北見工業大学研究報告　Memoirs of the Kitami Institute of Technology
第22巻 第2号 (平成3年)　Vol. 22 No. 2 (1991)

An Expert System Design for Alternating Job Shift Scheduling*

Yoshitaka FUJIWARA** and Toshiharu MATSUNISHI**

(Receiverd, October 5, 1990)

Abstract

In around-the-clock services such as hospital patient care, workers must be assigned fairly to different job types (called shifts), which require various number of workers for a given period of time. This paper proposes a design for an expert system to do this scheduling, under the condition of assignment-inhibiting irregular events such as sick leaves. In this design, the hierarchical problem-solving approach of a human expert is incorporated as a heuristic to reduce the number of searches required to find an optimum solution. That is, a rough solution is derived under the dominant constraints using an algorithmic assignment method called ARA followed by leveling of the resulting unbalanced work load by a trial-and-error replacement method of workers called MMR.

1. Introduction

In modern society such services as patient nursing in the hospital or building patroling for security have become important, in which around-the-clock service or almost its equivalent is required each working day.

Therefore, the total working time of each day is divided into several periods usually called shifts and each worker is scheduled to work in alternating shifts (day shift, night shift, etc.). It is also quite important for the job manager to make equal (fair) alternating shift schedules for all the workers. However, the difficulty of equal scheduling increases rapidly as the number of workers or the number of working days increase and when each worker's day off requirement (an assignment inhibiting condition) is considered.

This paper proposes a hierarchical scheduling strategy for a rule-based expert sysetm (abbreviated to ES hereafter)[1] under the assignment inhibiting conditions mentioned above.

2. Hierarchical scheduling strategy

2.1 Outline

A human expert usually takes a hierarchical approach to solve a problem when he tries to design or plan something. That is, he first tries to find a rough solution which satisfies only the major constraints. Then he tries to improve the tentative solution to an optimum one satisfying all constraints by

* Presented at the 3rd Annual Conference of JSAI (July 1989)

** Department of Electronic Engineering, Kitami Institute of Technology

the trial-and-error method.

The scheduling strategy in this paper adopts this approach as an effective heuristic to reduce the number of search operations necessary to find an optimum

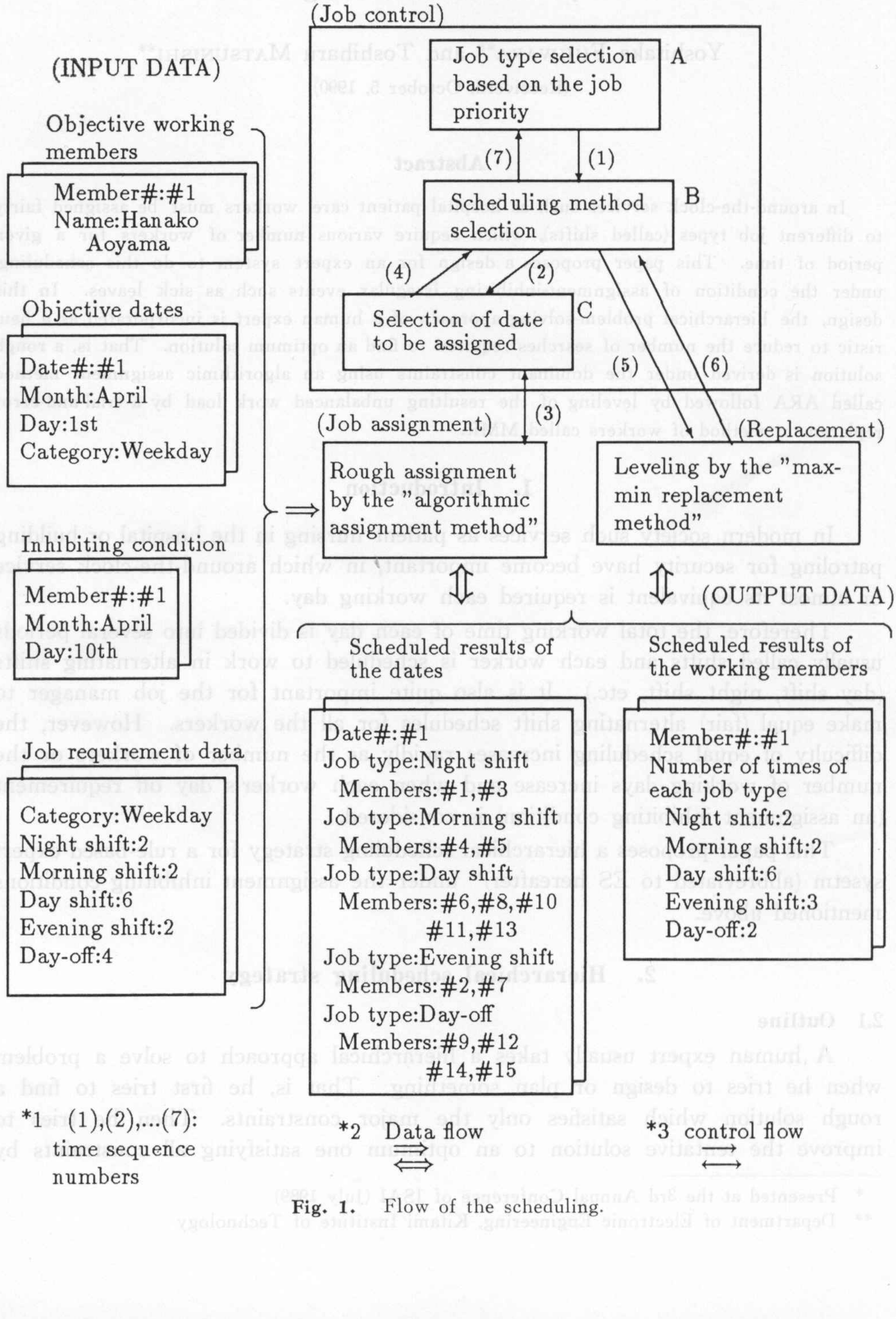

Fig. 1. Flow of the scheduling.

solution. The first step is to obtain a rough solution, called "Algorithmic assignment method" (abbreviated to ARA) and the second step is to derive an optimum one as "Max-Mini replacement method" (abbreviated to MMR)[2].

The ARA randomly selects candidates from the workers satisfying the following conditions for each objective working date under the targeted job shift (e. g. night shift) : (a) with no inhibiting condition on the targeted working date, (b) with the least number of assignments up to this time concerning the targeted shift, and (c) with a longer time period than the pre-determined threshold in relation to the interval between the targeted working date and the most recently assigned working date.

Possible combinations of the scheduling covering all job shifts can be represented by a "search tree," in which each node represents a specific assignment which determines a specific combination of the worker, the objective working date and the job shift. Furthermore obtaining a rough solution by the ARA corresponds to the selection of a specific path from the root node to a terminal node of the tree.
The ARA is formalized in Sect. 2.2.

On the other hand, the MMR first forms two groups of workers, max-group and mini-group ; members of the former group have the maximum value, and the members of the latter one have the minimum value concerning the number of assignments, respectively. Next, the MMR tries to replace the assignments of the max-group members by the min-group members in order to balance the assignments derived using the ARA. The replacement of a max-group member by a min-group member corresponds to a replacement of the partial path on the specific path on the search tree (selected by the ARA) by a new path starting from the node resulting from the replacement. The MMR is formalized in Sect. 2. 3.

The flow of the overall job scheduling is shown in Fig. 1. In Fig. 1, the workers, the dates, the required number of workers for each job shift and the assignment inhibiting conditions are inputted. Then, two kinds of results, "scheduled results of the dates" and "scheduled results of the working members," are produced through the processes of the two-staged scheduling strategy mentioned above.

2.2 Algorithmic assignment method (ARA)

(**Definitions**) T: a set of all workers, Z_j: a set of workers with the inhibiting conditions on the j-th date, M: number of workers ($|T|$), R_{ik} : number of assignments of the k-th shift for the i-th worker, F_{ij} : the shift assigned for the i-th worker on the j-th date, n : a serial number of the targeted date, k : an identification number of the targeted shift (1: night shift 2: morning shift 3: day off-1 4: day shift 5: evening shift 6: day off-2), $\min\{y_i\}$: a function for selecting elements having the smallest value, θ: a constant determining the minimum interval between consecutive assignments for the targeted shift.

(**Rules for selecting candidates**)

(1) night shift

$$\exists x \in T\left[(x \notin Z_j \middle| j=n, n+1) \wedge (F_{xj} \neq \text{"night shift"} \middle| j=n-\theta, \cdots, n-1) \wedge (F_{xn}=nil) \wedge \left(R_{x1}=\min\{R_{j1} \middle| j=1, \cdots, M\}\right)\right]$$

(2) morning shift

$$\exists x \in T\left[(F_{x,n-1}=\text{"night shift"}) \wedge (x \notin Z_n) \wedge (F_{xn}=nil)\right]$$

(3) day-off-1

$$\exists x \in T\left[(F_{x,n-1}=\text{"morning shift"}) \wedge (x \notin Z_n) \wedge (F_{xn}=nil)\right]$$

(4) day shift

$$\exists x \in T\left[(F_{xn}=nil) \wedge (x \notin Z_n) \wedge \left(R_{x4}=\min\{R_{j4} \middle| j=1, \cdots, M\}\right)\right]$$

(5) evening shift

$$\exists x \in T\left[(F_{xn}=nil) \wedge (x \notin Z_n) \wedge \left(R_{x5}=\min\{R_{j5} \middle| j=1, \cdots, M\}\right)\right]$$

(6) day-off-2

$$\exists x \in T\left[(F_{xn}=nil) \wedge (x \notin Z_n) \wedge \left(R_{x6}=\min\{R_{j6} \middle| j=1, \cdots, M\}\right)\right]$$

Fig. 2 shows an example of the candidate-selecting rule for the night shift by the KBMS[3]. The classes appearing in Fig. 2 and the meanings of their slots are shown in Table 1. By the first if-then rule in Fig. 2, the most adequate candidate is selected. Then, by the second if-then rule, a new data element for the "scheduled results of the dates" is added and the existing data element of the "scheduled results of the working members" is updated for this candidate. The execution of the paired if-then rules is iterated by the number of times which is equal to the number of workers required for the targeted shift. Other candidate selecting rules for the day shift, evening shift, and day-offs, can be represented like Fig. 2.

2.3 Max-Mini replacement method (MMR)

(**Definitions**)

$X=\{x_1, x_2, \cdots, x_m\}$: a list of workers having the largest value concerning the number of assignments on the targeted shift,

$Y=\{y_1, y_2, \cdots, y_n\}$: a list of workers having the smallest value concerning the number of assignments on the targeted shift,

$P_x(x_i)=\{p_{i1}, p_{i2}, \cdots, p_{im}\}$: a list of assigned dates of i-th worker concerning the targeted shift,

$P_y(y_j)=\{p_{j1}, p_{j2}, \cdots, p_{jn}\}$: a list of assigned dates of i-th worker concerning the targeted shift,

$T_y(y_j)=\{t_{j1}, t_{j2}, \cdots, t_{jh}\}$: a list of assigned dates of j-th worker concerning all shifts except the targeted shift.

(**Replacement procedure**)

The initial state of this procedure is a result derived by the ARA.

Let $S_0=\{X, Y, \{P_x(x_i)|x_i\in X\}, \{P_y(y_i)|y_i\in Y\}, \{T_y(y_j)|y_j\in Y\}\}$ be the expression

```
[RULE-NAME:NIGHT-SHIFT RULE1]
(FRAME (ASSIGN ? (SHIFT NIGHT-SHIFT) (S-NUMBER ?ZA)))
(FRAME (DCURRENT ? (DATE# ?XA) (CATEGORY ?XB)))
(FRAME (JNEED ? (CATEGORY ?XB) (SHIFT NIGHT-SHIFT)
       (R-NUMBER ?MAX !(< ?ZA ?MAX))))
(FRAME (MTOKEN ? (ID# ?YA) (T-NUMBER ?YB)
       (NS-NUMBER ?YC)))
(NOT (FRAME (MHISTORY ? (ID# ?YA)
       (DATE# ?XA)))
(NOT (FRAME (MHISTORY ? (ID# ?YA)
       (DATE# !(1+ ?XA)) (SHIFT DAY-OFF))))
(NOT (FRAME (MHISTORY ? (ID# ?YA)
       (DATE# !(- ?XA 1)) (SHIFT NIGHT-SHIFT))))
(NOT (FRAME (MHISTORY ? (ID# ?YA)
       (DATE# !(- ?XA 2)) (SHIFT NIGHT-SHIFT))))
(FRAME (WORK ?JJ (ID# ?JA) (T-NUMBER ?JB)
       (DATE# ?JC)))
(TEST (OR (< ?YB ?JB) (AND (= ?YB ?JB)
       (< ?YC ?JC))))
-->
(MODIFY ?JJ (ID# ?YA) (T-NUMBER ?YB)
       (DATE# ?YC))

[RULE-NAME:NIGHT-SHIFT RULE2]
(FRAME (ASSIGN ?ZZ (SHIFT NIGHT-SHIFT)
       (S-NUMBER ?ZA)))
(FRAME (DCURRENT ? (DATE# ?XA) (CATEGORY ?XB)))
(FRAME (JNEED ? (CATEGORY ?XB) (SHIFT NIGHT-SHIFT)
       (R-NUMBER ?MAX! (< ?ZA ?MAX))))
(FRAME (WORK ?WW (ID# ?YA)))
(FRAME (MTOKEN ?YY (ID# ?YA) (NS-NUMBER ?YD)
       (T-NUMBER ?YB)))
(FRAME (RANGE ? (DATE# ?NA)))
-->
(CREATE MHISTORY (ID# ?YA) (DATE# ?XA)
       (CATEGORY ?XB) (SHIFT NIGHT-SHIFT))
(MODIFY ?YY (NS-NUMBER !(1+ ?YD)) (NS-DATE# ?XA)
       (T-NUMBER !(1+ ?YB)))
(MODIFY ?ZZ (S-NUMBER !(1+ ?ZA)))
(MODIFY ?WW (ID# 0) (T-NUMBER ?NA)
       (DATE# ?NA))
```

Fig. 2. An example of the if-then rules performing the night shift assignment.

Table 1. Frame types and the meanings of their slots

Frame types	Slot names and their meanings
ASSIGN	SHIFT:targeted shift, S−NUMBER:accumulated number of workers selected
DCURRENT	DATE#:serial number of the targeted date, CATEGORY:weekday/holiday
JNEED	SHIFT:targeted shift, CATEGORY:weekday/holiday, R−NUMBER:required number of workers for the targeted shift
RANGE	DATE#:serial number of the last date of the term
WORK	ID#:identification number of a worker selected, T−NUMBER:total number of assignments of the worker concerning all shifts except day−offs, DATE#:serial number of the date assigned most recently for the targeted shift
MTOKEN	ID#:identification number of a worker, NS−NUMBER:number of assignments of the worker concerning the night shift, NS−DATE#:serial number of the date assigned most recently of a worker concerning the night shift, T−NUMBER:total number of assignments of a worker concerning all shifts except day−offs
MHISTORY	DATE#:serial number of the targeted date, SHIFT:targeted shift, ID#:identificateion number of the worker selected

of the initial state.

First, a 3-tuple $(\bar{x}_i, \bar{y}_j, \bar{p}_k)$ satisfying the following condition is searched.

$$\exists \bar{x}_i \in X \exists \bar{y}_j \in Y \exists \bar{p}_k \in P_x(\bar{x}_i)\Big[\bar{p}_k \notin P_y(\bar{y}_j) \wedge \bar{p}_k \notin T_y(\bar{y}_j) \wedge \big|P_y(\bar{y}_j)\big| < Av \wedge \big|P_x(\bar{x}_i)\big| > Av\Big] \qquad (1)$$

where Av : the largest integer which does not exceed the value of $(V_{\max} + V_{\min})/2$,

$V_{\max}$: the largest values among the numbers of assignments of all workers selected,

$V_{\min}$: a smallest value among the numbers of assignments of all workers selected.

When plural 3-tuples are found, one of them is chosen arbitrarily.

Next, a new state S_1 is generated using the elements $(\bar{x}_i, \bar{y}_j, \bar{p}_k)$ by the Eq. (2).

$$S_1 = \Big\{X, Y, \Big\{P'_x(x_i)\Big|x_i \in X\Big\}, \Big\{P'_y(y_j)\Big|y_j \in Y\Big\}, \Big\{T(y_j)\Big|y_j \in Y\Big\}, \Big\} \qquad (2)$$

Where $P'_x(x_i) = \{p_{ij} | p_{ij} \neq \bar{p}_k < p_{ij} \in P_x(x_i)\}$ for $x_i = \bar{x}_i$,

$P'_x(x_i) = \{p_{ij} | p_{ij} \in P_x(x_i)\}$ for $x_i \neq \bar{x}_i$,

$P'_y(y_j) = \{\bar{p}_k\} \cup P_y(y_j)$ for $y_j = \bar{y}_j$,

$P'_y(y_j) = P_y(y_j)$ for $y_j \neq \bar{y}_j$.

(**Qualification check for an optimum solution**)

The newly generated state S_1 is investigated to find whether it satisfies the following conditions required for an optimum solution.

$$\forall x_i \in X \forall y_j \in Y\left[\left|P'_x(x_i)\right| = Av \wedge \left|P'_y(y_j)\right| = Av\right] \tag{3}$$

When the S_1 does not satisfy the condition, the replacement procedure given by Eqs. (1) and (2) is repeated. Further, when an optimum solution is not obtained after performing possible replacements, a state with least deviation from the ideal state satisfying Eq. (3) is chosen as a quasi-optimum solution among all states generated by the replacement procedure.

3. Performance evaluation

In order to assess the performance of the scheduling strategy proposed, a prototyping ES is produced using the knowledge-base designing tool KBMS[3]. Here, two measures were used for evaluating performance; one is the time required for the scheduling, the other is the percentage of the successful assignments which achieve equal alternating shifts for all the workers. Six job shifts were chosen as a job shift model and the priority of assigning order between then is set as: (1) night shift (2) morning shift (3) say-off-1 (followed by the morning job shift) (4) day shift (5) evening shift (6) day-off-2 (ordinary day-off). And the ratio of the required number of workers for each job shift is set as follows: 1 for the night shift, 1 for the day-off-1, 1 for the evening shift, 3

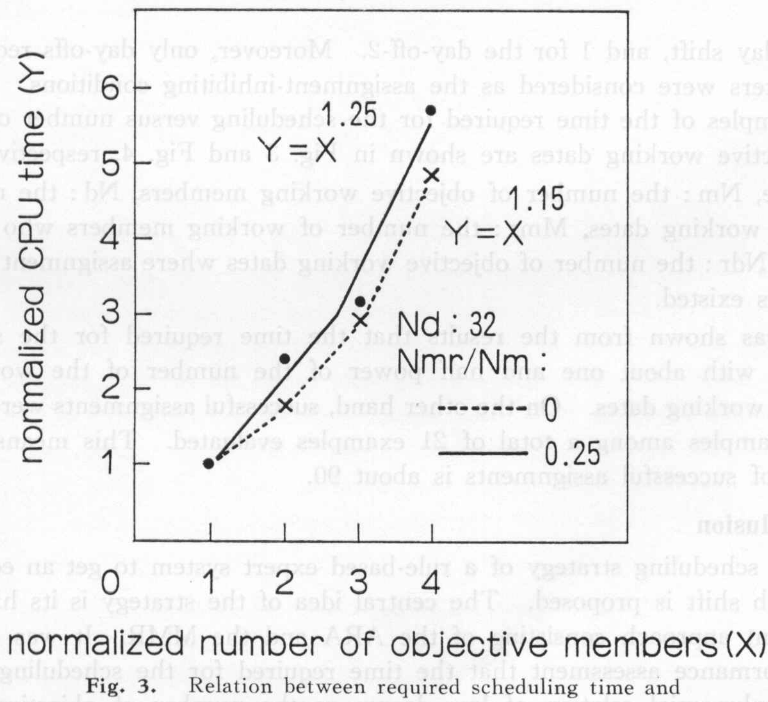

Fig. 3. Relation between required scheduling time and number of objective members.

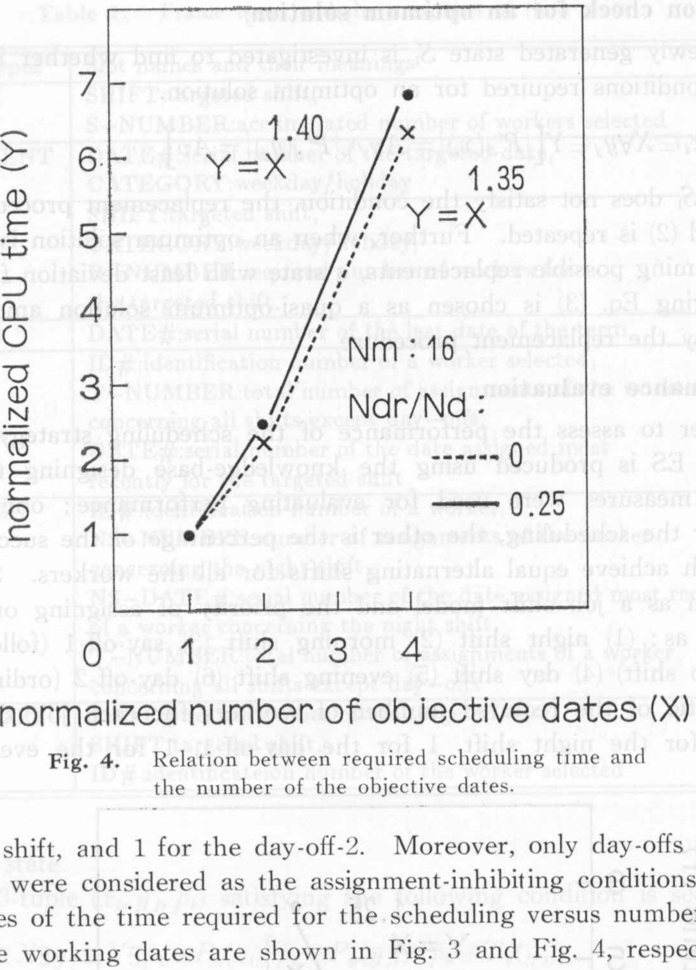

Fig. 4. Relation between required scheduling time and the number of the objective dates.

for the day shift, and 1 for the day-off-2. Moreover, only day-offs requested by the workers were considered as the assignment-inhibiting conditions.

Examples of the time required for the scheduling versus number of workers and objective working dates are shown in Fig. 3 and Fig. 4, respectively.

Here, Nm : the number of objective working members, Nd : the number of objective working dates, Mmr : the number of working members who requested day-offs, Ndr : the number of objective working dates where assignment inhibiting conditions existed.

It was shown from the results that the time required for the scheduling increases with about one and half power of the number of the workers and objective working dates. On the other hand, successful assignments were obtained in 19 examples among a total of 21 examples evaluated. This means the percentage of successful assignments is about 90.

4. Conclusion

The scheduling strategy of a rule-based expert system to get an equal alternating job shift is proposed. The central idea of the strategy is its hierarchical assignment approach consisting of the ARA and the MMR. It was shown by the performance assessment that the time required for the scheduling increases with a polynomial relation of low degree as the number of objective working members and the number of objective working dates increase and it was shown that the percentage of successful assignments is about 90.

References

1) J. Martin and S. Oxman: "Building Expert Systems (A Tutorial), Prenticehall (1988).
2) Y. Fujiwara: "Staff Scheduling Technique for the Around-the-Clock Services", Proceedings of the 3rd Annual Conference of JSAI, pp. 601–604 (1989).
3) NTT software edition: KBMS/PC Reference manuals, NTT software inc. (1989).