming (ILP) using IBM CPLEX. The results obtained from different instances were varied in terms of complexity; in some areas, the hyper-heuristic approach outperformed the ILP (and vice versa).

Martin Josef [19] presented a variable neighbourhood search for personal nurse rostering. The technique consisted of two phases. The first phase was an initial constructive approach that met all shift requirements with guaranteed feasible solutions. A principle called most-constrained-first was used to assign nurses in the first phase. The second was an iterative improvement phase based on a VNS local search. After each round of VNS, a neighbourhood structure called perturb structure was applied as a diversified technique to escape the local optima trap.

In the present work, the non-liner great deluge algorithm (NLGDA) with three neighbourhood structures are employed (coded as *N1_move*, *N2_swap* and *N3_block-swap*) with tuned parameters that are based on our preliminary experiments to deal with NRP. The proposed algorithm is tested on INRC-2010 datasets.

The paper is organized as follows: Section 2 presents the problem description and formulation. Section 3 describes the NLGDA algorithm. Section 4 provides the experimental results. Section 5 presents the concluding remarks.

PROBLEM DESCRIPTION AND FORMULATION

NRP focuses on the accurate generation of a valid roster, which is represented by assigning shifts for each nurse [14]. The final roster must satisfy the various personal preferences and work regulations of nurses in the form of soft constraints. The terms used in this work are described below:

- **Roster:** a plan formulated for a number of days for one hospital ward.
- **Shift types:** a time frame for which a nurse with a certain skill is required.
- **Employees:** refer to the number of required nurses

provided for each day and shift type.

- **Schedule constraints**: a large number of items (i.e., the min/max amount of work, weekends, and night shifts) to which the schedules (sometimes called work patterns) of each employee are usually subjected to.

Constraints

The nurse rostering problem considered in this work involves of assigning duties to employees in practice with a given set of constraints namely Hard and Soft Constraints. Usually, two types of constraints are defined as:

Hard constraints

Hard constraints represent the requirements that must be satisfied to guarantee the usability of the roster. Solutions which satisfy hard constraints are called feasible solutions. Usually hospital and law requirements have described as a hard constraint form. The law requirements, either state or contract based, mostly represent the limited working hours and number of shifts that allowed assigning to a nurse. The hospital requirements state the compulsory coverage requested for each day to maintain the needed level of care quality. When the hard constraints are satisfied, we can have a feasible (useable) roster.

Soft constraints.

Soft constraints are aimed to increase the practical roster quality. Because not only a feasible roster can fulfil the actual need for the hospital, but also a satisfied workforce is desired to come up with efficient care demands. Soft constraints can be very diverse. Common soft constraints represent the desired preferences for each nurse like requests for free days, preferred to exclude certain shifts type in specific days, demanding day rest after night shifts and so on. The nurses are bound by a set of assignments called a contract. Each contract characterized by a number of regulations that should be fulfilled. Different contracts have different number of rules as it has different penalty values. Some soft constraints have special demands such as (i.e. unwanted patterns). Unwanted patterns are a sequence of shifts that unwanted to take place on certain days during the work routine. Other constraints can only be covered by employees with higher skill level (i.e. head nurse). Sometimes with the exception of the head nurse, we can sacrifice some constrains to improve the roster quality of other nurses. After all, the actual quality of a solution is measured in soft constraint violations. Table 1 shows the common real-world constraints for this problem.

Table 1: Description of constraints

Constraints	Description
H1	All demanded shifts must be assigned during the schedule planning
H2	A nurse can only work one shift per day (i.e., no nurse is allowed to work two shifts during the same day)
S1-2	The maximum/minimum number of shifts that should be assigned to nurse *s*
S3-4	The maximum/minimum number of consecutive working days set for nurse *s*
S5-6	The maximum/minimum number of consecutive free days set for nurse *s*
S7	The night shift type
S8-9	The maximum/minimum number of consecutive working weekends set for nurse *s*
S10	The maximum number of working weekends for nurse *s*
S11	Complete weekends; if nurse *s* is assigned to work during a weekend day, then nurse *s* should work for the entire weekend
S12	Identical shift types during the weekend; nurse s should be assigned the same shift *T* on the working weekend days
S13-14	Request On (off) if nurse *s* requests (not) to work any shift at day *d*
S15-16	Request On (off) if nurse *s* requests (not) to work a specific shift T at specific day *d*
S17	Alternative skill; certain shifts can only be set for a nurse if the nurse has all the required skills for that assignment
S18	The set of unwanted patterns of nurse *s*

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 15 (2017) pp. 4959-4966

We present a symbolic definition of the problem below to introduce hard and soft constraints:

- A set D of days, during which nurses are scheduled: $|D| = D$. D usually consists of four weeks (i.e., $D = 28$);
- A set S of nurses, each associated with a set of available skills and works under exactly one contract: $|S| = S$;
- A set T of shifts, each characterized by a set of required skills: $|T| = T$.

The constraints in Table 1 are variations of the penalty values based on contracts. Thus, each individual solution can have a different penalty value according to the working contract even if the nurses share the same soft constraints.

The evaluation function $f(X)$ for this problem is intended to sum up all penalties associated with the violation of soft constraints in the planning period as defined in Formula (1):

$$f(x) = \sum_{s=1}^{S} \sum_{n=1}^{18} y_{s,n} \cdot f_{s,n} \tag{1}$$

where X is the current solution and $y_{s,n}$ is the weight of the soft constraint S_n for nurse s regulated by the contract of nurse s. Each soft constraint (i.e., S1 to S18) has a different penalty value based on the contracts. If one of the soft constraints is disabled based on the contract regulation, then the value of $y_{s,n}$ will be set as 0.

Thus, the objective is to find a feasible solution X' such that $f(X') \le f(X)$ for all X in the feasible search space. Further mathematical clarifications are presented in the studies of Lü et al.[20] and Haspeslagh et al.[14].

Solution Space and Initial Solution

NRP is modelled as a 2D 0–1 matrix, in which the columns represent the shifts arranged per day and the rows represent the nurses. Figure 1 shows that a nurse is assigned to a particular shift on a specific day if the resultant matrix intersection element is 1. The initial solution is the selection of a nurse and assigning a shift that satisfies some of the soft constraints. In case of failure, a shift is randomly assigned to another nurse based on feasibility. The coverage should be met and no nurse can work more than one shift per day. Feasibility is maintained during the subsequent search by considering only assignment moves within the same column. No assignment can be removed without making a new one within the same column.

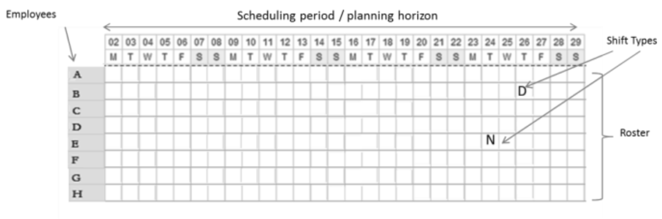

Figure 1: Planning roster

Neighbourhood Structures

Our local search procedure uses three distinct neighbourhood structures: s*ingle-move, single-swap,* and *block-swap,*

- ***N1_move*:** A shift is randomly transferred from a selected nurse on duty to any other free nurse on the same day (Fig. 2).
- ***N2_swap*:** For a single day, two shifts assigned to a pair of nurses are swapped on the same day, unless they are pre-assigned with the same shift type.
- ***N3_block-swap*:** For a selected number of days in a row (e.g., 2–4 days), selected shifts assigned to a pair of nurses are swapped on the same day (Fig. 3).

During any kind of movement, feasibility is always maintained with no violation of the hard constraints.

Figure 2: Move neighbourhood structure

Figure 3: Swap neighbourhood structure

Non-linear Great Deluge Approach (NLGDA)

Great deluge algorithm was developed by Dueck [21]. The basic idea behind this approach is exploring the neighboring region of search space for a better solution than the current one. If not, the new solution is accepted only if its fitness function is less than a pre-defined water level which declines linearly as the search progresses. The water level determines the speed of the search that controlled by decay rate value. The higher the decay rate the faster the water level goes down and the faster the algorithm terminates. Burke et al. [22] proposed to initialise the water level B_0 equal to the fitness function value of the initial solution. The decay rate at each iteration stay constant and they interpreted the parameter as a function of the expected search time and the expected solution quality. To calculate the decay rate ΔB, they first estimate the desired value DV for the best solution quality $f(s)$ and then calculate $\Delta B = B_0 - DV/Number\ of\ iterations.$

In our this work, we proposed a great deluge approach in which the decay rate of the water level is non-linear and is determined by the following expression:

$$\Delta B = f(s)_{current} - DV / I_{remaining} \qquad (2)$$

$$B = B - \Delta B \qquad (3)$$

The parameters in Eq. (2) control the speed and the shape of the water level decay rate. Therefore, the higher the values of current solution the faster the water level decreases. In return, the improvement is quickly achieved but it will suffer from this greediness by trapping itself in local optima. To equalize this greediness, re-initialising water level is necessary after a number of non-improved iterations controlled by parameter α . Then, in addition to using a non-linear decay rate for the water level B, we also allow B to go up when its value is about to converge with the penalty cost of the candidate solution. We increase the water level B by a controlled parameter γ. As the pseudo code of the proposed algorithm is represented in Named as NLGDA–NRP, it is shown in the following pseudo code:

```
Algorithm NLGDA-NRP

Construct initial feasible solution S
Set best solution so far Sbest ← S
Set Desired Value DV
Set Maximum number of  Iterations  I
Set initial water level B ← f(S)
Set Decay rate factor ΔB = f(s)_current – DV / I_remaining
Set water level increase factor γ
Set total number of non-improved iterations α
Set timeLimit according to problem

while elapsedTime ≤ timeLimit do
    Select move at random from neighborhoods N1,N2,N3 (see section 2.3)
    Apply the neighborhood on solution s and calculate f(s*)
    if ( f(S*) ≤ f(S) or f(S*) ≤ B ) then
        S ← S* {accept new solution}
        if ( f(S*) ≤ f(S)) then
            Sbest ← S {update best solution}
        end if
    end if
    B = B –  ΔB
    if (α is met) then
        B = B + (B*γ)
    end if
end while
```

EXPERIMENT RESULTS

The excremental environment for the proposed method was implemented in Java, and simulations were performed on 2.8 GHz CPU with 8 GB RAM. The tested datasets can be downloaded from https://www.kuleuven-kulak.be /nrpcompetition /instances-results. For each dataset, the algorithm was run for 10 Seconds based on the competition rules time. Each instance was tested 30 times.

The details of the parameter values conducted in the experiment are presented below. The parameters were tuned with a simple iterated search technique and without any special tuning tool.

- α =1500
- ɣ = 0.07

COMPARISON RESULTS

Table 2 shows the computational statistics of the proposed NLGDA–NRP. The second column shows the best known results for these datasets [20]. The rest of the table presents the best results obtained by NLGDA–NRP, the average function value, the best time needed to obtain the results, and the number of iterations.

NLGDA–NRP can obtain one best result on 17 datasets compared to the best known results in literature. Thus, NLGDA –NRP can reach most of the lower bound values under the competition time provided.

Table 2. NLGDA –NRP computational results

Dataset	Best known	NLGDA–NRP	Avr.	T_best (sec)	Itr
sprint01	56	56	56.07	0.147	9821
sprint02	58	58	58.20	0.17	12372
sprint03	51	51	51.57	0.154	10635
sprint04	59	59	59.70	0.476	28084
sprint05	58	58	58.00	0.123	7939
sprint06	54	54	54.10	0.151	9961
sprint07	56	56	56.23	0.279	19008
sprint08	56	56	56.40	0.102	6744
sprint09	55	55	55.37	0.333	21876
sprint10	52	52	52.23	0.125	8918
sprint_late01	37	38	40.90	0.605	28865
sprint_late02	42	43	46.17	0.311	19711
sprint_late03	48	49	51.50	0.36	16281
sprint_late04	73	83	94.53	0.236	10901
sprint_late05	44	45	46.20	0.32	14614
sprint_late06	42	42	43.13	0.133	14011
sprint_late07	42	53	58.30	0.111	11932
sprint_late08	17	17	22.17	0.076	8325
sprint_late09	17	17	21.93	0.051	5611
sprint_late10	43	49	59.70	0.084	8497
sprint_hidden01	32	32	37.83	0.289	19106
sprint_hidden02	32	32	37.57	0.267	18185
sprint_hidden03	62	63	69.43	0.631	30470
sprint_hidden04	66	66	69.40	0.364	18157
sprint_hidden05	59	59	66.07	0.138	6141
sprint_hidden06	130	169	195.30	0.212	13657
sprint_hidden07	153	173	210.97	0.143	7997
sprint_hidden08	204	233	272.27	0.161	7175
sprint_hidden09	338	348	388.90	0.263	12294
sprint_hidden10	306	320	377.10	0.215	9460

In this section, we compare our proposed algorithm with other INRC- 2010 Competition finalists, also to the work presented by Lü and Hao (2012) since this authors did manage to reduce the best known values in twelve instances. Table 3 presents the best results obtained by the NLGDA-NRP, (Lü and Hao, [20]), two-phase hybrid method the competition winner (Valouxis et al., [16]) and four other competitors from the finales of INRC-2010 competition on the 30 competition instances. A brief description was given in Section 1 about the methods been used by the competition finalists. Referenced by, general

constraint optimization solver by Nonobe [17], a hyper-heuristic algorithm by Bilgin et al. [18], an ILP algorithm using ILOG CPLEX also by Bilgin et al. [18], and a branch and price algorithm by Burke and Curtois [15]. Note that the results in the last column marked with "*" are proven to be optimal, and the mark "-" means no solution produced by the method.

We believe this is the first work that provides a study and development of a NLGDA algorithm on large scale nurse rostering problems in the literature. The proposed algorithm was able to reach the best known solution for nearly 57% of the tested datasets. The best results obtained by the NLGDA_NRP are still distant from some other algorithms in the literature, and the algorithm by now at a disadvantage position in comparison with the other methods. We believe if more interest given to this proposed approach, in term of neighborhood structures and parameter efficiency could be a key to a successful metaheuristic solver for general optimization problems.

CONCLUSION

In this study, we discussed the use of the non-liner great deluge algorithm (NLGDA) algorithm to solve nurse rostering problem. NLGDA is a single based solution algorithm that depends on a water level to accept new solutions. The overall goal was to extend the algorithm using a non-liner method to update the water level with the use of three neighbourhood structures. A total of 30 well-known datasets from INRC-2010 demonstrated the strength of the algorithm. Depending on the excited elements of the algorithm 17 instances for the best known solutions in literature was identified and matched. Our future work aims to examine different datasets with various complexities. The solutions can be further improved by enhancing the neighbourhood structures through advanced parameter-tuning methods.

Table 3. Comparison with other methods

Dataset	BKS	NLGDA–NRP	Lü and Hao	Winner	Burke	Nonobe	Bilgin	ILP
sprint01	56	56	56	56	56	56	56	56*
sprint02	58	58	58	58	58	58	58	58*
sprint03	51	51	51	51	51	51	51	51*
sprint04	59	59	59	59	59	59	59	59*
sprint05	58	58	58	58	58	58	58	58*
sprint06	54	54	54	54	54	54	54	54*
sprint07	56	56	56	56	56	56	56	56*
sprint08	56	56	56	56	56	56	56	56*
sprint09	55	55	55	55	55	55	55	55*
sprint10	52	52	52	52	52	52	52	52*
sprint_late01	37	38	37	37	37	37	37	39
sprint_late02	42	42	42	42	42	42	42	43
sprint_late03	48	48	48	48	48	48	48	54
sprint_late04	75	78	73	76	73	76	76	99
sprint_late05	44	44	44	44	44	45	45	47
sprint_late06	42	42	42	42	42	42	42	42*
sprint_late07	42	44	42	43	42	43	43	42*
sprint_late08	17	17	17	17	17	17	17	21
sprint_late09	17	17	17	17	17	17	17	35
sprint_late10	43	46	43	44	43	44	44	43*
sprint_hidden01	33	33	32	33	–	–	–	–
sprint_hidden02	32	32	32	33	–	–	–	–

sprint_hidden03	62	62	62	62	–	–	–	–
sprint_hidden04	67	67	66	67	–	–	–	–
sprint_hidden05	59	60	59	60	–	–	–	–
sprint_hidden06	134	136	130	139	–	–	–	–
sprint_hidden07	153	156	153	153	–	–	–	–
sprint_hidden08	209	226	204	220	–	–	–	–
sprint_hidden09	338	353	338	338	–	–	–	–
sprint_hidden10	306	314	306	306	–	–	–	–

REFERENCES

[1] Burke, E.K., De Causmaecker, P., Vanden Berghe, G.: Novel meta-heuristic approaches to nurse rostering problems in Belgian hospitals. Handbook of Scheduling: Algorithms, Models and Performance Analysis, pp. 44.41–44.18 (2004)

[2] Ernst, A.T., Jiang, H., Krishnamoorthy, M., Sier, D.: Staff scheduling and rostering: A review of applications, methods and models. European journal of operational research 153, 3-27 (2004)

[3] Cheang, B., Li, H., Lim, A., Rodrigues, B.: Nurse rostering problems—a bibliographic survey. European Journal of Operational Research 151, 447-460 (2003)

[4] Warner, D.M., Prawda, J.: A mathematical programming model for scheduling nursing personnel in a hospital. Management Science 19, 411-422 (1972)

[5] Thornton, J., Sattar, A.: Nurse rostering and integer programming revisited. In: International Conference on Computational Intelligence and Multimedia Applications, pp. 49-58. (1997)

[6] Millar, H.H., Kiragu, M.: Cyclic and non-cyclic scheduling of 12 h shift nurses by network programming. European journal of operational research 104, 582-592 (1998)

[7] Moz, M., Pato, M.V.: Solving the problem of rerostering nurse schedules with hard constraints: new multicommodity flow models. Annals of Operations Research 128, 179-197 (2004)

[8] Moz, M., Vaz Pato, M.: A genetic algorithm approach to a nurse rerostering problem. Computers & Operations Research 34, 667-691 (2007)

[9] Brusco, M.J., Jacobs, L.W.: Cost analysis of alternative formulations for personnel scheduling in continuously operating organizations. European Journal of Operational Research 86, 249-261 (1995)

[10] Burke, E., De Causmaecker, P., Berghe, G.V.: A hybrid tabu search algorithm for the nurse rostering problem. Simulated evolution and learning, pp. 187-194. Springer (1999)

[11] Burke, E., De Causmaecker, P., Petrovic, S., Berghe, G.V.: Variable neighborhood search for nurse rostering problems. Metaheuristics: computer decision-making, pp. 153-172. Springer (2004)

[12] Bellanti, F., Carello, G., Della Croce, F., Tadei, R.: A greedy-based neighborhood search approach to a nurse rostering problem. European Journal of Operational Research 153, 28-40 (2004)

[13] Burke, E.K., Curtois, T., Post, G., Qu, R., Veltman, B.: A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem. European Journal of Operational Research 188, 330-341 (2008)

[14] Haspeslagh, S., De Causmaecker, P., Schaerf, A., Stølevik, M.: The first international nurse rostering competition 2010. Annals of Operations Research 1-16 (2012)

[15] Burke, E.K., Curtois, T.: New approaches to nurse rostering benchmark instances. European Journal of Operational Research 237, 71-81 (2014)

[16] Valouxis, C., Gogos, C., Goulas, G., Alefragis, P., Housos, E.: A systematic two phase approach for the nurse rostering problem. European Journal of Operational Research 219, 425-433 (2012)

[17] Nonobe, K.: INRC2010: An approach using a general constraint optimization solver. The First International Nurse Rostering Competition (INRC 2010) (2010)

[18] Bilgin, B., De Causmaecker, P., Rossie, B., Berghe, G.V.: Local search neighbourhoods for dealing with a novel nurse rostering model. Annals of Operations Research 194, 33-57 (2012)

[19] Martin, J.G.: Personnel rostering by means of variable neighborhood search. Operations Research Proceedings 2010, pp. 219-224. Springer (2011)

[20] Lü, Z., Hao, J.-K.: Adaptive neighborhood search for nurse rostering. European Journal of Operational Research 218, 865-876 (2012)

[21] Dueck G., "New optimization heuristics: The great deluge algorithm and the record-to-record travel," *Journal of Computional Physics*, vol. 104, pp. 86–92, (1993)

[22] Burke, E.K., Bykov, Y., Newall, J., Petrovic, S.: A Time-predefined Approach to Course Timetabling. Yugoslav Journal of Operations Research (YUJOR) 13(2),139–151 (2003)