2013 IEEE International Conference on Systems, Man, and Cybernetics

An Ant Colony Optimization Approach For Nurse Rostering Problem

Jie-jun Wu, Ying Lin (corresponding author),
Zhi-hui Zhan, Wei-neng Chen, and Ying-biao Lin
Department of Computer Science
Sun Yat-sen University
Guangzhou, China 510006
yinglinchn@gmail.com

Jian-yong Chen
Department of Computer Science and Technology
Shenzhen University
Shenzhen, China 518060
desiredjychen@szu.edu.cn

Abstract—**Nurse rostering is a non-deterministic polynomial problem with many constraints. In the literature, a number of heuristic approaches have been proposed, but few of them can achieve satisfying performance on both solution quality and search speed. Inspired by the successful experience of ant colony optimization (ACO) on many highly-constrained problems, this paper proposed an ant colony optimization approach termed ACO-NR for solving the nurse rostering problem. First, the search space of the nurse rostering problem is remodeled as a graph, with each solution corresponding to a path on the graph. Then a heuristic function is designed to guide the path construction behavior of ACO-NR. The heuristic information comes not only from the static information defined by the problem-dependent knowledge, but also from the dynamic information generated by the solution construction procedure. A penalty function is defined to help ACO-NR handle problem constraints. Experimental results on 52 benchmark instances show that the proposed ACO-NR can achieve better performance than classic nurse rostering algorithms.**

Keywords-component; Constrained optimization; evolutionary algorithm; ant colony optimization; nurse rostering

I. Introduction

Nurse rostering is an important personnel scheduling problem that is faced by many large hospitals across the world. The problem involves producing daily schedules for nurses over a given time horizon. The objectives are to improve the hospitals' efficiency, to balance the workload among nurses and, more importantly, to satisfy various hard constraints, and as many soft constraints as possible, such as minimal nurse demands, "day-off" requests, personal preferences, etc. Depending on the practical situations and requirements in different hospitals, the type and the number of constraints can be varied. Due to these constraints, the solution search space of nurse rostering problems is highly constrained with feasible regions usually disconnected. Besides, it has been proven that nurse rostering problem is among the class of non-polynomial-time hardness (NP-hard). that is, no algorithms can surely find an optimal solution in polynomial time unless P = N. In this case, instead of using high computational efforts to find the exact optimum, heuristic algorithms that can find near-optimal solutions in considerably less time is more preferred. In literature, considerable research has been carried out in this area, such as genetic algorithm and its improved version. However, these approaches either cost too much computation, or have difficulty in constraint handling.

Ant colony optimization (ACO) is a meta-heuristic inspired from the foraging behaviors of ants in nature[1]. In ACO, each ant is a stochastic constructive procedure that builds a solution by walking on a solution construction graph representing the search space. The ants' walk is guided by two factors: pheromone and heuristic information. Pheromone records the searching experience of ants, while heuristic information reflects problem-specific knowledge. The constructive search behavior of ants facilitates ACO in finding feasible solutions that satisfy problem constraints. Historical information and problem-specific knowledge can guide ACO to find good solutions efficiently. Therefore, ACO has been widely applied to complex practical problems in the fields of engineering[2][3][4], project scheduling[5][6][7] and education[8], and data mining[9][10][11], among others. Inspired by the successful applications of ACO in many highly-constrained problems, this paper proposes an ACO approach for solving the nurse rostering problem.

The remainder of this paper is structured as follows. Section II presents the nurse rostering problem that is addressed in this paper. Section III presents the ant colony optimization for the problem (ACO-NR), along with the heuristic design and constraints handling. Section IV presents the experimental result comparing with other algorithms. Section V concludes the paper.

II. The Nurse Rostering Problem

Nurse rostering has been widely studied in recent years, and a general overview of various approaches can be found in[12][13][14][15]. The nurse rostering problem we consider in this paper is to create weekly schedules in a large U.K. hospital[16]. The schedules must be seen "fair" by the staff, which means unattractive patterns or weekend work has to be evenly spread. To achieve this all patterns carry a "preference cost" (from zero=perfect to 100) and the objective is to minimize this. Furthermore, all schedules have to conform to a number of constraints regarding demand and skill levels. Full details can be found in[16]. Typical problem dimensions are 30 nurses, three grade levels and over 400 theoretic weekly shift patterns per nurse. The integer program model of the problem is shown below.

Decision variables:
$x_{ij}=1$ if nurse i works shift pattern j, 0 otherwise.
Parameters:
$m=$ Number of possible shift patterns;
$n=$ Number of nurses;
$p=$ Number of grades;
$a_{jk}=1$ If shift pattern j covers day/night k, 0 otherwise.
$q_{ig}=1$ If nurse i is of grade g or higher, 0 otherwise;
$c_{ij}=$ Preference cost of nurse i working shift pattern j;
$R_{kg}=$ Demand of nurses with grade g on day/night k;
Objective function:

$$Minimize \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}x_{ij} \tag{1}$$

Subject to:

$$\sum_{j=1}^{m} x_{ij}=1, \forall i\in\{1,\cdots,n\} \tag{2}$$

$$\sum_{j=1}^{m}\sum_{i=1}^{n} q_{ig}a_{jk}x_{ij}\ge R_{kg}, \quad \forall k\in\{1,\cdots,14\}, g\in\{1,2,3\} \tag{3}$$

The goal of the objective function (1) is to minimize the total preference cost of all nurses. Constraint (2) ensures that every nurse works exactly one shift pattern from his/her feasible set, and constraint (3) ensures that the demand for nurses is fulfilled for every grade on every day and night. This problem can be regarded as a multiple-choice set-covering problem, in which the sets are given by the shift pattern vectors and the objective is to minimize the cost of the sets needed to provide sufficient cover for each shift at each grade. The constraints described in (2) enforce the choice of exactly one pattern (set) from the alternatives available for each nurse.

III. Ant Colony Optimization for Nurse Rostering

The design of ACO-NR requires settlement of four issues: (1) constraints handling strategy; (2) the construction procedure of ants; (3) design of heuristic information; (4) design of pheromone management scheme.

A. Definition of Penalty Function

Penalty functions are among the most popular techniques for constraints handling, and have been widely used in many applications [17][18][19][20][21]. The idea is to transform the constrained optimization problem into an unconstrained one by introducing a penalty term into the objective function to penalize constraint violations. Let *x* be the vector of decision variables and *f(x)* be the original objective function. The transformed objective function $\phi(x)$ is often presented in the form of

$$\phi(x)=f(x)+\lambda\varphi(g_\pi(x);\pi\in\Pi) \tag{4}$$

Where λ is the associated penalty coefficient and $\varphi(g_\pi(x))$ is a function that measures the severity of violations of the following constraints

$$g_\pi(x)\ge 0, \pi\in\Pi \tag{5}$$

In the case of the nurse rostering problem addressed in this paper, the following function can be used to measure the violation of the covering constraints (3)

$$\varphi(g_\pi(x))=\sum_{k=1}^{14}\sum_{g=1}^{3}\{\max\{0,R_{kg}-\sum_{i\in G_g}\sum_{j\in F_i}a_{jk}x_{ij}\}\} \tag{6}$$

B. The Construction Procedure of Ants

Let n be the number of nurses, F_i ($i=1,2,\cdots n$) be the set of feasible shift patterns of nurse i. We consider every element in F_i as a node in the construction graph. Besides, the construction graph has an additional node v_0 to represent the source of ant routes. The set of edge is denoted as $U=\{(v_i,v_{i+1}): i=0 \ or\ v_i\in F_i; i=1,2,\cdots,n-1\}$.

Based on the construction graph defined above, ACO-NR defines a solution x as a sequence of n elements, $x=v_1v_2\cdots v_n (v_i\in F_i)$ can be considered as a "trip" on the construction graph $v_0v_1\cdots v_n$ with n edges.

An example of construction graph is illustrated by Figure. 1. It shows a NRP with 5 nurses to be scheduled. The nodes below F_i represent the feasible shift patterns can be chosen by nurse *i*. Nurse 3 has 2 feasible shift patterns, nurse 4 has 4, the other has 3 feasible shift patterns. Besides, we add node to represent the source of ant routes. The solid and dotted lines denote to possible solutions built by ants.

Each ant builds a solution through the following steps:

Step 1) Let X to be an empty sequence, $i\leftarrow 1$

Step 2) If X is empty, $r\leftarrow v_0$, else $r\leftarrow x_{i-1}$

Step 3) For each node v_{ij} in F_i, calculate their heuristic score η_{ij} (the approach will be described in next section)

Step 4) For each node v_{ij} in F_i, their possibility to be chosen is calculate by the formulation

$$p_{ij}=\frac{\left[\tau_{r,v_{ij}}\right]^\alpha\cdot\left[\eta_{ij}\right]^\beta}{\sum_{s=1}^{|F_i|}\left[\tau_{r,v_{is}}\right]^\alpha\cdot\left[\eta_{is}\right]^\beta} \tag{7}$$

where $\tau_{r,v_{ij}}$ is the pheromone accumulated on the edge from r to v_{ij}, α and β are the parameters of ACO.

Step 5) Apply a *pseudorandom proportional rule to* Choose a node as x_i from F_i. In detail, the pseudorandom proportional rule is implemented as follows:

$$x_i=\begin{cases}\arg\max\limits_{v_{ij}\in F_i} p_{ij}, & if\ r<q_0\\ x, & otherwise\end{cases} \tag{8}$$

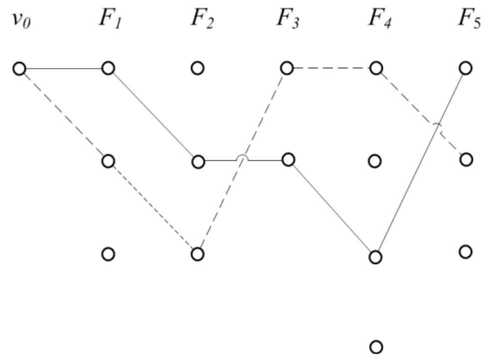

Figure 1. Example of a construction graph

where r is a random number uniformly distributed in $(0,1)$, q_0 is a parameter $(0 < q_0 < 1)$, which allows to give more emphasis to exploitation or exploration, and x is a random variable selected according to the probability distribution calculated in Step 4). Add x_i to x, and $i \leftarrow i+1$.

Step 6) If $i <= n$, return to Step 2). Otherwise, every nurse has chosen a pattern, a complete solution is built.

C. Design of Heuristic Information

When choose shift pattern for nurse i from F_i, ACO_NR defines two kinds of heuristic information. The heuristic value of node v_{ij} can be expressed by

$$\eta_{ij} = \eta_{ij}^s \cdot \eta_{ij}^d \tag{9}$$

η_{ij}^s is static heuristic information which denotes the contribution from the cost of nurse i work on the shift pattern v_{ij}. The word "static" here means this heuristic value can be calculated at the beginning of the search procedure and remains unchanged afterwards.

$$\eta_{ij}^s = (1+c_{ij})^{-1} \tag{10}$$

η_{ij}^d is a dynamic heuristic information that denotes the contribution towards the reduction in shortfall of qualified nurses for nurse i work on shift pattern v_{ij}.

In (9), η_{ij}^d is calculated as the weighted sum of uncovered shifts on grade one, two and three after ($i-1$) nurses have chosen their shift patterns that would be covered if the nurse i works on shift pattern v_{ij}. It can be formulated as:

$$\eta_{ij}^d = \sum_{g=1}^{3} w_g q_{ig} (\sum_{k=1}^{14} a_{jk} d_{kg}) \tag{11}$$

q_{ig} and a_{jk} use the same definitions as in the inequality represented in (3), d_{kg} is the shortage number of nurses during period k of grade g. w_g is a parameter to emphasis the demand of higher grade is harder to be satisfied, as a higher grade nurse can cover the demand for a lower grade nurse but not vice versa.

D. Deign of Pheromone Management Scheme

In ACO-NR, pheromone is laid on the arcs between feasible shift patterns from two "adjacent" nurses, the word adjacent means the connection on the order of nurse to be considered, when construct a solution. Initially, all pheromone trials are uniformly set as $\tau_0 = (\phi_0 + 1)^{-1}$, where ϕ_0 is the fitness of a solution generated by heuristic information only. A local pheromone updating rule is applied to renew the pheromone trial τ_{x_{i-1},x_i} on the arc (x_{i-1}, x_i) after an ant build the current solution x. Specifically, the pheromone τ_{x_{i-1},x_i} is renewed by

$$\tau_{x_{i-1},x_i} = (1-\varepsilon)\tau_{x_{i-1},x_i} + \varepsilon\tau_0 \tag{12}$$

where $\varepsilon \in (0,1)$ is a parameter. After all the ants finish building their own solutions, a global pheromone updating rule is applied to renew pheromone trials as follows:

$$\tau_{wl} = \begin{cases} (1-\rho)\tau_{wl} + \rho(1+\phi^{bsf})^{-1}, & if\,(v_w, v_l) \in x^{bsf} \\ \tau_{wl} & otherwise \end{cases} \tag{13}$$

where ρ is a parameter and ϕ^{bsf} is the fitness of the best-so-far solution x^{bsf}

E. Overall Procedure of ACO-NR

As a summary, the overall procedure of ACO-NR is presented as below.

```
ACO-NR

read in data and initialize the construction graph.
construct a solution x^0 using static heuristic information only
x^bfs ← x^0
apply a local pheromone update using x^0
while terminate constraints hasn't be satisfied do
   for every ant do
      construct a solution x
      evaluate x
      if x is better than x^bfs then x^bfs ← x
      apply a local pheromone update using x^bfs
   apply a global pheromone update using x^bfs
return x^bfs
```

IV. EXPERIMENTAL RESULTES

This section describes the computational experiments of ACO-NR. In the experiments, the 52 benchmark instances [16] are used. Each instance consists of one week requirements for all shift and grade combinations and a list of available nurses together with their preference costs and qualifications.

In addition to the ILP approach[16], a variety of recently developed meta-heuristic techniques have also used the 52 instances as a benchmark test-bed. The meta-heuristic approaches include a basic GA, an adaptive GA, a multi-population GA, a hill-climbing GA (all from[21], an indirect

GA[22], an estimated distribution algorithm [23] and a learning classifier system[24]), among which the indirect GA(IGA) and hill-climbing GA(HGA) perform the best, we compare ACO-NR with these two approaches.

All of these approaches use the following same objective function.

$$Minimize\sum_{i=1}^{n}\sum_{j=1}^{m}c_{ij}x_{ij}+\lambda\sum_{k=1}^{14}\sum_{g=1}^{3}\{\max\{0,R_{kg}-\sum_{i\in G_g}\sum_{j\in F_i}a_{jk}x_{ij}\}\} \quad (13)$$

The stopping criterions is set as no improvement of objective function for 1000 iterations or a known optimal solution has been found. The parameters are set as follows: the α and β in (7) is set to $\alpha=1,\beta=2$; ε and ρ in (12) and (13) is set to $\varepsilon=\rho=0.1$, w_1 , w_2 and w_3 in (11) is set to $\{8,2,1\}$,the q_0 in (8) is set to 0.9. The penalty λ in function (13) is 200.

Table I lists the optimal results found by a commercial ILP solver called XPRESS MP[22]. It also lists the comparative results of the ACO-NR against IGA and HGA, based on 20 runs of each data instance using different random number seeds. These comparative results include the best, the mean and the p-value of t-test. The p-value indicates the significance of the hypothesis that, on average, the ACO-NR generates better solutions than the others approach for a given data instance at a 95% confidence level (p=0.05).

From Table I we can see how the ACO-NR performs as well as ILP approach (with 47 of 52 instances being optimal and the rest near-optimal) and how ACO-NR obtains better results than IGA and HGA. Compared with the HGA, the ACO-NR performs better in 51 out of 52 instances, in which the difference is statistically significant for 43 instances. Compared with IGA, for 38 of 52 instances the performance of ACO-NR is statistically better. Out of the remaining 14 instances, in seven of them (i.e., instances 01, 04, 05, 06, 12, 14 and 25) the performance of the ACO-NR is 100% optimal or very nearly optimal (with only 1 of 20 runs being no optimal), thus there is no difference between ACO-NR and IGA. For The five instances 0f 21,24,29,45, and 52, the performance of ACO-NR is better but the difference is not statistically significant. The proposed ACO-NR is slightly outperformed by IGA in only 2 out of 52 instances.

V. CONCLUSION

This paper proposes a new ant colony optimization technique approach to solve nurse rostering problems. Experiments show that ACO-NR significantly outperforms two of the best nurse rostering algorithms, IGA and HGA.

The ACO-NR can be treated as a frame work to use ACO to solve rostering problems. The techniques which improve ACO can also be used in this algorithm, such as pheromone update scheme and local search. As the same reason, other approaches of constraints handling can be import to ACO-NR easily. For future work, we will try to improve the performance of ACO-NR on these two aspects.

TABLE I. COMPARISON RESULTS FROM ACO-NR, IGA, HGA

Data	*ILP*	*HGA Best*	*HGA Mean*	*IGA Best*	*IGA Mean*	*ACO Best*	*ACO Mean*	*p-value HGA*	*p-value IGA*
01	8	8	10	8	8	8	8	0.008	0.153
02	49	50	52	51	60	52	56	0.000	0.004
03	50	50	52	51	61	50	51	0.105	0.000
04	17	17	18	17	17	17	17	0.023	1.000
05	11	11	13	11	11	11	11	0.110	0.324
06	2	2	5	2	2	2	2	0.026	0.075
07	11	13	139	12	22	11	12	0.000	0.000
08	14	14	16	15	19	14	14	0.003	0.000
09	3	3	40	4	6	3	4	0.013	0.031
10	2	2	9	3	4	2	3	0.000	0.000
11	2	2	5	2	3	2	2	0.005	0.027
12	2	2	26	2	2	2	2	0.010	1.000
13	2	2	3	2	2	2	2	0.028	0.768
14	3	3	5	3	8	3	3	0.020	0.000
15	3	3	6	3	5	3	4	0.066	0.000
16	37	38	98	39	44	37	38	0.000	0.000
17	9	9	22	10	15	9	9	0.000	0.001
18	18	19	68	18	20	18	19	0.001	0.000
19	1	1	15	1	2	1	1	0.021	0.033
20	7	8	41	7	15	7	8	0.009	0.000
21	0	0	10	0	0	0	0	0.011	1.000
22	25	25	32	25	26	25	26	0.022	0.846
23	0	0	48	0	1	0	0	0.025	0.000
24	1	1	84	1	1	1	1	0.005	0.154
25	0	0	3	0	0	0	0	0.139	0.324
26	48	48	204	48	132	48	49	0.000	0.001
27	2	2	30	4	14	2	3	0.003	0.000
28	63	63	68	64	150	63	64	0.081	0.000
29	15	17	216	15	27	15	16	0.000	0.365
30	35	35	42	38	41	35	37	0.398	0.000
31	62	161	237	65	77	65	70	0.000	0.000
32	40	41	90	42	46	40	41	0.000	0.000
33	10	12	45	12	15	10	11	0.046	0.000
34	38	40	107	39	44	38	40	0.000	0.000
35	35	35	36	36	44	35	36	0.002	0.000
36	32	33	77	32	37	32	34	0.006	0.020
37	5	5	10	5	9	5	6	0.083	0.000
38	13	16	96	15	19	13	15	0.000	0.000
39	5	5	8	5	8	5	6	0.286	0.028
40	7	8	16	7	10	7	8	0.003	0.000
41	54	54	109	55	68	54	54	0.002	0.000
42	38	38	48	39	44	38	39	0.001	0.005
43	22	24	99	23	24	22	22	0.000	0.000
44	19	48	117	25	31	19	21	0.000	0.000
45	3	3	5	3	5	3	4	0.040	0.193
46	3	6	59	6	8	3	5	0.002	0.000
47	3	3	9	3	4	3	3	0.000	0.006
48	4	4	12	4	9	4	5	0.004	0.000
49	27	29	104	30	233	28	31	0.000	0.000
50	107	110	223	110	241	108	110	0.000	0.000
51	74	75	246	74	93	74	107	0.000	0.373
52	58	75	169	58	67	59	65	0.000	0.293
mean	21	24	63	22	36	21	23	n/a	n/a

ACKNOWLEDGMENT

This work was partially supported by the National High-Technology Research and Development Program ("863" Program) of China under Grand No. 2013AA01A212, by the National Science Fund for Distinguished Young Scholars under Grant 61125205, by the National Natural Science Foundation of China under Grant 61070004 and 61202130, by the NSFC Joint Fund with Guangdong under Key Project U1201258 and U1135005.

REFERENCES

[1] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 26, pp. 29-41, 1996.

[2] Y. Lin, J. Zhang, H.-H. Chung, W. H. Ip, Y. Li, and Y.-H. Shi, "An ant colony optimization approach for maximizing the lifetime of heterogeneous wireless sensor networks," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol. 42, pp. 408-420, 2012.

[3] Z.-H. Zhan, J. Zhang, Y. Li, O. Liu, S. Kwok, and W. Ip, "An efficient ant colony system based on receding horizon control for the aircraft arrival sequencing and scheduling problem," Intelligent Transportation Systems, IEEE Transactions on, vol. 11, pp. 399-412, 2010.

[4] J. Zhang, H.-H. Chung, A.-L. Lo, and T. Huang, "Extended ant colony optimization algorithm for power electronic circuit design," Power Electronics, IEEE Transactions on, vol. 24, pp. 147-162, 2009.

[5] W. Chen and J. Zhang, "Ant Colony Optimization for Software Project Scheduling and Staffing with an Event-Based Scheduler," 2013.

[6] W.-N. Chen and J. Zhang, "An ant colony optimization approach to a grid workflow scheduling problem with various QoS requirements," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol. 39, pp. 29-43, 2009.

[7] W.-N. Chen, J. Zhang, H.-H. Chung, R.-Z. Huang, and O. Liu, "Optimizing discounted cash flows in project scheduling—An ant colony optimization approach," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol. 40, pp. 64-77, 2010.

[8] X.-M. Hu, J. Zhang, H.-H. Chung, O. Liu, and J. Xiao, "An intelligent testing system embedded with an ant-colony-optimization-based test composition method," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol. 39, pp. 659-669, 2009.

[9] Juan Luis Olmo, José Raúl Romero, Sebastián Ventura: Using Ant Programming Guided by Grammar for Building Rule-Based Classifiers. IEEE Transactions on Systems, Man, and Cybernetics, Part B 41(6): 1585-1599 (2011)

[10] Fernando E. B. Otero, Alex Alves Freitas, Colin G. Johnson: A New Sequential Covering Strategy for Inducing Classification Rules With Ant Colony Algorithms. IEEE Trans. Evolutionary Computation 17(1): 64-76 (2013)

[11] Juan Luis Olmo, José María Luna, José Raúl Romero, Sebastián Ventura: Mining association rules with single and multi-objective grammar guided ant programming. Integrated Computer-Aided Engineering 20(3): 217-234 (2013)

[12] E. K. Burke, P. De Causmaecker, G. V. Berghe, and H. Van Landeghem, "The state of the art of nurse rostering," Journal of scheduling, vol. 7, pp. 441-499, 2004.

[13] B. Cheang, H. Li, A. Lim, and B. Rodrigues, "Nurse rostering problems––a bibliographic survey," European Journal of Operational Research, vol. 151, pp. 447-460, 2003.

[14] B. Cheng, J. H.-M. Lee, and J. Wu, "A nurse rostering system using constraint programming and redundant modeling," Information Technology in Biomedicine, IEEE Transactions on, vol. 1, pp. 44-54, 1997.

[15] Patrick De Causmaecker, Greet Vanden Berghe: A categorisation of nurse rostering problems. J. Scheduling 14(1): 3-16 (2011)

[16] U. Aickelin and K. Dowsland, "Exploiting problem structure in a genetic algorithm approach to a nurse rostering problem," arXiv preprint arXiv:0802.2001, 2008.

[17] E. K. Burke, G. Kendall, and E. Soubeiga, "A tabu-search hyperheuristic for timetabling and rostering," Journal of Heuristics, vol. 9, pp. 451-470, 2003.

[18] D. W. Coit, A. E. Smith, and D. M. Tate, "Adaptive penalty methods for genetic optimization of constrained combinatorial problems," INFORMS Journal on Computing, vol. 8, pp. 173-182, 1996.

[19] K. Deb, "An efficient constraint handling method for genetic algorithms," Computer methods in applied mechanics and engineering, vol. 186, pp. 311-338, 2000.

[20] S. Venkatraman and G. G. Yen, "A generic framework for constrained optimization using genetic algorithms," Evolutionary Computation, IEEE Transactions on, vol. 9, pp. 424-435, 2005.

[21] U. Aickelin and P. White, "Building better nurse scheduling algorithms," Annals of Operations Research, vol. 128, pp. 159-177, 2004.

[22] U. Aickelin and K. A. Dowsland, "An indirect Genetic Algorithm for a nurse-scheduling problem," Computers & Operations Research, vol. 31, pp. 761-778, 4// 2004.

[23] U. Aickelin, E. K. Burke, and J. Li, "An estimation of distribution algorithm with intelligent local search for rule-based nurse rostering," Journal of the Operational Research Society, vol. 58, pp. 1574-1585, 2006.

[24] J. Li and U. Aickelin, "The application of Bayesian optimization and classifier systems in nurse scheduling," in Parallel Problem Solving from Nature-PPSN VIII, 2004, pp. 581-590.