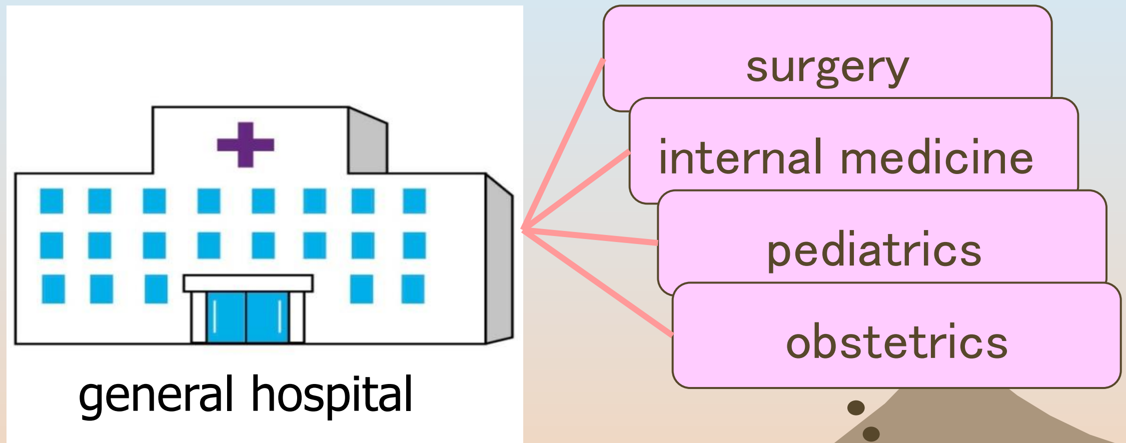
Nurse Scheduling By Cooperative GA with Penalty Coefficient Adjustment

Department of Information & Electronics
Graduate School of Engineering, Tottori University

1. Introduction



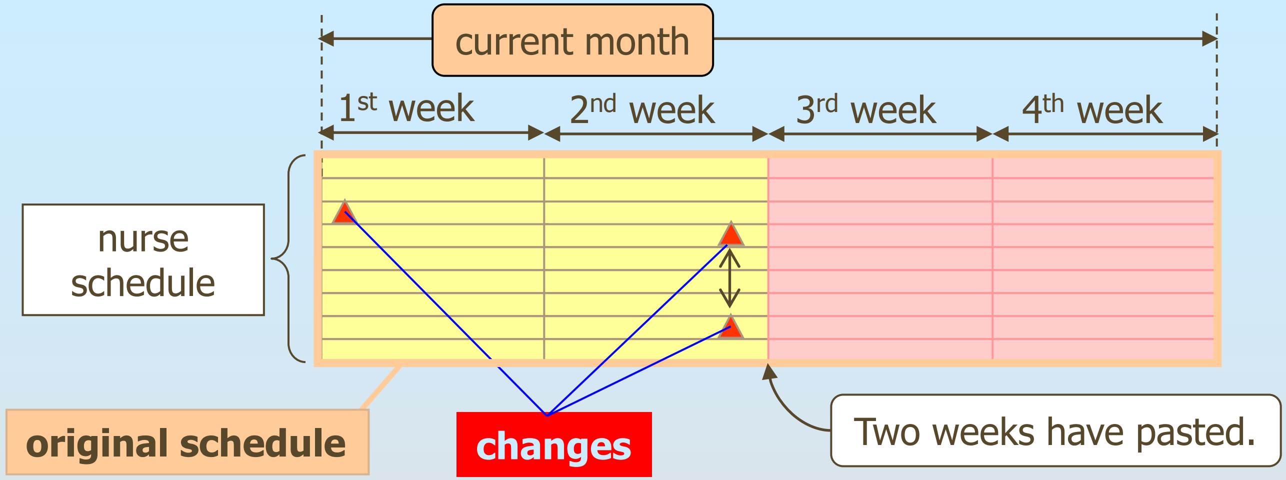
- A section manager must arranges a shift schedule for nurses in the section.
- In our investigation, even veteran manager needs one or two weeks for the nurse scheduling. This means a great loss of work force and time. Therefore, computer software for the nurse scheduling is strongly required recently.
- We know there are several commercial software to generate nurse schedule. However, they are not used, because they cannot generate a schedule available in practical use.
- That is, the optimized result of the schedule is unsatisfactory.

many requirements

- requirements on the hope holiday.
- duty load in equality.
- the number of the night shift in equality.
- affinity between the nurses in the night shift.
- etc. ・・・・

- We discuss a technique to apply the cooperative GA (CGA) to generate & optimize the nurse schedule.
- The conventional CGA searches solutions **only by using crossover operator**, because it is considered as the only one operator keeping consistency of the population.
- A **mutation** changing small part of the population brings very important modification to the population.

- We have proposed an **effective mutation operator** activated depending on the optimization speed [M.Ohki2006].
- We have proposed a mutation operator activated periodically, **Periodic Mutation**, defined by fewer parameter [M.Ohki2007]. The periodic mutation is advantage in that it needs fewer parameters to define itself and it brings almost similar result as the early mutation technique.
- We have also proposed a **parallel mutation technique** [M.Ohki2010a, M.Ohki2010b].



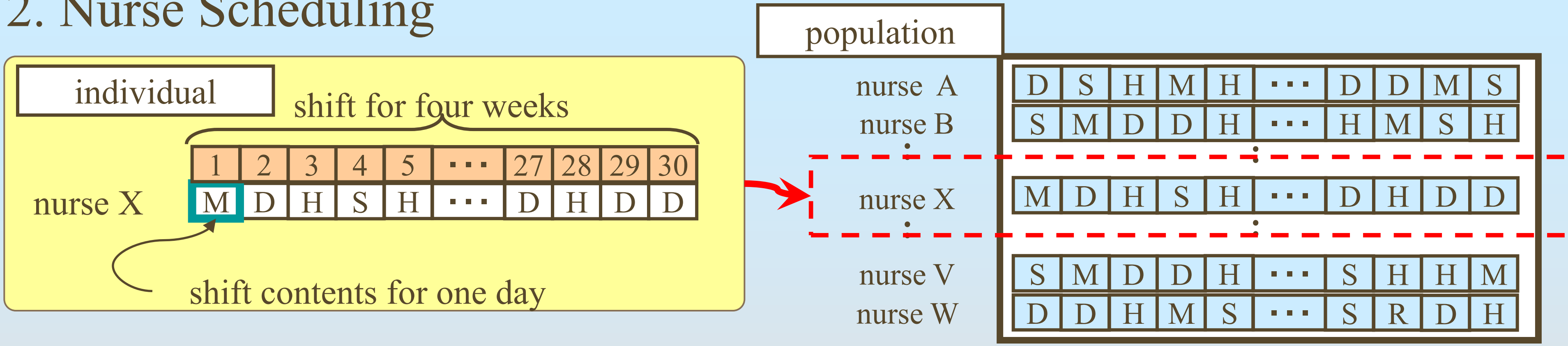
- In the actual cases, there are the cases that nurses originally assigned to a rest are forced to attendance by means of emergency. There are also the cases that a nurse whom duty has been assigned originally takes a rest due to a disease.
- By means of such the **changes**, several **inconvenience** occurs, for example, imbalance of the number of the holidays/attendance, etc.
- Such an **inconvenience** causes the fall of the nursing level of the whole nurse organization.
- The changed schedule must be **re-optimized** to break off the **inconvenience** as much as possible.

- The **re-optimization** such the schedule is difficult even by using the effective mutation operator and the parallel processing.

in this research,・・・

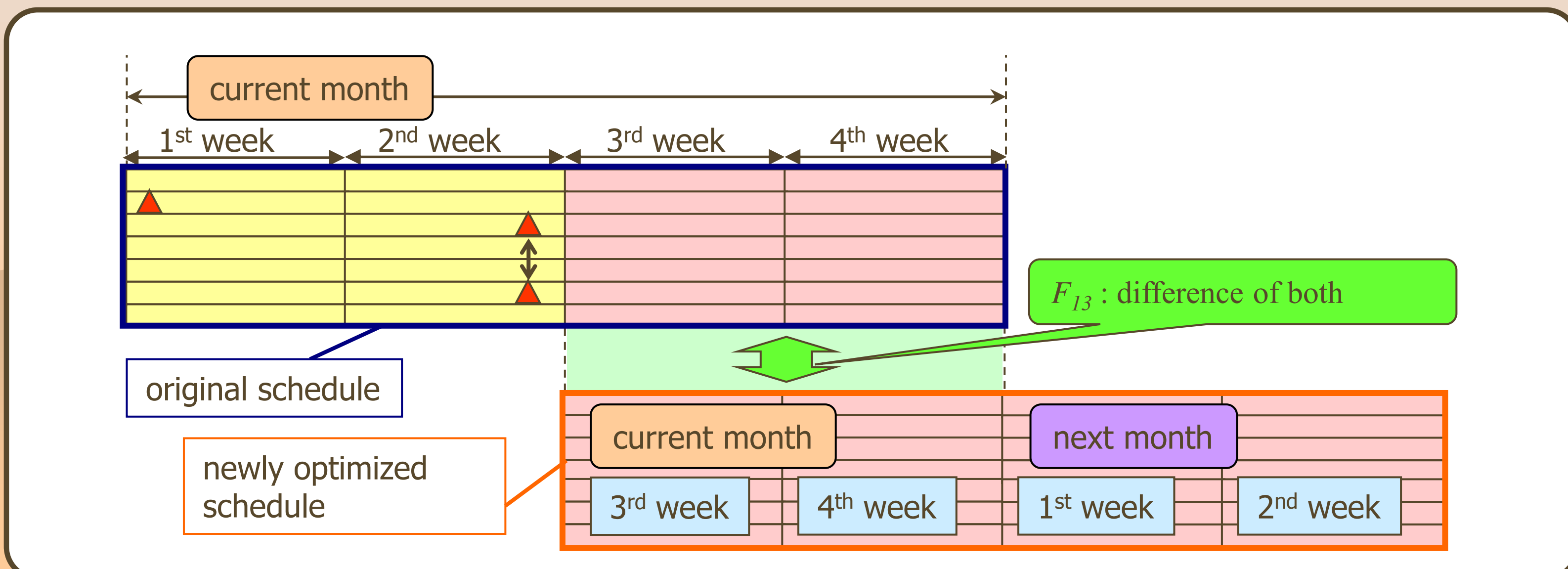
Penalty coefficient Adjustment

2. Nurse Scheduling



- In the CGA, there are not two or more individuals giving the same nurse's schedule.
- For the nurse scheduling, the manager considers many requirements. These requirements are defined as 13 penalty functions.

$$E = \sum_{i=1}^{M} \sum_{k=1}^{6} h_k F_{ki} + \sum_{j=1}^{D} \sum_{k=7}^{12} h_k F_{kj} + h_{13} F_{13} \qquad (1)$$



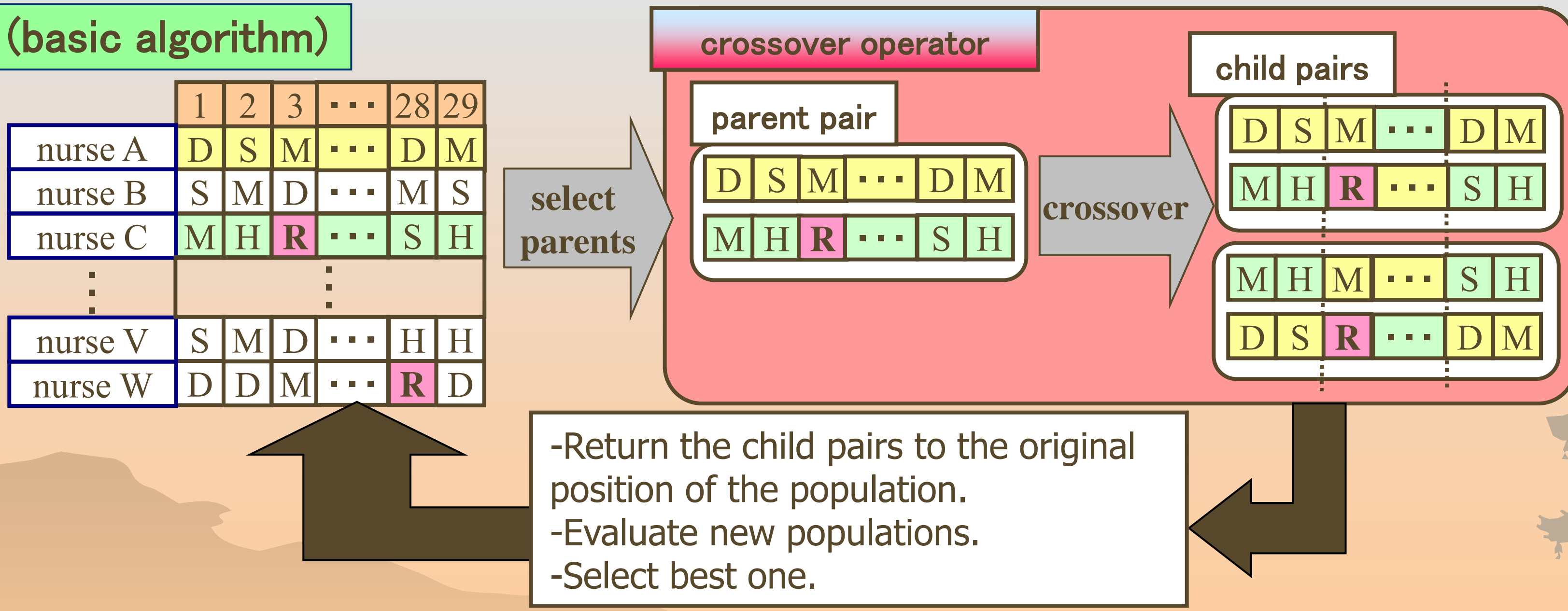
3. Basic Algorithm of CGA

D: day shift, S: semi-night shift,
M: midnight shift, R: requested holiday,
H: holiday, m: meeting,
T: training

CGA (initialization)

- First, CGA initialize the population.
- The requested holiday (R), the meeting (m) and the training (T) are treated as the fixed duty, which CGA does not move them.
- CGA put them onto the population initially.
- We suppose that the number of nurses in the day time, the semi-night and the midnight shift are defined as 6, 3 and 3 respectively in the application here.
- CGA randomly assigns the duty symbols satisfying the specific numbers.

CGA (basic algorithm)



- This procedure is applied to **100 parent pairs** in 1 generation.
- Therefore, 200 new populations are (locally) searched around the original population.

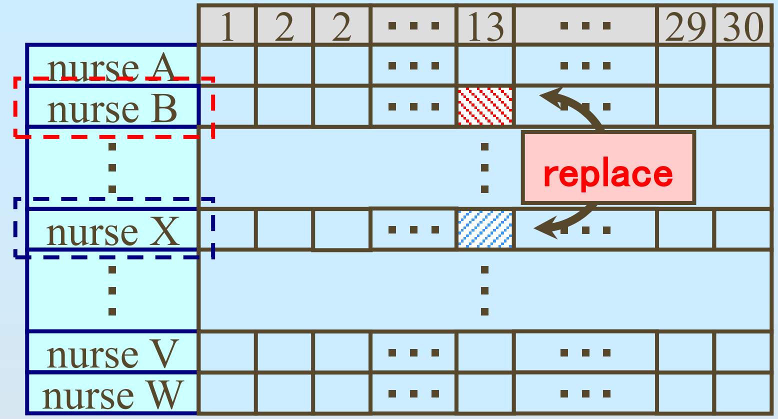
CGA (mutation operator)

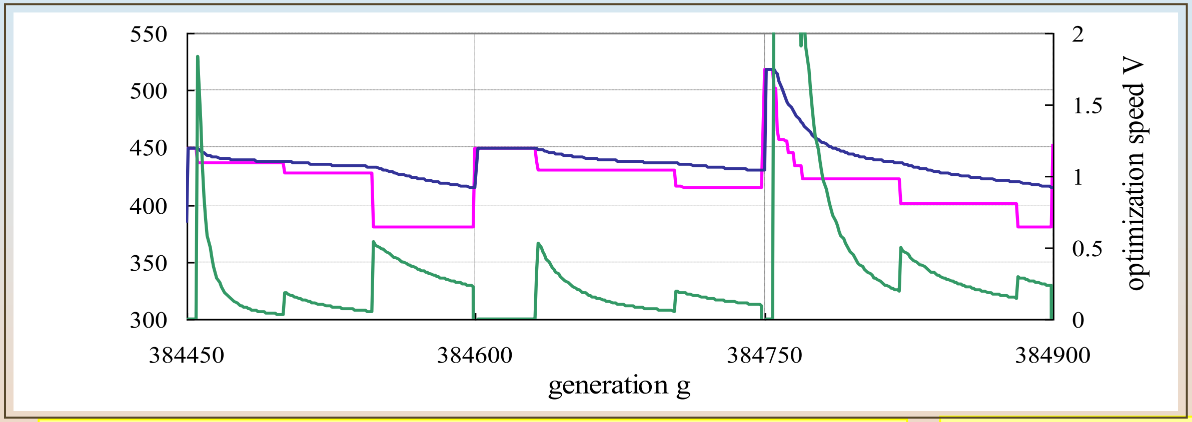
When the mutation is activated, ･･･.

① Randomly select one date.

② Randomly select two nurses. If one of them or both two are fixed shift, return to ①.

③ Replace these two shifts.





- Average value A_{E} of the penalty value for N_g generations after mutation:

$$N_g = g - g_{prim}, \qquad (2)$$

$$A_E(g) = \frac{1}{N_g} \sum_{i=0}^{N_g - 1} E(g - i). \qquad (3)$$

- Optimization Speed V :

$$V_E(g) = A_E(g-1) - A_E(g) \qquad (4)$$

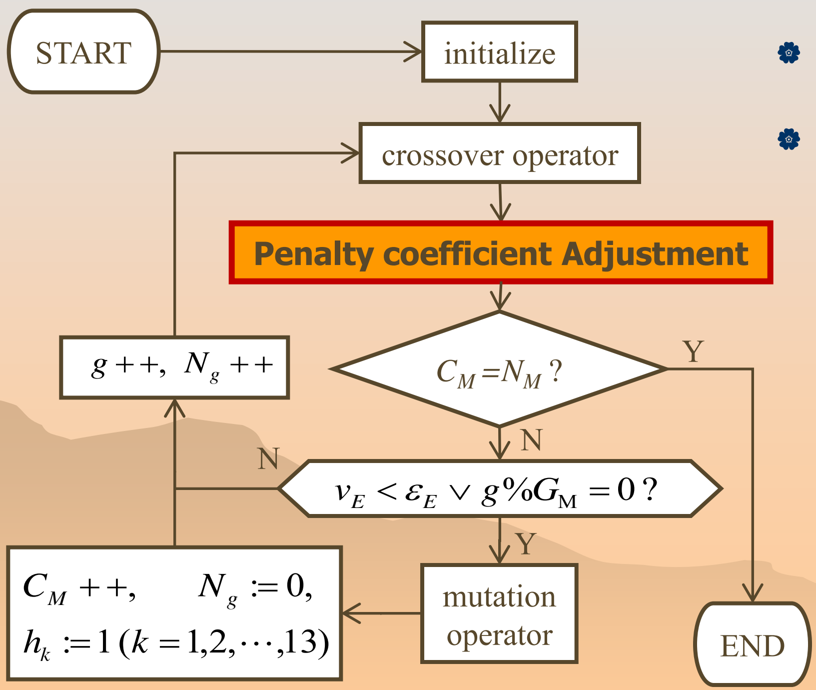
- The mutation is executed when the optimization speed becomes less than a **speedo-threshold ε_{E}**.

$$V(\mathrm{g}) < \varepsilon_{\mathrm{E}}.$$

- Optimization may not advance for several generations right after the mutation.
- Then, the mutation is prohibited for **guard interval G_G** generations right after the mutation.
- When the mutation is executed N_M times, the optimization finishes.

4. Penalty Coefficient Adjustment

- Re-optimization of the schedule is very hard task even by the parallel computing and then requires very long computing time.
- We consider that this problem is caused by the complexity of the solution space. (There are many local minima.)
- When the optimization is caught in a local minimum, some penalties stagnate decreasing as still greater value.
- If the shape of the **solution space** can be deformed, the searching point can **escape** from the local minimum. The shape of the solution space is defined by E. By **changing the penalty coefficients**, the shape of the solution space is deformed.

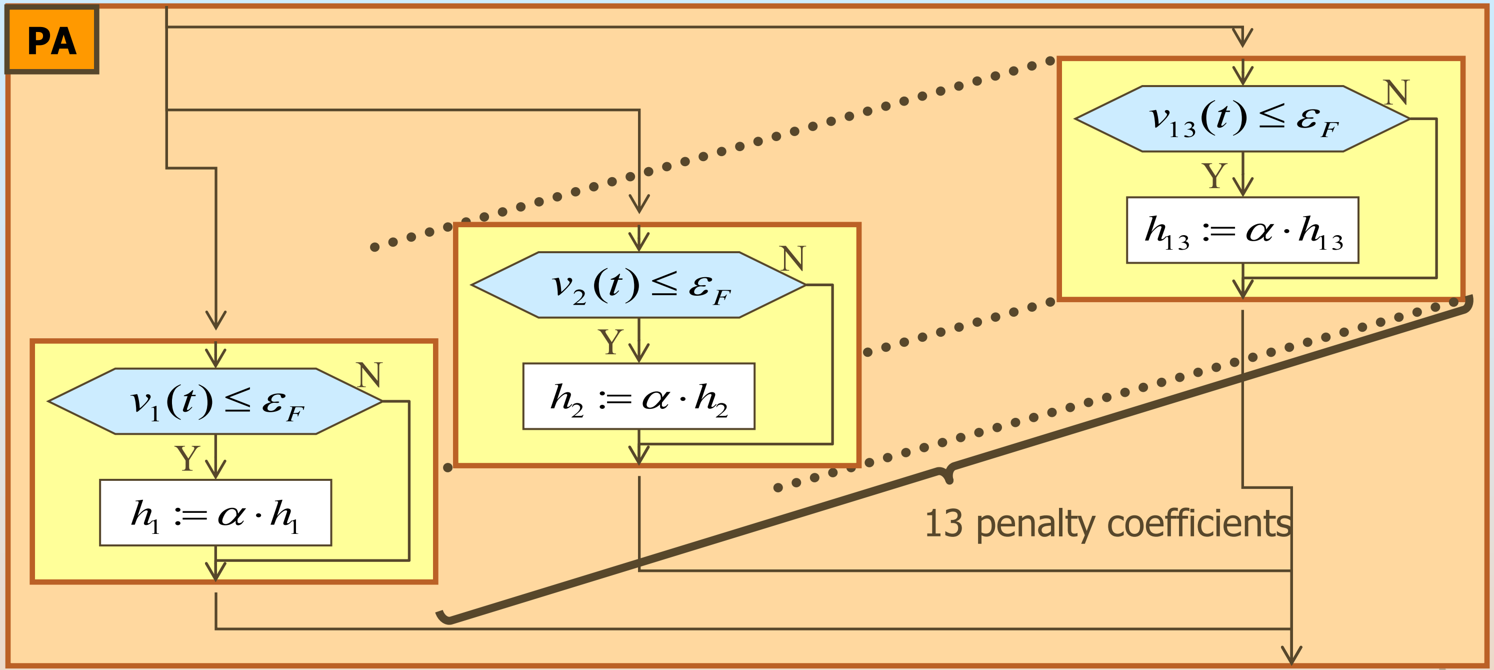


- Initially, all penalty coefficients $h_1 \cdots h_{13}$ are initialized to 1.
- **Decreasing speed**, v_k, of the k-th penalty is defined as

$$A_k(g) = \begin{cases} \dfrac{1}{N_g} \displaystyle\sum_{h=0}^{N_g-1} \sum_{i=1}^{M} F_{ki}(g-h) & (k \le 6) \\ \dfrac{1}{N_g} \displaystyle\sum_{h=0}^{N_g-1} \sum_{j=1}^{D} F_{kj}(g-h) & (k > 6) \end{cases} \qquad (2)$$

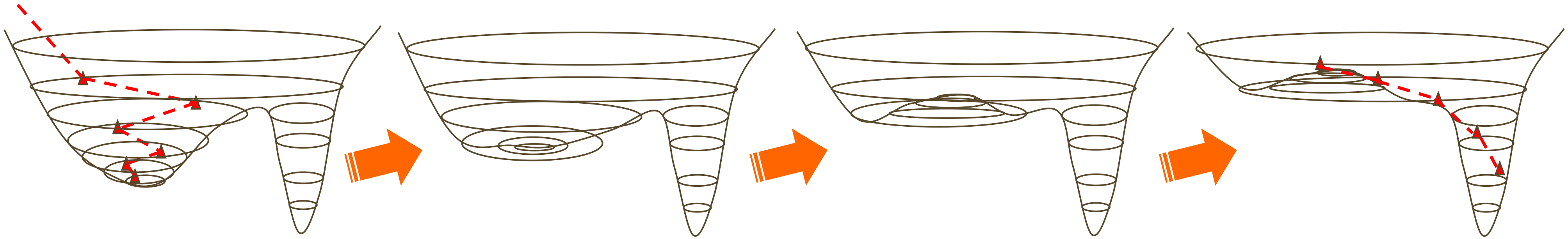
$$v_k = A_k(g-1) - A_k(g). \qquad (3)$$

- When the **decreasing speed**, v_k, becomes less than or equal to a **speed threshold** ε_F, the penalty weight h_k increased by multiplying with α . (ε_F=0.01, α =1.01)
- When the mutation is activated, all penalty weights are initialized to 1.



★Effect of Penalty coefficient Adjustment

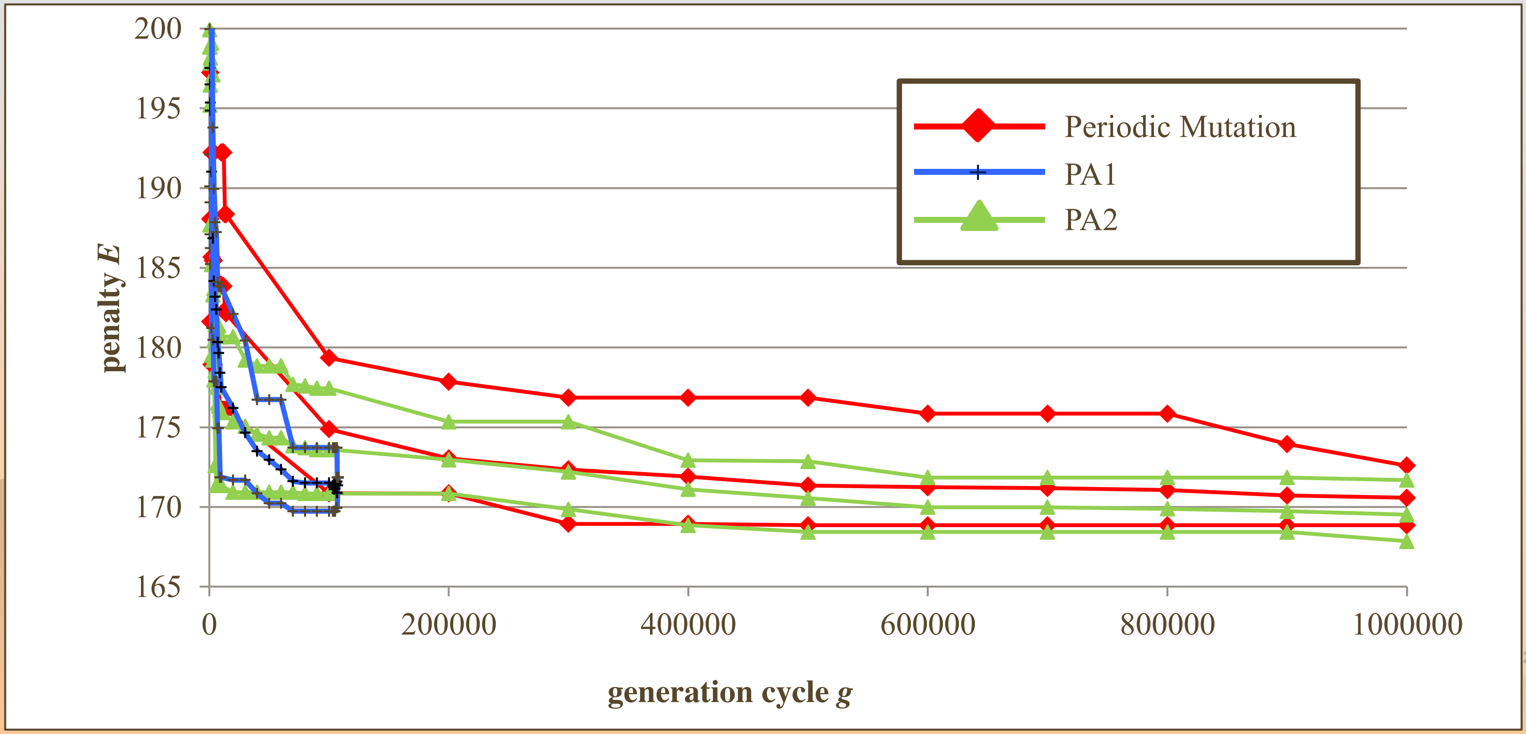
Optimization stagnates… ⇒ caught in a wide local minimum region



5. Practical Experiment of NS

- The number of nurses : 23
- We suppose that two weeks have passed. Two changes (one emergency attendance, one unplanned absence) in the past two weeks.

- 1,000,000 generations are performed.
- Ten time of the optimization are executed.
- The guard interval G_G : 50.
- The speed threshold ε : 0.01.
- The mutation number N_M : 500. (Periodic Mutation & PA1)



- PA2 denotes results by applying the PA for 1,000,000 generation cycles.

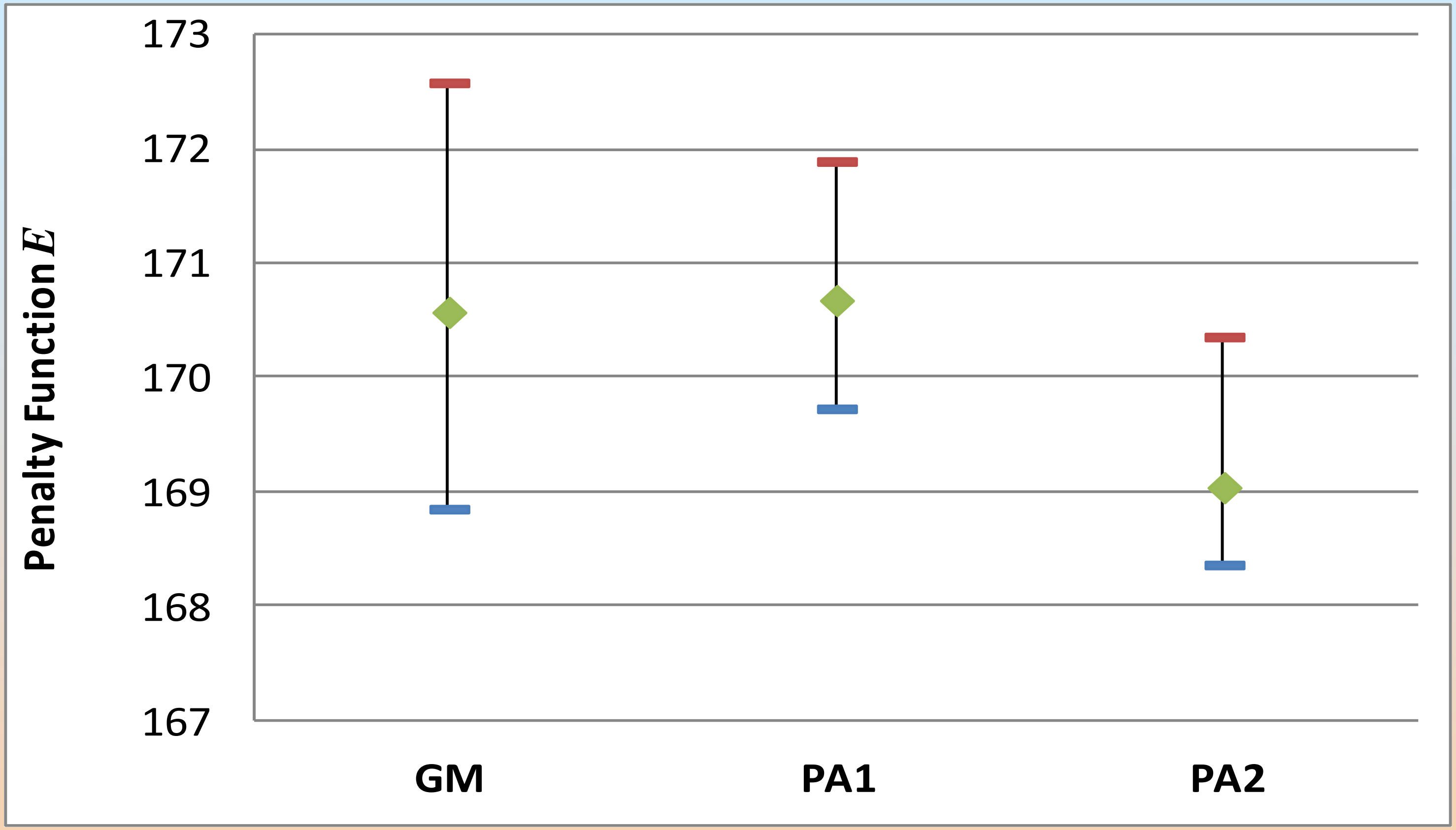


Fig.7: Comparison of the optimization result between the **periodic mutation (GM)** and the mutation depending on the optimization speed with the **Penalty coefficient Adjustment**.

5. Practical Experiment of NS

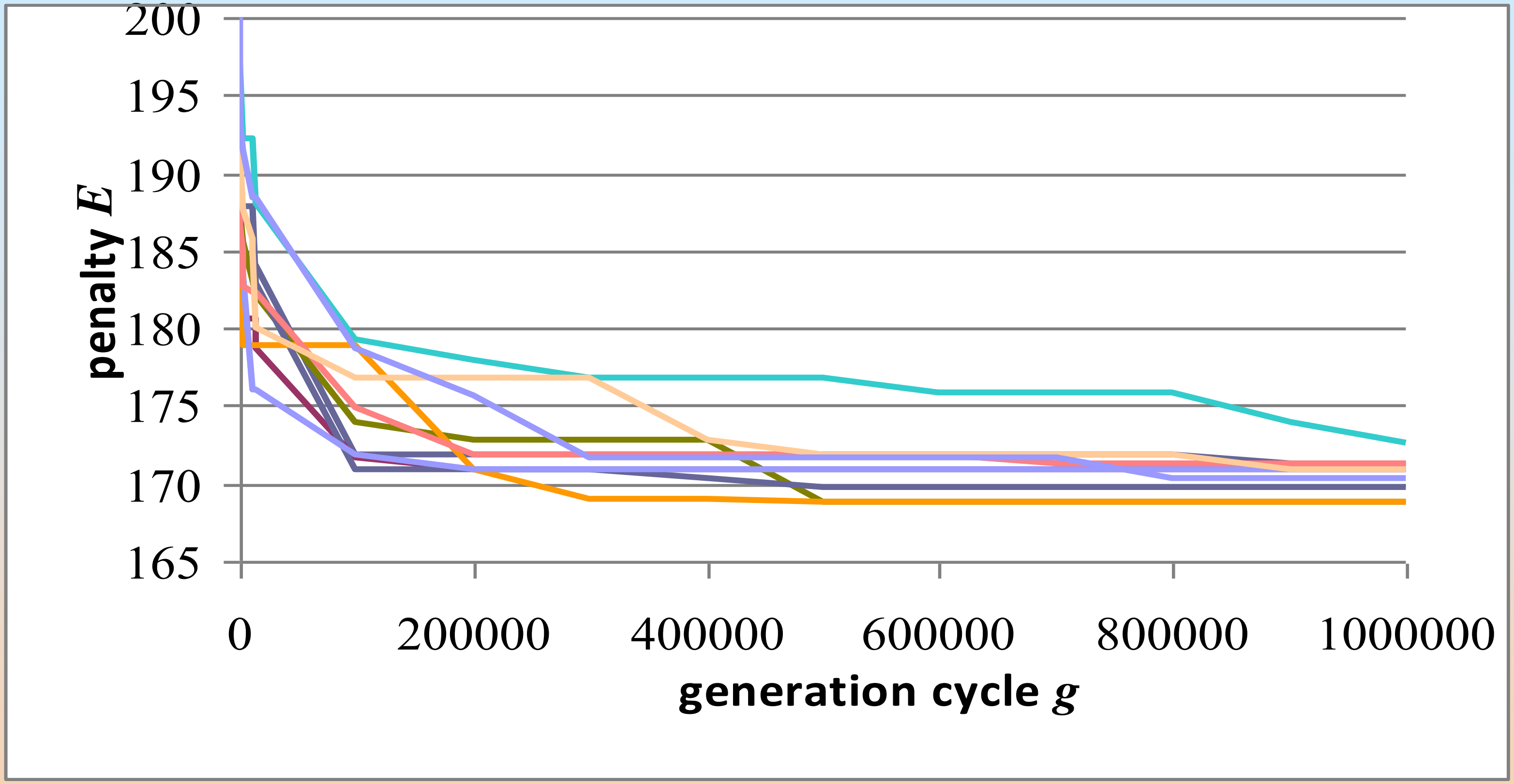


Fig.A: Ten optimization progresses when **the** **periodic mutation** is applied to CGA. (**GM**)

This result is almost same as the result given by CGA with the mutation depending on the optimization speed.

5. Practical Experiment of NS

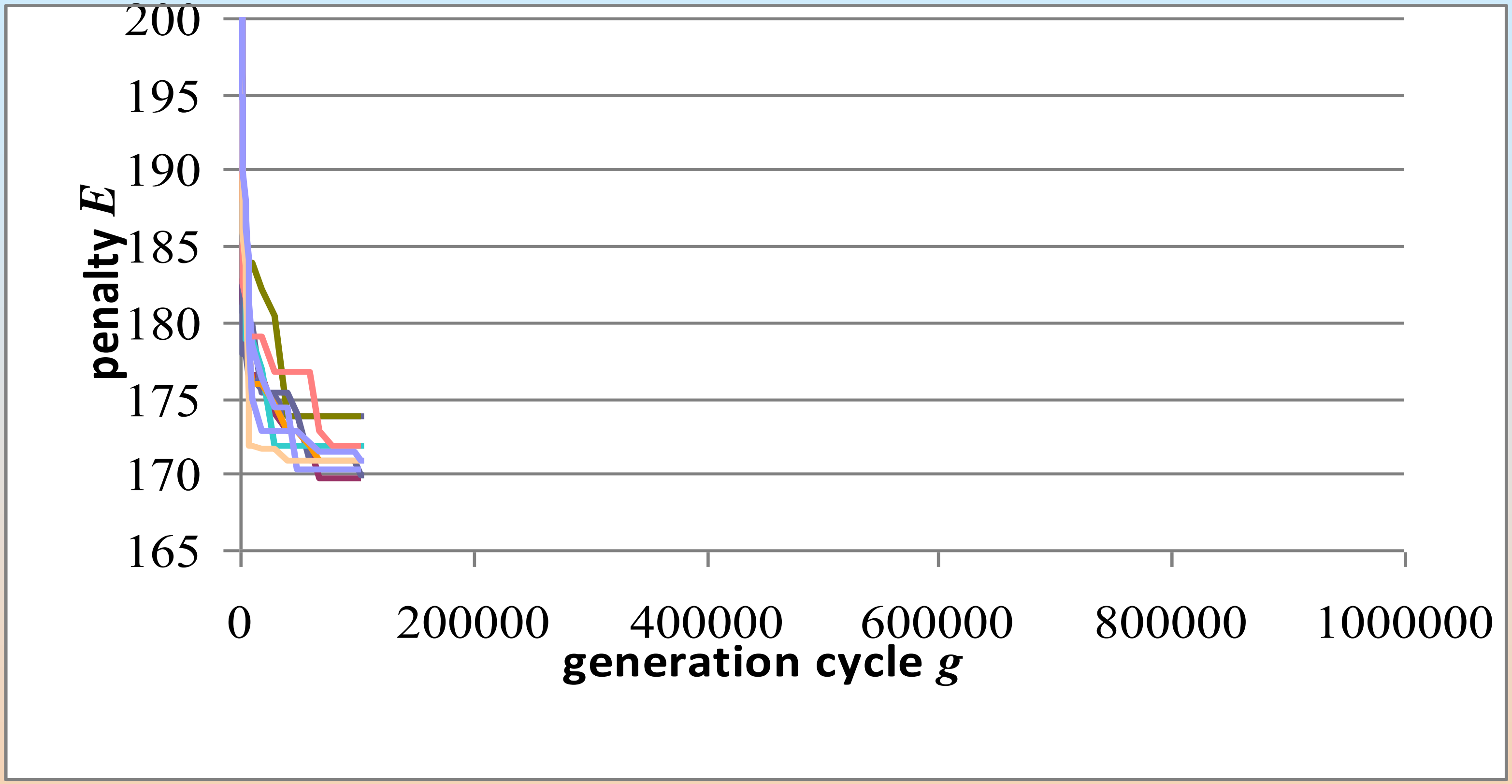


Fig.B: Ten optimization progresses when the mutation depending on the optimization speed with the **Penalty coefficient Adjustment** is applied to CGA. (**PA1**)

The mutation period has been shorten by PA.
Then the optimization has been accelerated.

5. Practical Experiment of NS

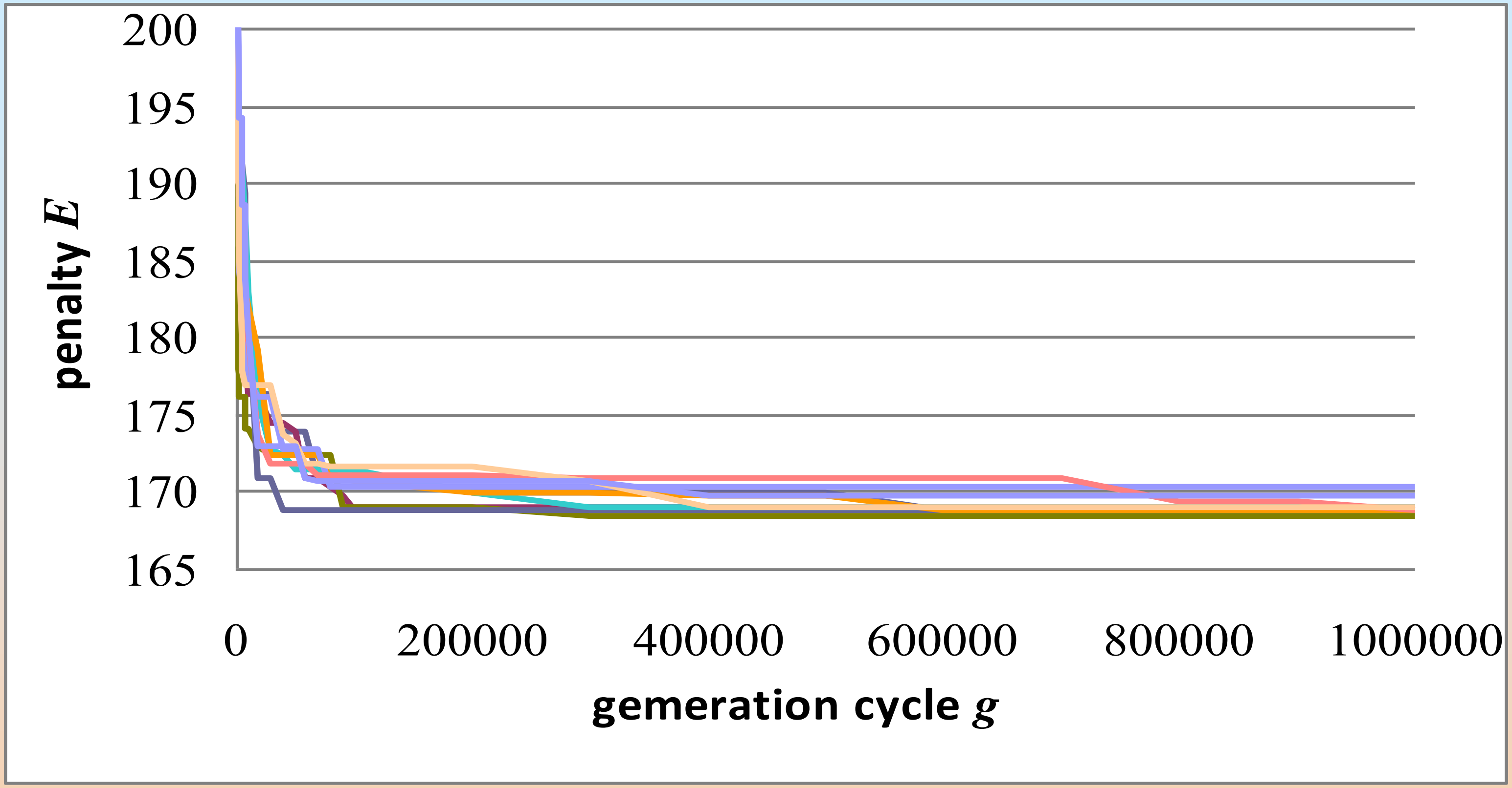


Fig.C: Ten optimization progresses when the mutation depending on the optimization speed with the **Penalty coefficient Adjustment** is applied to CGA. The optimization is executed for 1,000,000 generation cycles. (**PA2**)