Solving Nurse Scheduling Problem Using Constraint Programming (CP) Technique

Oluwaseun M. Alade[1], Akeem O. Amusat[1*] and Oluyinka T. Adedeji[1]

[1]*Department of Computer Science and Engineering, Faculty of Engineering and Technology, Ladoke Akintola University of Technology, Ogbomoso, Nigeria.*

Authors' contributions

This work was carried out in collaboration among all Authors. Author OMA formulated the constraint programming algorithm for solving nurse scheduling problem and wrote the first draft of the manuscript. Author AOA implemented the formulated algorithm using Python programming language. Author OTA managed the literature searches, references and does the final manuscript. All authors read and approved the final manuscript.

Original Research Article

ABSTRACT

Staff scheduling is a universal problem that can be encountered in many organizations, such as call centers, educational institution, industry, hospital, and any other public services. It is one of the most important aspects of workforce management strategy. Mostly, it is prone to errors or issues as there are many entities that should be addressed, such as the staff turnover, employee availability, time between rotations, unusual periods of activity, and even the last-minute shift changes. In this paper, constraint programming (CP) algorithm was developed to solve the nurse scheduling problem. The developed constraint programming algorithm was then implemented using python programming language. The developed CP algorithm was experimented with varying number of nurses. Experimental result confirmed that CP algorithm was able to solve nurse scheduling problem with promising results.

_____

*Corresponding author: E-mail: oalade75@lautech.edu.ng, aoamusat@student.lautech.edu.ng, otadedeji@lautech.edu.ng;

1. INTRODUCTION

Nurse scheduling is nothing but a weekly or monthly plan for all nurses in hospital, and is obtained by assigning shift categories to the nurses or assigning nurses to shift. Nurse Scheduling represents a task which consists of creating a schedule for the nurses in a hospital. The Nurse Scheduling Problem (NSP) is a common problem every hospital faces every day.

Constraint Programming (CP) is a relatively modern technology for solving constraint satisfaction and constraint optimization problems. It has arisen as a combination of techniques mainly coming from the operational research domain, artificial intelligence, and programming languages. The last 20 years CP has been successfully applied in different application areas, for instance to express geometric coherence in computer graphics, for the conception of complex mechanical structures, to ensure and/or restore data consistency, to locate faults in electrical engineering, and even for DNA sequencing in molecular biology [1].

2. LITERATURE REVIEW

Literature on nurse rostering and scheduling is very extensive. One may refer to literature reviews on the subject that provide in-depth studies on this problem such as [2,3]. A wide variety of methods have been used to tackle nurse scheduling which includes: mathematical programming [4], constraint programming [5], heuristics and meta-heuristics, hybrid methods as well as simulation. Even creative methods such as auction systems have been applied to tackle nurses' preferences [6].

Abernathy et al.[7] isolated nurse scheduling from the general staffing problem and solved it using mathematical (stochastic) programming techniques. [8] propose a two-phase goal programming heuristic for the nurse scheduling problem. [9] describe a software system called "Horoplan" for scheduling nurses in a large hospital. Apart from rostering, the system also covers some short-term staffing decisions.

Burke et al.[10] hybridize a tabu search approach with algorithms that are based upon human-inspired improvement techniques. [11] present a mixed-integer quadratic programming formulation to calculate the number of nurses from a certain skill category to do a number of shifts per day.

Three non-overlapping shift types of 8 hours each are used. The goal function aims at minimizing the difference between a given lower limit for the number of nurses and the variables, which are the number of nurses.

Rosenbloom and Goertzen [12] developed an integer programming algorithm for cyclical scheduling. The approach consists of 3 stages. A set of possible schedules is generated in the first stage. The resulting schedules are evaluated with respect to work regulations and work patterns. In the second stage, the minimum daily coverage constraints are solved with an integer program. The third stage converts the solution into work patterns for each nurse. Even though optimal solutions are generated, the approach only considers work stretches and days off.

Jaumard et al.[13] propose an exact solution approach for a flexible realistic model of the nurse scheduling problem. The generalized linear programming model applies column generation and branch and bound. It allows full exploration of the set of feasible solutions. The authors claim that their model is more flexible to address changes in the scheduling environment than a heuristic model is. However, in practice, the conflicting nature of the nurse scheduling constraints makes it very difficult to find feasible solutions.

Cheng and Wu [14] developed a nurse rostering system for solving one particular hospital problem. They make use of the ILOG solver for generating a schedule that satisfies a large set of rules (such as preferred consecutive patterns and shift balance among nurses). The rules are divided into hard and soft constraints. Solutions are generated with a 4-step procedure, that is specifically oriented towards solving the particular problem. This contribution is less generic because of the problem specific modelling but nonetheless, it has been applied in a real modern hospital and as such is a significant relatively recent addition to the literature.

Smith and Wiggins [15] presents an interactive algorithm which helps the scheduler to construct a cyclical schedule. The algorithm takes coverage constraints and days off policies into account and it determines the number of personnel members, which is a staffing decision. Not all the staff members can have rotating schedules, however. In view of more recent

developments in nurse rostering, this early heuristic for cyclical schedules cannot really be considered for practical use today.

Berrada et al.[16] combined tabu search with a multi-objective approach (see also goal programming in the above sections). It is interesting that a meta-heuristic is applied instead of an optimization approach but we think that the problem is of relatively low relevance to the real-world situation because it looks only at switching days off and working days for different people.

Dowsland and Aickelin [17] makes use of different neighborhood search strategies in a tabu search algorithm. The heuristic oscillates between feasible solutions meeting the personnel requirements and schedules concentrating on the nurses' preferences. At any time of the planning period, the algorithm must provide enough personnel with the requested qualities, while satisfying the people by granting personal requests in a fair manner. The attractiveness of work patterns differs from person to person. Rather than designing a generic, widely applicable algorithm, the model was developed for solving the personnel scheduling problem in one particular hospital and produces very good quality results for that data.

Tanomaru [18] presents a genetic algorithm to solve a staff scheduling problem. The objective is to minimize the total wage cost in a situation where the number of personnel is not fixed. Solutions have to meet the total workforce requirements while respecting the maximum number of individual working shifts. Overtime is allowed, however. Although the problem dimensions are very basic, this is one of the few research papers which allow flexible starting times for the shifts. Solutions for the personnel are represented by 7 pairs of integers, giving the start and stop times per day. For real-life problems, Tanomaru concludes that his heuristic mutation operators might be too time consuming. Also, the number of constraints that are tackled is very low.

Easton and Mansour [19] developed a distributed genetic algorithm for an employee staffing and scheduling problem called 'tour scheduling'. The algorithm aims at minimizing the number of personnel members to fulfill the demands. The fitness function represents violations of constraints and individual solutions are improved with local hill climbing operators.

The genetic algorithm works very well for a set of test problems but we think that the model is possibly too simple (e.g. no personal preferences) for real modern applications.

3. METHODOLOGY

In this paper, constraint programming technique was used to solve the nurse scheduling problem. Constraint programming is declarative in nature and does not fix information flow in one direction. It specifies the properties of the solution to be found and has close affinity with object-oriented paradigm. The constraint programming algorithm for solving nurse scheduling problem was formulated, implemented using python programming language. The implemented algorithm was evaluated using runtime and fitness value as performance metrics. The algorithm was programmed on Windows operating system with a system specification of 8GB memory and CPU of 2.6 GHz.

3.1 Formulation of Constraint Programming Algorithm

In formulating a constraint programming problem, the constraint algorithm tries to find the best solution to the problem. If no solution or inconsistency is found, then one of the variables with domain size larger than 1 is selected and a new CSP is created for each possible assignment of this variable. The following are the steps involved in solving a constraint satisfaction problem:

- Initial variable assignment: Each time the solver makes a variable assignment, it has a couple of choices to make in order to select a solution.
- The solver evaluates the solution for optimality i.e. fitness value computation
- Backtracking: if the current solution is not optimal, the solver moves back in the search tree to try other variable assignment. This is called backtracking.
- Feasible solution: the solver arrives at the best feasible solution (solution with the best fitness value) when all variable has been assigned.

The pseudocode for the CP algorithm is below:

Procedure allocate-shift (nurse[], shift-type, shift-num)
Discard nurses who cannot take up shift-type for the week

```
sort nurse[] according to the date of last shift-
type
i = size(nurse[] ) ? shift-num
if i < 0
copy the first i nurses in nurse[] to the end of
nurse[]
endif
i=0
day=1
while shift-num > 0
current-nurse = nurse[i]
if current-nurse is unavailable for shift-type
on day
swap current-nurse with one in assigned-
nurse[]
 if not successful
swap current-nurse with nurse[j] ( j > i )
 endif
 if not successful
 swap current-nurse with a discarded nurse
 endif
if not successful
replace current-nurse with one in assigned-
nurse[]
endif
 if not successful
output "No shift can be allocated"
endif
endif
assign shift-type to current-nurse on day
insert current-nurse into assigned-nurse[]
increment( i )
 increment( day )
 decrement( shift-num )
endwhile
endprocedure
```

There's need to highlight the constraints associated with nurse scheduling problem. Constraints can be classified into two: hard constraints and soft constraints. Hard constraints are those that cannot be violated and define the feasibility of solutions. Hard constraints are concerned with the hospital's needs as opposed to the nurses' preferences. Soft constraints are desirable but not obligatory, and thus can be violated. Table 1 shows various constraints associated with our constraint programming algorithm.

3.2 Mathematical Representation of the Problem

The NSP can be mathematical represented as thus:

$$\alpha \times f \left(\sum_{i=1}^{I} \sum_{k=1}^{K} \sum_{s=1}^{S} Di,k,s \right) + (1-\alpha)$$
$$\times G \left(\sum_{i=1}^{I} \sum_{k=1}^{K} \sum_{s=1}^{S} Di,k,s \right)$$

where *i* represent some nurse *i*.
where *k* represents the days.
where *s* represents the shifts.

4. EXPERIMENTAL RESULTS

The developed algorithm was tested with 14 and 20 nurses using four different shifts (morning, afternoon, night and off). The results for CP algorithm are shown in Table 2 and Table 3 using 14 and 20 nurses respectively. The algorithm was tested under four separate runs.

Table 1. Constraints and their description

Constraint	Description
HC1	Each day is divided into three 8-hour shifts (morning, afternoon and evening).
HC2	On each day, all nurses are assigned to different shifts and one nurse has the day off.
HC3	Each nurse works five or six days a week.
HC4	No shift is staffed by more than two different nurses in a week.
HC5	If a nurse works shifts 2 or 3 on a given day, he must also work the same shift either the previous day or the following day.

Table 2. Result after 4 runs using 14 nurses

Run	Runtime of the Program (sec)	Fitness Value
1	40.75	7.87
2	40.92	7.87
3	40.72	6.93
4	40.73	6.12
Average	**40.79**	**7.19**

Table 3. Result after 4 runs using 20 nurses

Run	Runtime of the Program (sec)	Fitness Value
1	43.90	7.87
2	43.87	6.71
3	43.97	7.01
4	43.92	7.21
Average	**43.92**	**7.20**

The result showed that all hard constraints are fulfilled and there's no unallocated nurses

Table 4. Shifts generated by constraint programming algorithm for 14 nurses

Nurse/Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
N1	A	N	O	O	N	A	M	M	N	N	N	M	M	M	N	O	O	M	O	A	O	A	N	N	N	N	N	M	A
N2	O	O	M	N	M	A	N	A	N	N	A	N	M	A	O	A	M	A	O	M	A	A	O	O	N	N	O	A	O
N3	N	M	O	A	N	N	O	O	O	N	A	A	M	O	A	O	O	O	O	M	N	A	A	M	O	A	A	A	M
N4	N	N	N	N	O	A	N	M	A	N	A	A	A	O	N	A	O	N	M	O	A	N	A	A	A	N	O	O	O
N5	A	A	N	A	A	M	A	M	A	A	A	A	N	O	O	M	A	M	A	M	A	M	N	O	O	A	A	M	N
N6	M	O	A	M	N	A	M	O	O	M	A	M	A	A	O	A	M	N	A	A	N	A	N	N	O	N	M
N7	M	O	N	N	M	O	N	A	A	M	O	A	O	M	O	O	M	N	O	O	N	O	M	A	M	O	N	O	O
N8	A	M	A	N	O	A	M	M	O	O	M	M	A	M	O	N	N	O	A	M	A	N	O	N	A	O	O	M	O
N9	O	O	O	N	O	O	N	A	M	A	O	A	N	N	A	A	A	M	A	N	O	M	O	N	O	M	O	A	N
N10	N	O	N	O	A	N	A	N	N	N	O	M	M	A	O	N	M	N	M	O	A	A	A	O	N	O	N	O	A
N11	O	O	N	M	M	N	O	O	O	N	M	O	A	O	O	O	N	A	A	O	O	M	O	O	O	A	M	O	M
N12	M	M	N	N	A	A	N	A	N	O	N	M	N	M	N	N	M	N	M	M	O	N	N	N	M	M	A	M	M
N13	A	M	A	N	A	M	A	O	A	N	N	M	M	A	M	A	N	M	N	N	M	N	O	N	O	N	O	N	M
N14	M	O	M	A	M	M	A	M	M	O	A	M	O	A	M	N	O	O	O	O	A	A	O	A	N	A	O	O	A

5

Table 5. Shifts generated by constraint programming algorithm for 20 nurses

Nurse/Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
N1	A	N	O	O	N	A	M	M	N	N	N	M	M	M	N	O	O	M	O	A	O	A	N	N	N	N	N	M	A
N2	O	O	M	N	M	A	N	A	N	N	A	N	M	A	O	A	M	A	O	M	A	A	O	O	N	N	O	A	O
N3	N	M	O	A	N	N	O	O	O	N	A	A	M	O	A	O	O	O	O	M	N	A	A	M	O	A	A	A	M
N4	N	N	N	N	O	A	N	M	A	N	A	A	A	O	N	A	O	N	M	O	A	N	A	A	A	N	O	O	O
N5	A	A	N	A	A	M	A	M	A	A	A	A	N	O	O	M	A	M	A	M	A	M	N	O	O	A	A	M	N
N6	M	O	A	M	N	A	M	O	O	M	A	M	A	A	O	A	M	N	A	A	N	A	N	A	N	N	O	N	M
N7	M	O	N	N	M	O	N	A	A	M	O	A	O	M	O	O	M	N	O	O	N	O	M	A	M	O	N	O	O
N8	A	M	A	N	O	A	M	M	O	O	M	M	A	M	O	N	N	O	A	M	A	N	O	N	A	O	O	M	O
N9	O	O	O	N	O	O	N	A	M	A	O	A	N	N	A	A	M	A	N	O	M	O	N	O	M	O	A	N	
N10	N	O	N	O	A	N	A	N	N	N	O	M	M	A	O	N	M	N	M	O	A	A	A	O	N	O	N	O	A
N11	O	O	N	M	M	N	O	O	O	N	M	O	A	O	O	O	N	A	A	O	O	M	O	O	O	A	M	O	M
N12	M	M	N	N	A	A	N	A	N	O	N	M	N	M	N	N	M	N	M	M	O	N	N	N	M	M	A	M	M
N13	A	M	A	N	A	M	A	O	A	N	N	M	A	M	A	N	M	N	N	M	N	O	N	O	N	O	N	M	
N14	M	O	M	A	M	M	A	M	M	O	A	M	O	A	M	N	O	O	O	O	A	A	O	A	N	A	O	O	A
N15	A	N	N	N	O	M	N	N	M	A	O	A	O	N	O	N	A	O	A	A	N	A	A	O	A	O	N	A	O
N16	A	M	M	M	A	N	O	O	M	A	N	A	M	N	N	A	M	O	O	M	N	A	O	A	A	N	O	M	O
N17	N	A	M	A	A	N	N	O	A	N	O	M	A	O	O	O	M	M	N	N	M	M	N	M	M	M	M	A	A
N18	A	O	N	N	N	A	A	N	A	N	O	M	O	O	M	N	M	N	M	M	M	O	N	O	A	M	M	N	O
N19	O	O	A	M	O	N	O	A	A	A	A	M	O	O	O	A	M	O	M	N	M	N	A	O	O	N	O	M	M
N20	N	N	N	O	N	M	N	A	O	N	N	N	N	M	N	O	A	A	A	N	A	N	M	O	M	A	A	M	N

5. DISCUSSION

Table 4 shows the shifts generated by constraint programming (CP) algorithm for 14 nurses. Table 5 shows the shifts generated by constraint programming (CP) algorithm for 20 nurses. It is important to know that the CP algorithm was tested 4 times with variable number of nurses. A schedule that produced manually has shown that there is an obviously unbalance shift pattern for the nurses. On the other hand, the schedule produced by CP approaches shown more balance in giving a shift to the nurses where all nurses had assigned to shift morning, afternoon, night and off. This led to fitness value equal to 7.1975. The developed algorithm using the CP approach produce better nurse scheduling due to the minimum fitness value. Although there have nurses which got less weightage but on the next cycle of schedule, the weightage value will be maintained back using swap process.

The CP algorithm developed was able to generate schedules for up to 250 nurses. During experimentation of the algorithm, we found out that the space and time requirement of the algorithm increases as the number of input (i.e. nurses) increase. One interesting feature of the CP algorithm is that, more constraints can be added to improve the quality of the generated solution.

Table 2 and Table 3 shows the result of the algorithm experimentation with 14 nurses and 20 nurses respectively. From Table 2 and Table 3, we can see that the time it took the CP algorithm to generate the schedule for 20 nurses is larger than the time it took it to generate the schedule for 14 nurses.

6. CONCLUSION

An effective and practical way of generating nurse schedule is vital in order to ensure the effectiveness of the operation in the hospital.

In this paper, constraint programming (CP) technique was used to solve nurse scheduling problem (NSP). In regards with the result obtained after various test on different number of nurses, the CP technique shows its ability in finding optimal solution to NSP with higher footprint on computational resources. The CP technique can be summarily described below as:

- CP is a general technique and can encapsulate a lot of work.
- CP allows the use of symbolic representation.
- The performance of the CP algorithm depends on the number input and available computational resources.

7. FURTHER WORK

After reviewing the performance of CP technique on NSP, the following future studies can be done:

- Add more constraints (e.g. taking the level and experience of nurses into consideration.) to the algorithm.
- Evaluate the performance of CP algorithm with other scheduling algorithms to solve nurse scheduling problem.
- Implement the CP algorithm on GPU (graphics processing unit) to minimize the footprint on system resources and also provide GUI (graphical user interface) for easy use by naive users.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. Rossi F, Beek P, Walsh T. Handbook of Constraint Programming. 2006:32(3):672-695.
2. Burke EK, De Causmaecker P, Vanden Berghe G, Van Landegem H. The state of the art of Nurse Rostering. Journal of Scheduling. 2004:7:441–499.
3. Ernst AT, Jiang H, Krishnamoorthy M, Sier D. Staff scheduling and rostering: A review of applications, methods and models, European Journal of Operational Research. 2004;3–27.
4. Ikegami Niwa A. A Subproblem-centric Model and Approach to the Nurse Scheduling Problem, Mathematical Programming. 2003;97(3):517-541.
5. Weil G, Heus K, Francois P, Poujade M. Constraint programming for nurse scheduling. IEEE Engineering in Medicine and Biology. 1995;417-422.
6. De Grano L, Medeiros DJ, Eitel D. Accomodating Individual preferences in nurse scheduling via Auctions and optimization. Health Care Management Science. 2009;12:228-242.

7. Abernathy WJ, Baloff N, Hershey JC, Wandel S. A three-stage manpower planning and scheduling model: A service-sector example. Operations Research 1973;22:693–711.
8. Arthur JL. Contributions to the Theory and Applications of Goal Programming. Unpublished PhD dissertation, School of Industrial Engineering, Purdue University, West Lafayette.
9. Darmoni SJ, Fajner A, Mahé N, Leforestier A, Vondracek M, Stelian O, Baldenweck Horoplan: Computer-assisted nurse scheduling using constraint based programming. Journal of the Society for Health Systems. 1995;5:41-54
10. Burke EK, Bykov Y, Petrovic S. A multicriteria approach to examination timetabling, Burke EK, Erben W. (Eds.): Practice and theory of automated timetabling, Third International Conference, Konstanz, Springer. 2000;118-131
11. Warner DM, Prawda J. A mathematical programming model for scheduling nursing personnel in a hospital. Journal of Management Science. 1972;19:4:411-422.
12. Rosenbloom ES, Goertzen NF. Cyclic nurse scheduling. European Journal of Operational Research. 1987;31(1):19-23.
13. Jaumard Brigitte, Semet Frederic and Tsevi Vovor. A generalized linear programming model for nurse scheduling. European Journal of Operational Research.1998;107(1):1-18
14. Cheng BMW, Wu JCK. Nursing staff rostering using ILOG solver. Final year project report, Department of Computer Science and Engineering, The Chinese University of Hong Kong; 1996.
15. Smith ID, Wiggins A. A computer-based nurse scheduling system, computers and Operations Research. 1977;4(3):195-212.
16. Berrada J, Ferland P, Michelon : A Multi-Objective Approach to Nurse Scheduling with both Hard and Soft Constraints, Socio-Economic Planning Science. 1996:183-193.
17. Dowsland K, Aickelin U: Exploiting problem structure in a genetic algorithm approach to a nurse rostering problem, Journal of Scheduling. 2000;3:139-153.
18. Tanomaru Julio. Staff scheduling by a genetic algorithm with heuristic operators. Conference: Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., IEEE International Conference. 1995;3:456-461.
19. Easton FF, Mansour N. A distributed genetic algorithm for deterministic and stochastic labor scheduling problems. European Journal of Operational Research. 1999;118(3):505-523.