The Gamut and Time Arrow of Automated Nurse Rostering

M.A. El-dosuky*

Computer Science Department, Mansoura University, Mansoura, Egypt.
e-mail: mouh_sal_010@mans.edu.eg

The date of receipt and acceptance will be inserted by the editor

Abstract There is an undeniable global shortage of skillful nurses. This is a problem of high priority, which is correlated to workforce management issues. These issues can be palliated by increasing nurses' satisfaction based on flexible rosters using automated nurse rostering. This paper in concerned with nurse rostering based on constraint programming by satisfying global constraints, such as REGULAR, which is powerful but requires devising automata of acceptable states. It was proven that REGULAR is reformulatable to SLIDE. En route to reformulate a REGULAR-based into SLIDE-based solution to nurse rostering, and through elaboration of Minizinc implementation, the author of this paper proposes a couple of metaphors namely the gamut and time arrow to model the nurse rostering. A gamut is a whole-subset of objects, such as within a given color-space or by an output device. The time arrow may refer to the direction of time as comprehended in physics. The paper in hand elucidates the new formulation based on these two metaphors, and presents implementation in Minizinc.

Keywords: Automated nurse rostering, Constraint programming, REGULAR constraint, SLIDE constraint, Gamut, Time Arrow, Minizinc

---

1 Introduction

Personnel scheduling is a core task in workforce management (Lesaint et al. (1997), Mason et al. (1998)), in many service-providing work milieus such as call centers (Canon (2007)), hospitals (Brunner (2010)), and the like.

* Corresponding author: M.A. El-dosuky, Computer Science Department, Mansoura University, 35516, Egypt.

It comes with many flavors such as considering manpower skills (Eitzen et al. (2004)), breaks (Beer et al. (2008)), days-off (Klinkert (2008)) and overlapping (Dodin and Elimam (1997)).

There is an undeniable global shortage of skillful nurses (OECD (2013)). This is a problem of high priority (Buchan and Aiken (2008)), which is correlated to workforce management issues (Organization (2006), Hayes et al. (2010)). These issues can be palliated by increasing nurses' satisfaction based on flexible rosters and strewing non-day shifts (Dunn et al. (2005)). Automated Nurse Rostering(Akai et al. (1994)) is the rescue.

Since the earlier formulation of nurse rostering (Berrada et al. (1996)), and the employment of linear programming (Jaumard et al. (1998)), a huge body of literature is flourished(Cheang et al. (2003), Burke et al. (2004))

Many algorithms and heuristics are proposed such as evolutionary (Jan et al. (2000), Aickelin and Dowsland (2004)), memetic(Burke et al. (2001)), electromagnetic (Maenhout and Vanhoucke (2007)), scatter search (Maenhout and Vanhoucke (2006)), branching strategies (Maenhout and Vanhoucke (2010)), neighborhood (Osogami and Imai (2000), Hansen et al. (2008)), column generation preferences (Bard and Purnomo (2005)).

One particular approach in vogue is the local search Tsang and Voudouris (1997) and its guided version using tabu search(Dowsland et al. (2000)).

Shift design, and its optimization (Gärtner et al. (2001)), may be considered as a preliminary step for rostering (Di Gaspero et al. (2007)).

You may refer to Beliën (2007), Tein and Ramli (2010) and Smet (2015) for recent comprehensive surveys on nurse rostering models and methodologies.

This paper in concerned with nurse rostering based on constraint programming (Soto et al. (2013)) by satisfying global constraints (Métivier et al. (2009)), such as REGULAR (Pesant (2004)) which is powerful but requires devising automata of acceptable states (Hopcroft et al. (2006)).

It was proven that REGULAR is reformulatable to SLIDE(Bessiere et al. (2007)). En route to reformulate a REGULAR-based into SLIDE-based solution to nurse rostering, and through elaboration of Minizinc implementation, the author of this paper proposes a couple of metaphors namely the gamut and time arrow to model the nurse rostering.

A gamut is a whole-subset of objects, such as within a given color-space or by an output device (Chen et al. (2004)). The time arrow may refer to the direction of time as comprehended in physics (Reichenbach (1991)).

The rest of the paper in hand is divided into the following sections. Section 2 elucidates the new formulation based on these two metaphors, while Section 3 presents implementation in Minizinc. Sections 4 and 5 are for conclusions and future work respectively.

2 Problem Formulation

2.1 Gamut

A gamut is a whole-subset of objects, such as colors within the RGB color-space (Chen et al. (2004)). A gamut can be viewed as a SET of all possible values for a VECTOR, spanning certain basis dimensions. Henceforth, it may be denoted as $\lightning$.

Example 1. The gamut of the playing-cards, $C = \lightning^{N,S}$, is the set of all possibilities of cards. A card has a number N and a suit S.

$N = \{1, 2, ...10, J, K, Q\}$ $\quad \oint N = 13$

$S = \{\clubsuit, \diamondsuit, \heartsuit, \spadesuit\}$ $\quad \oint S = 4$

The symbol used to denote the cardinality of a gamut dimension should not be confused with the closed path integral.

Note that this gamut can be decomposed, based on the second dimension, into 4 sub-gamuts namely $C_\clubsuit, C_\diamondsuit, C_\heartsuit$, and $C_\spadesuit$ each of which is $\lightning^N$

Example 2. The gamut, in nurse rostering, is a bit elusive, as it is constituted from three consecutive shifts and has an exclusion of the state of three consecutive night shifts.

So the gamut of nurse state is $G = \lightning^{S_1,S_2,S_3} - \langle night, night, night\rangle$.

$S_1 = S_2 = S_3 = \{day, night, off\}$

$\oint S_1 = \oint S_2 = \oint S_3 = 3$

Figure 2.1 shows a full probability tree of shifts. A white node denotes a day shift, while a black node denotes a night shift.

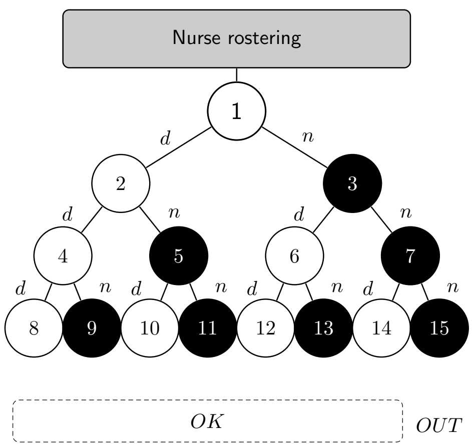

Fig. 2.1. Full probability tree of shifts

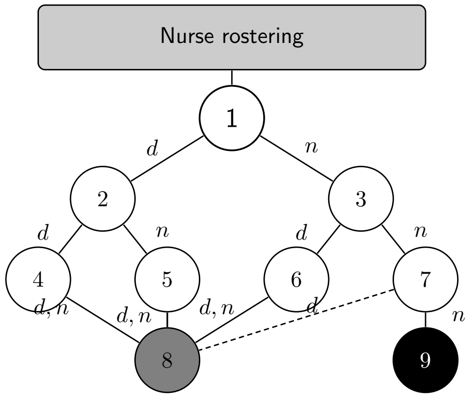

Fig. 2.2. Leaf acceptance

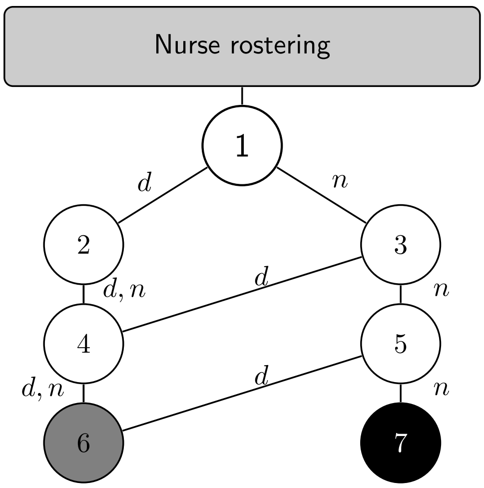

Fig. 2.3. Transition Diagram

For a matter of simplification, we assume that each dimension has only two possibilities namely the night or day shift.

Note that the gamut only contains the set of leaves except the last one that represents $\langle night, night, night \rangle$.

Now, if we group and rename all the accepted nodes, as shown in Figure 2.2, then group links then we reach to a Transition Diagram as shown in Figure 2.3.

2.2 Time Arrow

SLIDE is useful if propagated in an efficient and effective way (Bessiere et al. (2008)).

In encoding nurse rostering, dealing with time arrow $\mathbb{T}$ can be one of three ways :

- **retrospective** $\forall t, \quad 3 \leq t \leq |\mathbb{T}| \qquad |\mathbb{T}^{(t-2)} \cap (\mathbb{T}^{(t-1)} \cap \mathbb{T}^{(t)})| = 0$
- **knitting** $\forall t, \quad 2 \leq t < |\mathbb{T}| - 1 \qquad |\mathbb{T}^{(t-1)} \cap \mathbb{T}^{(t)} \cap \mathbb{T}^{(t+1)}| = 0$
- **anticipatory** $\forall t, \quad 1 \leq t \leq |\mathbb{T}| - 2 \qquad |\mathbb{T}^{(t)} \cap \mathbb{T}^{(t+1)} \cap \mathbb{T}^{(t+2)}| = 0$

They are assumedly equivalent by shifting the index of the current time. The retrospective way compares the current day with the previous two days for ensuring that no nurse has occurred in a night shift. It only looks behind in a generate-and-evaluate fashion causing high probability to backtrack. The knitting way does the same but has a look-ahead to the next day. The anticipatory way has a look-ahead to two days to come.

3 Implementation

If the total number of nurses in a certain hospital is 7 nurses, along a time arrow spanning over 10 days, then time arrow of nurse availability is something like the following:

```
1..7
1..7
1..7
1..7
1..7
1..7
1..7
1..7
1..7
1..7
```

The MiniZinc (Nethercote et al. (2007)) model to implement the retrospective way is shown in Listing 1

Listing 1. Retrospective implementation

```
include "globals.mzn";
int: num_nurses=7;       set of int: NURSES = 1..num_nurses;
int: num_days=10;        set of int: DAYS = 1..num_days;
array[DAYS] of var set of NURSES: timeArrow;
constraint forall(i in 3..num_days)(
    disjoint( timeArrow[i],
        timeArrow[i-2] intersect timeArrow[i-1] ) );
solve satisfy;
output [ show(timeArrow[i])  ++ "\n"        | i in DAYS ];
```

The corresponding time arrow is

```
1..7
1..7
{}
1..7
7..7
1..6
1..7
{}
1..7
1..7
```

This means that if all the nurses are available for the first couple days to choose from. However, if one attends in both days in a night shift, she can not attend in the third day.

The problem can be decomposed into a dichotomy of two time arrows, one for day and another for night, as shown in Listing 2. This may help in parallelizing the execution.

Notice the introduction of constraints relating to the required nurses on each day and night shifts. Each of these constraints is applied on the corresponding time arrow.

A constraint relating to the minimum night shifts per nurse is introduced too.

Listing 2. Two time arrows

```
include "globals.mzn";
int: total_nurses=7;       set of int: NURSES = 1..total_nurses;
int: total_days=10;        set of int: DAYS = 1..total_days;
int: required_days   = 3;
int: required_nights = 2;
int: minimum_nights  = 2;
array[DAYS] of var set of NURSES: timeArrow1;  % For night
array[DAYS] of var set of NURSES: timeArrow2;  % For day
constraint forall(i in 3..total_days)(
    disjoint( timeArrow1[i],
        timeArrow1[i-2] intersect timeArrow1[i-1]));
constraint forall(i in 1..total_days)(
            card( timeArrow1[i])== required_nights  /\
            card( timeArrow2[i])== required_days    /\
            disjoint( timeArrow1[i],  timeArrow2[i]) );
constraint forall(j in 1..total_nurses)(
    sum(i in 1..total_days)(
        bool2int(j in timeArrow1[i])) >= minimum_nights );
solve satisfy;
output [ show(timeArrow1[i])
    ++ " | "
    ++ show(timeArrow2[i])
    ++ "\n"
    | i in DAYS];
```

Thus, the corresponding time arrows are:

```
1..2  | 3..5
1..2  | 3..5
{3,5} | {1,2,4}
{3,5} | {1,2,4}
{4,7} | 1..3
4..5  | 1..3
5..6  | 1..3
6..7  | 1..3
4..5  | 1..3
4..5  | 1..3
```

Generating the roster is shown in Listing 3, and the generated roster is shown in Figure 3.1.

Listing 3. Generating the roster

```
include "globals.mzn";
int: total_nurses=7;     set of int: NURSES = 1..total_nurses;
int: total_days=10;      set of int: DAYS = 1..total_days;
int: required_days  =3;
int: required_nights=2;
int: minimum_nights =2;
array[DAYS] of var set of NURSES: timeArrow1;
array[DAYS] of var set of NURSES: timeArrow2;
constraint forall(i in 1..total_days)(
    card( timeArrow1[i])== required_nights  /\
    card( timeArrow2[i])== required_days    /\
    disjoint( timeArrow1[i],  timeArrow2[i]) );
constraint forall(i in 3..total_days)(
    disjoint( timeArrow1[i],
        timeArrow1[i-2]  intersect  timeArrow1[i-1]));
constraint forall(j in 1..total_nurses)(
    sum(i in 1..total_days)(
        bool2int(j in timeArrow1[i])) >= minimum_nights );
int: total_options = 3;
set of int: SHIFTS = 1..total_options;
int: day_shift = 1;
int: night_shift = 2;
int: off_shift = 3;
array[SHIFTS] of string: options = ["d","n","-"];
array[NURSES, DAYS] of var SHIFTS: roster;
constraint forall(j in 1..total_nurses)(
    forall(i in 1..total_days)(
        roster[j,i] = if j in timeArrow1[i] then night_shift
            elseif j in timeArrow2[i] then day_shift
            else off_shift endif ));
solve satisfy;
output [ options[fix(roster[i,j])]
         ++ if j==total_days then "\n" else " " endif
       |  i in NURSES, j in DAYS ];
```

	Day1	Day2	Day3	Day4	Day5	Day6	Day7	Day8	Day9	Day10
Nurse1					Night			Night		
Nurse2					Night			Night		
Nurse3	Day	Day	Night	Night		Night	Night		Night	Night
Nurse4	Day	Day	Day	Day		Night	Night		Night	Night
Nurse5	Day	Day	Night	Night	Day	Day	Day	Day	Day	Day
Nurse6	Night	Night	Day	Day	Day	Day	Day	Day	Day	Day
Nurse7	Night	Night	Day	Day	Day	Day	Day	Day	Day	Day

Fig. 3.1. Roster

4 Conclusions

En route to reformulate a REGULAR-based into SLIDE-based solution to nurse rostering by simplifying the full probability tree of shifts, the author of this paper proposes a the metaphor of gamut. A gamut may be compared to set-covering formalization that may be contributed to Dantzig (1954) in his elaboration on the possibility of using linear programming in scheduling in a paper entitled "Traffic Delays at Toll Booths" (Edie (1954)). His formalization assumes a variable for every possible shift.

A gamut, on the other hand, provides a compact nomenclature for a SET of all possible values for a VECTOR, spanning certain basis dimensions.

But, the gamut, as seen in nurse rostering example, can be a bit elusive, as it is constituted from three consecutive shifts and has an exclusion of the state of three consecutive night shifts.

Through elaboration of Minizinc implementation, time arrow metaphor is presented and exploited. Decomposing the time arrow into a two or more arrows, may help in parallelizing the execution by applying constraints on corresponding time arrows.

5 Future Work

There could be specialized propagators for the knitting and the anticipatory ways to be proposed.

yet, one possible future direction is to attempt generate fair rosters (Martin et al. (2013))

Another direction is to extend the proposed work of nurse-to-shift assignment, by considering other processes of planning decisions in health care(Hulshof et al. (2012)).

A philosophical future direction may be the consideration of bewildering aspects of the time arrow such as the reversibility (Reichenbach (1991)), and paradoxes (Clark (2012)).

Compliance with Ethical Standards

Conflict of interest The author declares that he has no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by the author.

References

Aickelin, U. and Dowsland, K. A. (2004). An indirect genetic algorithm for a nurse-scheduling problem. *Computers & Operations Research*, 31(5):761–778.

Akai, Y., Iwata, R., Ozawa, M., Honda, M., Suzuki, T., and Satomura, Y. (1994). Automated nurse rostering system for administrative efficiency as a subsystem of a total hospital information system. *Medinfo. MEDINFO*, 8:1417–1417.

Bard, J. F. and Purnomo, H. W. (2005). Preference scheduling for nurses using column generation. *European Journal of Operational Research*, 164(2):510–534.

Beer, A., Gärtner, J., Musliu, N., Schafhauser, W., and Slany, W. (2008). Scheduling breaks in shift plans for call centers. In *Proc. of the 7th Int. Conf. on the Practice and Theory of Automated Timetabling, Montréal, Canada.*

Beliën, J. (2007). Exact and heuristic methodologies for scheduling in hospitals: problems, formulations and algorithms. *4OR: A Quarterly Journal of Operations Research*, 5(2):157–160.

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

Bessiere, C., Hebrard, E., Hnich, B., Kiziltan, Z., Quimper, C.-G., and Walsh, T. (2007). Reformulating global constraints: The slide and regular constraints. In *International Symposium on Abstraction, Reformulation, and Approximation*, pages 80–92. Springer.

Bessiere, C., Hebrard, E., Hnich, B., Kiziltan, Z., and Walsh, T. (2008). Slide: A useful special case of the cardpath constraint. In *18th European Conf. on Artificial Intelligence*. IOS Press.

Brunner, J. (2010). *Flexible shift planning in the service industry: The case of physicians in hospitals*, volume 640 of *Lecture notes in economics and mathematical systems*. Springer.

Buchan, J. and Aiken, L. (2008). Solving nursing shortages: a common priority. *Journal of clinical nursing*, 17(24):3262–3268.

Burke, E., Cowling, P., De Causmaecker, P., and Berghe, G. V. (2001). A memetic approach to the nurse rostering problem. *Applied intelligence*, 15(3):199–214.

Burke, E. K., De Causmaecker, P., Berghe, G. V., and Van Landeghem, H. (2004). The state of the art of nurse rostering. *Journal of scheduling*, 7(6):441–499.

Canon, C. (2007). *Personnel scheduling in the call center industry*. PhD thesis, Springer.

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

Chen, Y., Berns, R. S., and Taplin, L. A. (2004). Extending printing color gamut by optimizing the spectral reflectance of inks. In *Color and Imaging Conference*, volume 2004, pages 163–169. Society for Imaging Science and Technology.

Clark, M. (2012). *Paradoxes from A to Z*. Routledge.

Dantzig, G. B. (1954). Letter to the editora comment on edie's traffic delays at toll booths. *Journal of the Operations Research Society of America*, 2(3):339–341.

Di Gaspero, L., Gärtner, J., Kortsarz, G., Musliu, N., Schaerf, A., and Slany, W. (2007). The minimum shift design problem. *Annals of operations research*, 155(1):79–105.

Dodin, B. and Elimam, A. (1997). Audit scheduling with overlapping activities and sequence-dependent setup costs. *European Journal of Operational Research*, 97(1):22 – 33.

Dowsland, K. A., Thompson, J. M., et al. (2000). Solving a nurse scheduling problem with knapsacks, networks and tabu search. *Journal of the Operational Research Society*, 51(7):825–833.

Dunn, S., Wilson, B., and Esterman, A. (2005). Perceptions of working as a nurse in an acute care setting. *Journal of Nursing management*, 13(1):22–31.

Edie, L. C. (1954). Traffic delays at toll booths. *Journal of the operations research society of America*, 2(2):107–138.

Eitzen, G., Panton, D., and Mills, G. (2004). Multi-skilled workforce optimisation. *Annals of Operations Research*, 127(1):359–372.

Gärtner, J., Musliu, N., and Slany, W. (2001). Rota: a research project on algorithms for workforce scheduling and shift design optimization. *AI Communications*, 14(2):83–92.

Hansen, P., Mladenović, N., and Pérez, J. A. M. (2008). Variable neighbourhood search: methods and applications. *4OR*, 6(4):319–360.

Hayes, B., Bonner, A., and Pryor, J. (2010). Factors contributing to nurse job satisfaction in the acute hospital setting: a review of recent literature. *Journal of Nursing Management*, 18(7):804–814.

Hopcroft, J. E., Motwani, R., and Ullman, J. D. (2006). Automata theory, languages, and computation. *International Edition*, 24.

Hulshof, P. J., Kortbeek, N., Boucherie, R. J., Hans, E. W., and Bakker, P. J. (2012). Taxonomic classification of planning decisions in health care: a structured review of the state of the art in or/ms. *Health systems*, 1(2):129–175.

Jan, A., Yamamoto, M., and Ohuchi, A. (2000). Evolutionary algorithms for nurse scheduling problem. In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, volume 1, pages 196–203. IEEE.

Jaumard, B., Semet, F., and Vovor, T. (1998). A generalized linear programming model for nurse scheduling. *European journal of operational research*, 107(1):1–18.

Klinkert, A. (2008). Days-off scheduling in large-scale multi-skill staff rostering: an integer programming solution. In *Proceedings of the 7th International Conference on the Practice and Theory of Automated Timetabling (PATAT08)*.

Lesaint, D., Voudouris, C., Azarmi, N., and Laithwaite, B. (1997). Dynamic workforce management.

Maenhout, B. and Vanhoucke, M. (2006). New computational results for the nurse scheduling problem: a scatter search algorithm. In *EvoCOP*, pages 159–170. Springer.

Maenhout, B. and Vanhoucke, M. (2007). An electromagnetic meta-heuristic for the nurse scheduling problem. *Journal of Heuristics*, 13(4):359–385.

Maenhout, B. and Vanhoucke, M. (2010). Branching strategies in a branch-and-price approach for a multiple objective nurse scheduling problem. *Journal of Scheduling*, 13(1):77–93.

Martin, S., Ouelhadj, D., Smet, P., Berghe, G. V., and ÖZcan, E. (2013). Cooperative search for fair nurse rosters. *Expert Systems with Applications*, 40(16):6674–6683.

Mason, A. J., Ryan, D. M., and Panton, D. M. (1998). Integrated simulation, heuristic and optimisation approaches to staff scheduling. *Operations research*, 46(2):161–175.

Métivier, J.-P., Boizumault, P., and Loudni, S. (2009). Solving nurse rostering problems using soft global constraints. In *International Conference on Principles and Practice of Constraint Programming*, pages 73–87. Springer.

Nethercote, N., Stuckey, P., Becket, R., Brand, S., Duck, G., and Tack, G. (2007). Minizinc: Towards a standard cp modelling language. *Principles and Practice of Constraint Programming–CP 2007*, pages 529–543.

OECD (2013). *Health at a glance 2013: OECD indicators*. OECD publishing.

Organization, W. H. (2006). *The world health report 2006: working together for health*. World Health Organization.

Osogami, T. and Imai, H. (2000). Classification of various neighborhood operations for the nurse scheduling problem. *Algorithms and computation*, pages 72–83.

Pesant, G. (2004). A regular language membership constraint for finite sequences of variables. *CP*, 3258:482–495.

Reichenbach, H. (1991). *The direction of time*, volume 65. Univ of California Press.

Smet, P. (2015). Nurse rostering: models and algorithms for theory, practice and integration with other problems.

Soto, R., Crawford, B., Monfroy, E., Palma, W., and Paredes, F. (2013). Nurse and paramedic rostering with constraint programming: A case study. *Romanian Journal of Information Science and Technology*, 16(1):52–64.

Tein, L. H. and Ramli, R. (2010). Recent advancements of nurse scheduling models and a potential path. In *Proc. 6th IMT-GT Conference on Mathematics, Statistics and its Applications (ICMSA 2010)*, pages 395–409.

Tsang, E. and Voudouris, C. (1997). Fast local search and guided local search and their application to british telecom's workforce scheduling problem. *Operations Research Letters*, 20(3):119–127.