

# CONSTRAINT SOLVING ENGINE BASED NURSE ROSTERING WITH INTELLIGENT BACKTRACKING

<sup>1</sup>Chitrakala, S., <sup>2</sup>S.P. Surendernath and <sup>3</sup>R. Priyatharshini

<sup>1</sup>Department of Computer Science and Engineering, Anna University, Chennai, India

<sup>2</sup>Department of Information Science and Technology, Anna University, Chennai, India

<sup>3</sup>Department of Information Technology, Easwari Engineering College, Chennai, India

Received 2013-12-09; Revised 2013-12-14; Accepted 2014-05-22

## ABSTRACT

Efficient utilization of time and effort is essential in Personnel scheduling problems to evenly balance the workload among the people and attempt to satisfy the personnel preferences. In Constraint Satisfaction Problem based scheduling problems, when a branch of the search fails the backtracking search algorithm back up to the preceding variable and try a different value for it. So here the most recent decision point is revisited. Its run-time complexity for most nontrivial problems is still exponential. A solution is intelligent backtracking scheme in which backtracking is done directly to the variable that caused the failure. This study proposes Constraint Satisfaction Problem based Nurse Rostering using Intelligent Backtracking approach. The proposed Minimal Critical Set based Intelligent Backtracking (MCS-IBT) algorithm incorporates Critical Set detection which is followed by Minimal Critical Set reduction in order to reduce the search space for nurse rostering. MCS\_IBT overcomes missing good MCSs by visiting optimal number of sets. This study finds its applications in scheduling, temporal reasoning, graph problems, floor plan design, the planning of genetic experiments and the satisfiability problems. The implemented system is tested on the real life data from the hospital and the results shown remarkable performance.

**Keywords:** Constraint Satisfaction Problem, Intelligent Backtracking, Nurse Rostering, Minimal Critical Set

## 1. INTRODUCTION

Nurse Rostering is a complex scheduling problem that affects hospital personnel on a daily basis all over the world. Nurse Rostering is defined as the process of generating timetables for specifying the work shifts of nurses over a given period of time. Various approaches for Nurse Rostering are Mathematical programming, Goal programming, Declarative and constraint programming and meta-heuristics approaches. The optimal combination of these techniques is different for different problems and is a topic of current investigation. The objective of this study is to design and implement a nurse rostering system which involves generating monthly timetables for a group of nurses. A constraint solving engine based approach for nurse rostering system has been proposed. Constraint-Satisfaction problems that can be stated as follows: We are given a set of variables,

a finite and discrete domain for each variable and a set of constraints. Each constraint is defined over some subset of the original set of variables and limits the combinations of values that the variables in this subset can take. The goal is to find one assignment to the variables such that the assignment satisfies all the constraints. In some problems, the goal is to find all such assignments. There are a number of hard hospital rules and soft nurse preference rules to be obeyed. A backtracking search is a depth-first traversal of a search. The search technique is generated as the search progresses and represents alternative choices that can be examined in order to find a solution. In Constraint Satisfaction Problem (CSP) based scheduling problems, when a branch of the search fails, the backtracking search algorithm back up to the preceding variable and try a different value for it. Each possible combination of the variables is systematically generated and then tested

**Corresponding Author:** Chitrakala, S., Department of Computer Science and Engineering, Anna University, Chennai, India

to see if it satisfies all the constraints. The first combination that satisfies all the constraints is the solution. Here variables are instantiated sequentially. As soon as all the variables relevant to a constraint are instantiated, the validity of the constraint is checked. If a partial instantiation violates any of the constraints, backtracking is performed to the most recently instantiated variable that still has alternatives available. Although backtracking is strictly better than the generate-and-test method, its run-time complexity for most nontrivial problems is still exponential. A solution is intelligent backtracking scheme in which backtracking is done directly to the variable that caused the failure. Minimal Critical Set based Intelligent Backtracking (MCS\_IBT) approach for nurse rostering using degree heuristics has been proposed in this study for intelligent backtracking. The rest of this study is organized as follows. Section II provides an account of related work in the field of nurse rostering. In section III, we give the overview of our proposed system. Section IV provides the experimental results and discussion section V summarizes our contributions and sheds light on future direction of research.

### 1.1. Related Works

There have been extensive attempts to solve the nurse rostering problem. Cheng *et al.* (1997) implemented Constraint-based nurse rostering system using a redundant modeling. Redundant modeling is an effective way to increase constraint propagation through cooperation among different models for the same problem in order to reduce the search time. Hierarchical and stochastic constraint methods can also be investigated for their suitability to cope with soft constraints.

Beddoe *et al.* (2008) developed a Case-Based Reasoning (CBR) system called CABAROST (CAsed-Based ROS Tering) for personnel scheduling problems which captures nurse rostering experience from personnel managers by storing examples of their repairs of constraint violations. Future research work can be carried out for personnel managers to violate more of the nurses' shift preferences due to an increasing patient load and the decisions made by CABAROST should reflect this.

Lombardi and Milano (2009) proposed a method for computing resource conflicts based on the minimum flow on the resource. Here a Precedence Constraint Posting approach is used which adds precedence constraints to the original project graph so that all resource conflicts are solved.

Farzianpour *et al.* (2011) developed an approach in order to provide quality services in emergency departments and to enable managers to assess and

evaluate the healthcare system the manner of service provision in these departments.

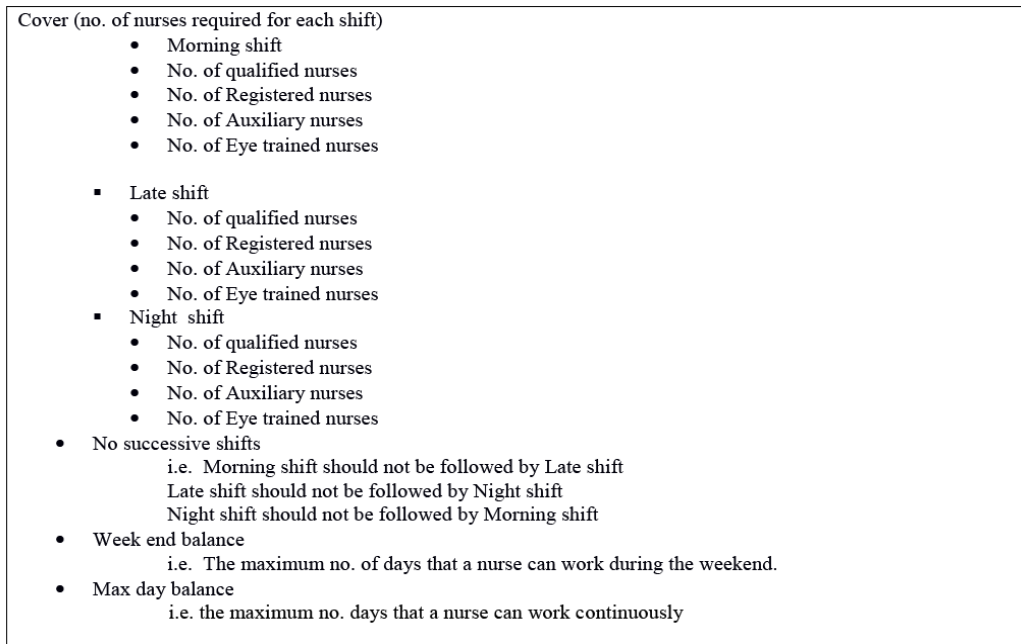
Muenvanichakul and Charnsethikul (2009) developed a Decomposition Based Heuristics for Large- Scale Dynamic Quadratic Assignment Problems. Approximate Benders Decomposition generates the ensemble of a subset of feasible layout for Approximate Dynamic Programming to determine the sub-optimal optimal solution.

Lombardi and Milano (2012) developed a method for Minimal Critical Set identification using Min-Flow algorithm is used for getting the solution of scheduling problems via Precedence Constraint Posting has been implemented. The method is based on a mini-mum-flow problem and a heuristic minimization step. The proposed approach is much more scalable, faster and easier to implement and more versatile. As a main disadvantage, this method visits a small number of sets, with the risk to miss good MCSs according to the ranking heuristic.

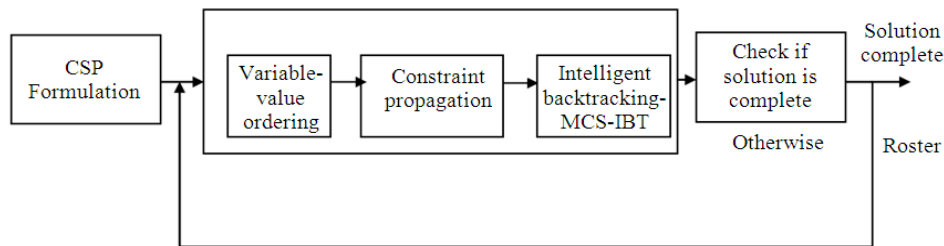
Qamal and Ayob (2008) developed CSP search based solutions for Examination Room Assignment Problem. Gîrbea *et al.* (2011) implemented a fully automated planner schedule for constraint satisfaction problems. It is observed from the literature that Detecting critical set is a key issue and the search space for finding solutions has to be reduced in an intelligent manner. The above issues have been addressed in this study.

## 2. MATERIALS AND METHODS

This study proposes CSP based Nurse Rostering using intelligent backtracking approach. Proposed MCS\_IBT algorithm incorporates Minimal Critical Set reduction by choosing optimal critical sets in order to reduce the search space. Dataset for implementation of this study is obtained from real life data from the hospital. The constraints that are followed for nurse rostering is shown in **Fig. 1**. CSP assign value to appropriate variables using variable ordering and once the variable has been selected the algorithm must decide on the order in which to examine its values using value ordering. Constraints are propagated among variables using arc consistency. If it is not able to find the solution, Backtracking search is done directly to the variable in the identified set that caused the failure in an intelligent way. This identified set of variables is called conflict set. Conflict set is encoded as critical set and this critical set is minimized to produce Minimal Critical Set (MCS) based on the significance of the variables from critical set. The proposed MCS-IBT algorithm uses the conflict sets defined in this way for finding the rostering solution. The Block Diagram of proposed system is shown in **Fig. 2**.



**Fig. 1.** Nurse rostering constraints



**Fig. 2.** Block diagram of proposed system

### 2.1. CSP Formulation

Initial CSP formulation is done by assigning values to some or all of the variables,  $\{X_i = V_i, X_j = V_j \dots\}$  where the variable is the Nurse and the value is the shift assigned for the nurse. Each variable  $X_i$  has a nonempty domain  $D_i$  of possible values. Basically there are four types of nurses Qualified, Registered, Eye trained and auxiliary.

Each nurse is assigned a type which is presented as a 6-tuple: NurseType $_t$  = qual, gend, intl, train, grade, hours,  $t = 1 \dots T$ , where the elements of NurseType $_t$  where  $t$  is the number of different types of nurses are given in **Table 1**. The nurse rostering problem is characterized by a variety of different constraint types.

In total 11 constraint types were identified: *Cover* constraints define the minimum number of nurses of a particular type that must be assigned to a particular

shift. For example, the early shift requires 4 qualified nurses. A number of cover constraints imposed on a single shift together define the skill mix that they must be available on the ward for that shift. An assignment that does not violate any constraints is called a consistent or legal assignment. A solution to CSP is a complete assignment that satisfies all the constraints. Some CSPs also require a solution that maximizes an objective function. The CSP has a constraint graph representation as shown in **Fig. 3** where nodes are variable and the edges denote the constraint between the variable pairs. The Notations used in our constraint graph are Q-Qualified, R-Registered, ET-Eye trained and A-Auxiliary.

A constraint-based approach for nurse rostering system has been proposed which performs the rostering of nurses for 30 days based on various given constraints. A roster is defined as a set of assignments of employees

to shifts on each day over a specific period. Rosters should conform to a set of constraints, which reflect legal, management and staff requirements.

## 2.2. Hard Request

Constraints are violated if a nurse's preferred shift is different from their assigned shift and the preferred shift is a hard request.

## 2.3. Max Hours

Constraints set the maximum number of hours a nurse may work over a period. This can differ considerably between nurses and depends on their individual working contracts. For example, full time nurses at the QMC ward may not work more than 75 h in a fortnight.

## 2.4. Single Night

Constraints are violated if nurses are as-signed individual night shifts. Nurses at the QMC ward prefer to work night shifts in blocks of two or more.

## 2.5. Soft Request

Constraints are violated if a nurse's preferred shift is different from their assigned shift and the preferred shift is a soft request.

## 2.6. Succession

Constraints define illegal shift combinations for nurses. It is not desirable to work shifts of one type on one day followed by shifts of another type on the day after. For example, a NIGHT shift followed by an EARLY shift is one such combination.

## 2.7. Weekends in A Row

Constraints set the maximum number of weekends that nurses may work in a row. This is usually three in the QMC ward.

## 2.8. Weekend Split

Constraints are violated if a nurse works only one of the days in a weekend.

## 2.9. Weekend Balance

Constraints set the number of week-ends that nurses may work over a period. They state the maximum number of weekends a nurse may work in any given number of weekends. For example, nurses in the QMC ward may not work more than 3 weekends out of every 4, unless it is stipulated in their contracts.

## 2.10. Min Hours

Constraints set the minimum number of hours a nurse may work over a period. This constraint is violated if nurse's time is under-utilized by the roster.

## 2.11. Max Days On

Constraints limit the number of days that shift is a hard request. Nurses may work in a row. For the QMC ward this is generally 6 for all nurses.

## 2.12. Min Days On

Constraints define the minimum number of days that nurses may work successively.

## 2.13. Backtracking Search for CSP's

Using the CSP formulation, Backtracking search algorithm is used to solve CSP. CSP search algorithms generate successors by considering possible assignments for only a single variable at each node in the search tree.

## 2.14. Variable and Value Ordering

The variable is ordered based on Minimum Remaining Value (MRV) heuristic i.e., choosing the variable with fewest legal values. Once the variable has been selected the algorithm must decide on the order in which to examine its values. The value is ordered using least-constraining-value heuristic.

## 2.15. Constraint Propagation

Constraint propagation is the general form of propagating the implications of a constraint on one variable to the other variable. To better make use of constraints during search, a forward checking technique is used. Although forward checking detects many inconsistencies, it does not detect all of them. In order to re-duce the amount of search during constraint propagation Constraints are propagated among variables using arc consistency which is a fast method of constraint propagation. An arc refers to a directed arc in the constraint graph. In constraint graph, the forward and reverse arc consistency between 2 nodes is checked is shown in **Fig. 4**. The above process is repeated iteratively till it finds the solution.

## 2.16. Basic Rostering Function

The `rosterutil()` function performs the rostering for the day and return true if the operation is successful for the day else, the assigned shifts are backtracked by forward checking and arc consistency by calling the function `reassign`.

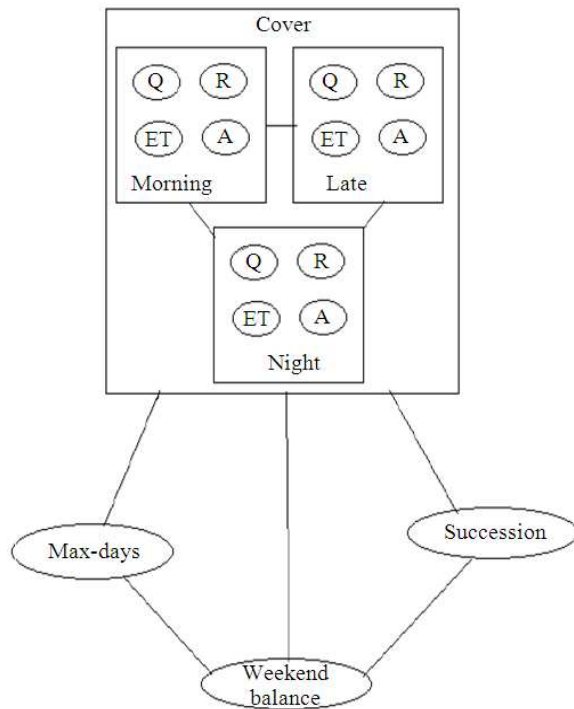


Fig. 3. Constraint graph

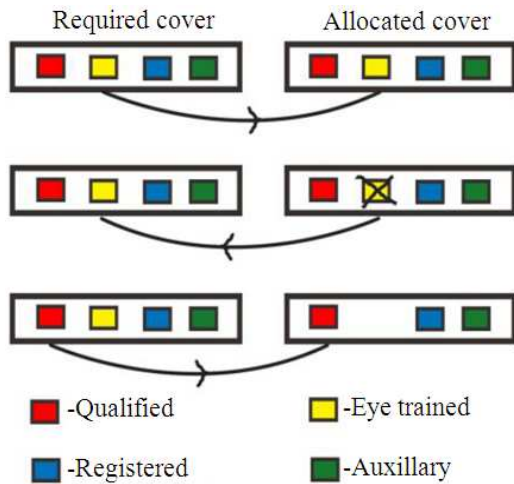


Fig.4. Arc consistency diagram

**Algorithm**

Function roster (constraints) returns a solution  
 For day = 1 to 30  
 While RoasterUtil (0, day) return false  
 Alter constraint one by one and  
 reassign

```

    by forward checking and arc
    consistency
    return solution
    Utility function performs the rostering for the given
    day. This function returns either failure or success. It
    assigns shift by calling the fix_morning (), fix_late() and
    fix_night () functions if the constraints are not violated.
    It assigns shifts to all the nurses recursively.
    Roaster Util (nurse_no, day) returns success or
    failureIf nurse_no > max. no. of nurses then
    Return failure
    If maxday constraint is not violated
    then
    If fix_morning(nurse_no, day) is failure
    then
    If fix_late (nurse_no, day) returns failure
    then
    Fix_night (nurse no, day)
    If all the cover are satisfied
    then
    Return true
    Return RoasterUtil (nurse_no+1, day)
    Shift Assignment functions return success if the
    given nurse is assigned with the shift otherwise return
    failure. These functions check the succession constraint,
    max day constraint, weekend balance constraint and the
    cover constraint. If all are satisfied, then it assigns the
    particular shift and alters the values of the nurse and the
    constraints of the day accordingly. One of the functions
    to fix morning shift is shown below.
    fix_morning (nurse_no, day)returns success or failure
    If the previous shift is night
    then
    return failure
    if max day constraint is violated
    then
    return failure
    if weekend constraint is violated
    then
    return failure
    if cover for the shift and the
    type cover
    are satisfied
    then
    return failure
    solution [nurse_no][day]='morning'
    increment type cover of the day by one
    increment weekend if it is a weekend
    increment max day
    return success
    
```



**Table 1.** The fields of nurse type

Field	Domain	Description
Qual	{RN, EN, AN, SN, QN, PN, XN}	The qualification level of the nurse
Gend	{M, F}	The gender of the nurse-M) ale or (F) emale
Intl	{I, H}	The nationality of the nurse-I) nternational or (H) ome
Train	{ET, NT}	The specialty training of the nurse-Eye-Trained (ET) or Non-eye-Trained (NT)
Grade	{A, B, C, D, E, F, G, H, I}	The grade of the nurse according to the NHS grading system
Hours	[8, 37.5]	The number of hours the nurse should work each week (by contract)

## 2.17. Intelligent Backtracking

Run-time complexity of backtracking algorithms for most nontrivial problems is still exponential. A solution is intelligent backtracking scheme in which backtracking is done directly to the variable in the identified set that caused the failure in an intelligent way. This identified set of variables is called conflict set. In general the conflict set for variable X is the set of previously assigned variables that are connected to X by constraints. The back jumping method backtracks to the most recent variable in the conflict set.

Let  $X_i$  be the current variable and let  $conf(X_i)$  be its conflict set. Conflict set is encoded as critical set and this critical set is minimized to produce Minimal Critical Set (MCS) based on the significance of the variables from critical set. The significance is determined by degree heuristics. The Degree of all the variables in the conflict set is determined. Then the variables are ranked based on the degree. Minimization of the above conflict set is done by selecting the variables that are involved in the largest number of constraints on other variables. The proposed Intelligent Backtracking algorithm backtracks to the right point in the search tree for finding the rostering solution. MCS represents the nodes (nurses) in the constraint graph which are to be backtracked to find the rostering solution. Once the point to backtrack in the search tree has been identified the shifts are reassigned for the nurses. Algorithm for the proposed MCS-IBT is as follows:

```

Let CS[n] = Critical set nodes
D [] = Degree of critical set nodes
MCS [] =Minimal Critical Set
M=no of variables with largest no of constraints
MCS_IBT ()
{
  CS[n] =CS_Deduction();
  For i=1 to n of CS
  {
    D[j]=Degree(i);
  }
  Sort (D[j]);
  MCS [] =Select Top M () from sorted array

```

```

Reassign (MCS)
{
  For nurse=1 to MCS []
    If the nurse is assigned
      then
        Modify the successive shift to the
        previous shift
        Decrement the max day
    End if
    //arc consistency no need to alter constraint if
    cover is satisfied//
    if(cov_Assign[day][MORNING][type[nurse]]
    ==cover[MORNING][type[nurse]])
      then
        continue the loop
    if(cov_Assign[day][LATE][type[nurse]]equals
    cover[LATE][type[nurse]])
      then
        continue the loop

    if(cov_Assign[day][NIGHT][type[nurse]]==co
    ver[NIGHT][type[nurse]])
      then
        continue the loop
    removes the allocated shift for the nurse for the given
    day
    reset the cover for the day
  }}

```

## 3. RESULTS

The Proposed Nurse rostering system is implemented with four types of nurses and 11 constraints. The implemented system is tested on the real life data from Soorya hospital, Chennai. The generated rosters are sent to the hospital duty officer for evaluation and comments are considered to improve the quality of the solutions.

One particular preference roster with constraint violation has been shown in **Fig. 5**. Since the ET nurses are limited, there is no other way that we should overload them by giving more shifts though the maximum consecutive working days is 3 which is shown in the highlighted part of **Fig. 5** as constraint violation.

ROASTERING FOR ONE MONTH

	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	30	
nurse_1(Q)	M	N		L	L	L	M	N		L	L	L	M		L	M	N		L	L	L		L	L		L	L	M		M	
nurse_2(Q)	M	N		L	L	L	M	N		L	L	L	M		L	M	N		L	L	L		L	L		L	L	M		M	
nurse_3(Q)	M	N		N	N		L	L	L	M	N		L	L	M	N		N		N		N		L	M		M	M		N	
nurse_4(R)	M	N		N	N		N	N		N	N		N	N		N	N		N		N		L	M		N	N		N		
nurse_5(A)	M	N		M	N		L	L	L	M	N		L	L	M	L	L	M		N		N		L	M		N	N		N	
nurse_6(ET)	M	N		L	L	L	M	N		L	L	L	M		L	M	N		N		N		L	M		N	N		N		
nurse_7(Q)	L	L		M	M		N	L		M	M		M	M	N	N		N		N		N		L	M		M	N		N	
nurse_8(ET)	L	L		M	M		N	L		N	N		N	L	M	N		N		L	L	L		L	L		L	M		M	
nurse_9(A)	L	L		L	M		M	M		N	N		N	L	M	N		N		L	L	L		M	M		L	M		M	
nurse_10(R)	N		N		N		N		N		N		N		N		N		N		N		N		N		N		N		N
nurse_11(ET)	N		L	M	N		L	M		M	M		M	N	N		N		N		N		N		N		N		N		N
nurse_12(Q)	L	L		M	M		N	N		N	N		N	N	L	L	M		N		M	M		M	M		L	L		L	
nurse_13(Q)	N		L	M	N		L	L		M	M		L	L	M	N		M		M		M		M	M		L	L		L	
nurse_14(Q)	N		L	M	N		N	N		N	N		N	N	L	M	M		N		L	M		M	M		N	N		N	
nurse_15(Q)	N		N	N	N		N	N		N	N		N	N	L	M	M		N		M	M		N	N		N	N		N	
nurse_16(ET)	M	M		N	N		N	L		L				L	M	N		N		N		N		N		N		N		N	
nurse_17(Q)	N		N		N		N		N		N		N		N		N		N		N		N		N		N		N		N
nurse_18(Q)	N		N		N		N		N		N		N		N		N		N		N		L	M		N		M	N		N
nurse_19(Q)	M	M		M	M		M	M		M	M		M	N		M	N		N		N		N		N		N		N		N
nurse_20(Q)	M	M		N			N							N		M	N		N		N		N		N		N		N		N
nurse_21(A)	N		L	L	M		N	N		L			N	N	N	L		N		N		N		L	L		M	N		N	
nurse_22(Q)	M	M		N			M							N		N		N		N		N		N		N		N		N	
nurse_23(Q)	M	M		M	M		M	M		M	M		M	M		M	M		M		M		M	M		M	M		M		M
nurse_24(R)	M	M		M	M		M	M		M	M		M	M		M	M		M		M		M	M		M	M		M		M
nurse_25(Q)	N						M						M						N		M		N		N		N		N		N
nurse_26(R)																															M
nurse_27(A)	M	M		N			L			M	M							M	M		N		N		N		N		N		N
nurse_28(ET)	N												N																	M	

Fig. 5. Constraint violation screenshot

The quality of the roster generated has been evaluated using the percentage of soft constraints satisfied as in Table 2 which shows the degree of satisfaction of soft constraints in the generated roster. The percentage for each rule represents the ratio of the number of satisfied soft constraints to the total number of soft constraints generated from the rule. The implemented system has been checked upon with different instances by altering the Number of nurses, changing the different types of constraints and altering constraints. In Table 2, some of the constraints are less satisfied which gives future extension of this study. For example, Succession constraints define illegal shift combinations for nurses. It is not desirable to work shifts of one type on one day followed by shifts of another type on the day after. For example, a NIGHT shift followed by an EARLY shift is one such combination. The total number of possible soft constraints is 28, as there are totally 28 nurses. Out of these 28 nurses, only 18 of them have a NIGHT shift not followed by an EARLY shift. This gives a soft constraint satisfaction of 64.2%. Other percentages are calculated similarly.

4. DISCUSSION

The effectiveness of our proposed MCS\_IBT based NRP algorithm has been demonstrated on several benchmark NRPs at <http://www.cs.nott.ac.uk/~tec/NRP> by comparing with other reported works such as CBRG (Case-based repair generation) (Li and Lim, 2003) and CP-CG (He and Qu, 2012) for important planning rules such as Succession, Soft requests, Single Night, Min Days on, Max Days on. It is observed that our approach gives better satisfaction rate (percentage) compared to the above approaches as shown in Table 3.

Table 2. Soft constraint satisfaction of a single week

Rule	Satisfied	Percentage
Soft request	16	57
Succession	18	64
Single night	26	92
Min days on	27	96
Max hours	25	89

Table 3. Soft constraint satisfaction of eight weeks

Rule	Satisfied	Percentage
Soft request	14	50
Succession	17	62
Single night	25	89
Min days on	24	85
Max hours	22	78

In Hybrid AI approach some requested shifts make it impossible to satisfy related patterns. In CP-CG approach, night shifts are assigned first before the other shifts are randomly selected and assigned to the nurses in value ordering. It leads to less satisfaction rate for Succession. Compared to the above 2 approaches, proposed approach provides comparable results for various constraints due to optimal rostering solutions using MCS based IBT.

5. CONCLUSION

Real life scheduling problems have been successfully Implemented which consists of a constraint solving Engine by incorporating intelligent backtracking Approach which includes Critical Set detection followed By Minimal Critical Set reduction in order to reduce search space The proposed MCS\_IBT algorithm uses

degree heuristics for identifying optimal number of critical set nodes so as not to miss good MCS candidates. The system is able to produce rosters for 28 nurses of different skill classes with 4 shifts in a timely manner. Experimental results have shown the promising results of the potential of constraint technology in the computerization of hospital management. More efficient techniques for Minimal Critical Set detection need to be focused. As Constraints are being added there is a need for Consistency Enforcement system to dynamically handle this soft constraint addition.

## 6. REFERENCES

- Beddoe, G., S. Petrovic and J. Li, 2008. A hybrid metaheuristic case-based reasoning system for nurse rostering. *Springer Sci. Bus.*, 12: 99-119. DOI: 10.1007/s10951-008-0082-8
- Cheng, B.M.W, J.H.M. Lee and J.C.K. Wu, 1997. A nurse rostering system using constraint programming and redundant modeling. *IEEE Trans. Inf. Technol. Biomed.*, 1: 44-54. DOI: 10.1109/4233.594027
- Farzianpour, F., R. Askari, A.T. Hamedani, G. Khorshidi and S. Amirifar *et al.*, 2011. Accreditation of emergency department at a teaching hospital in Tehran University medical sciences in 2010. *Am. J. Econ. Bus. Admin.*, 3: 498-505. DOI: 10.3844/ajebasp.2011.498.505
- Gârbea, A., C. Suciu and F. Sisak, 2011. Design and Implementation of a fully automated planner-scheduler constraint satisfaction problem. *Proceedings of the 6th IEEE International Symposium on Applied Computational Intelligence and Informatics*, May 19-21, IEEE Xplore Press, Timisoara, pp: 477-482. DOI: 10.1109/SACI.2011.5873051
- He, F. and R. Qu, 2012. A constraint programming based column generation approach to nurse rostering problems. *Comput. Operat. Res.*, 39: 3331-3343. DOI: 10.1016/j.cor.2012.04.018
- Li, H. and A. Lim, 2003. A metaheuristic for the pickup and delivery problem with time windows. *Int. J. Artif. Intell. Tools*, 12: 173-173. DOI: 10.1142/S0218213003001186
- Lombardi, M. and M. Milano, 2009. Precedence constraint posting for RCPSP with uncertain and bounded durations with time lags and variable durations.
- Lombardi, M. and M. Milano, 2012. A min-flow algorithm for minimal critical set detection in resource constrained project scheduling. *Artif. Intell.*, 182: 58-67. DOI: 10.1016/j.artint.2011.12.001
- Muenvanichakul, S. and P. Charnsethikul, 2009. Benders' decomposition based heuristics for large-scale dynamic quadratic assignment problems. *J. Comput. Sci.*, 5: 64-70. DOI: 10.3844/jcssp.2009.64.70
- Qamal, M. and M. Ayob, 2008. A new model for examination room assignment problem: Case Study at University Kebangsaan Malaysia. *Proceedings of the International Symposium on Information Technology*, Aug. 26-28, IEEE Xplore Press, Kuala Lumpur, Malaysia, pp: 1-6. DOI: 10.1109/ITSIM.2008.4631712