CHAPTER 5

HEURISTIC GENETIC ALGORITHM FOR WORKFORCE SCHEDULING

This chapter discusses a heuristic Genetic Algorithm approach to determine work assignments with minimum total worker-location changeover. Two mathematical models (model A1 and A2) for the administrative control of noise hazard prevention problem were discussed in Chapter 3. Since the optimization approach has a limitation in optimally solving large-sized problem, the heuristic GA is then developed to find the optimal (or near-optimal) noise control solution for this problem. Heuristic crossover and mutation operation are used to search the work assignment solution having minimum total worker-location changeover. The final section presents the effectiveness of GA by comparing GA solutions to those from the optimization approach.

5.1 Problem Description

Frequently occurring injuries and health problems in the workplace are caused by excessive exposure to occupational hazards. For example, low back injury is caused by overexertion and hearing loss is caused by excessive exposure to loud noise. In most industrial facilities, the presence of occupational hazards is inevitable. To protect workers from such hazards, both the allowable exposure duration and permissible exposure level are usually established. It is also common to set the permissible level as a quantity that must not be exceeded within an 8-hour workday. For example, OSHA (1983) imposes an 8-hour time-weighted average sound level (8-hour TWA) of 90 dB*A* as a permissible daily noise exposure. NIOSH (1997) recommends a daily energy expenditure limit to be 33 percent of maximum oxygen uptake of an individual worker. Permissible levels for other occupational hazards such as thermal, toxic chemical substances, and radiation can be found in the literature.

Here, we emphasize the industrial noise hazard since it exists in most industrial facilities. Moreover, noise-induced hearing loss is one of the most common occupational diseases and the second most self-reported occupational illness or injury. Basically, workers are assigned to do various jobs and also rotate their jobs in different periods during one workday. In this way, the effect from hazardous jobs can be split and shared by many workers, instead of concentrating on some particular workers. Job rotation offers a trade-off between safety and productivity (Olishifski and Standard, 1988). However, detailed discussion on job rotation is relatively scarce.

Job rotation is usually (and mistakenly) judged to be simple and easy to implement. In practice, work assignments that specify work areas where individual workers are to be assigned to and work duration at each worker location must be defined. To search for the safety work assignments for workers (such that their daily noise exposures do not exceed the permissible level) is not an easy task. Generally, a job rotation problem can be categorized as a *balanced* work assignment or an *unbalanced* work assignment problem, depending on the numbers of workers and of worker locations. For the *balanced* work assignment problem, Nanthavanij and Kullpattaranirun (2001) introduced a genetic algorithm to determine near-optimal *minimax* work assignments. A heuristic genetic algorithm for the *minimax* work assignment problem that improves the computation time and quality of solution was later developed by Kullpattaranirun and Nanthavanij (2005). Readers should note that those two GAs are unconstrained GAs; thus, the resulting *minimax* noise exposure may exceed the permissible level.

From an engineering viewpoint, not only safety but also productivity of workers needs to be taken into account when job rotation is implemented. The work assignments that have many worker-location changeovers may affect work productivity. Therefore, to achieve productive work assignments, a total worker-location changeover must be minimized. Chapter 3 presents two mathematical models (i.e., model A1 and A2) representing the job rotation for noise hazard prevention. When the size of the problem increases, the optimization approach has difficulty in optimally solving the problem within given computation time; therefore, a new *constrained* GA for the workforce scheduling problem is proposed. The algorithm employs a hybrid procedure developed by Yaoyuenyong and Nanthavanij (2004) to initially determine a lower bound of the number of workers and to generate an initial population. The GA then uses heuristic crossover and mutation operations to search for the work assignment solution with the minimum total worker-location changeover. The swap and multi-start techniques are also used to improve the GA solution. It is important to note that the heuristic GA yields the *safety* work assignment solution since all daily noise exposures do not exceed the permissible level.

5.2 GA Procedure

The GA for *safety* workforce scheduling with the minimum total worker-location changeover requires conventional parameters, namely, population size *Popsize*, crossover probability *Pc*, mutation probability *Pm*, and maximum generation *Max_gen* (or termination time). Briefly, at an initial iteration, set generation as *gen* = 0. Next, initial chromosome v_k's (k = 1, 2, …, *Popsize*) are created. The GA operations including crossover, mutation, and selection perform the evolutionary process. Before selection, the fitness value of each chromosome is computed from the evaluation function. The best chromosome is registered after the selection process. Then, update the *gen* value (*gen* = *gen* +1). Repeat the GA procedure until *gen* = *Max_gen* or the computation time reaches the termination time.

The solution procedure can be divided into two phases.

Phase 1: Generating initial population

The hybrid procedure developed by Yaoyuenyong and Nanthavanij (2004) is adopted to determine the minimum number of workers (m*) for safety work assignments. The work assignments obtained from this phase will serve as the initial population for the next phase.

Phase 2: Finding safety work assignment solution with minimum total worker-location changeover

With an optimal workforce m* and an initial set of work assignments, the GA is applied to improve the work assignment solution to obtain the solution with the minimum total worker-location changeover and all daily noise exposures of workers not exceeding 90 dB*A*.

The above two phases will yield the work assignment solution that minimizes both the number of workers required for job rotation and the total worker-location changeover. Further, all workers' daily noise exposures will not exceed 90 dB*A*.

5.3 GA Operations

5.3.1 Chromosome Coding and Initial Population

To encode work assignment solutions as chromosomes, one needs to understand the structure of work assignments and how decision variables can be encoded into strings. Firstly, let us consider a simple case of daily work assignments in which each worker is rotated among worker locations throughout the entire day. Table 5.1 gives a possible set of work assignments for five workers (Wl, W2, W3, W4, and W5) and four worker locations (WLl, WL2, WL3, and WL4) when there are four work periods (P1, P2, P3 and P4) per workday. It is noted that a permutation representation scheme is suitable for this type of problem. From Table 5.1, it is seen that workers W2, W3, and W4 must work in all four work periods, while worker W1 works only in the first two work periods (at worker location WL1) and worker W5 also works only in the last two work periods (at worker location WL1).

Table 5.1 Example of a *unbalanced* work assignment problem (m = 5, n = 4)

Worker	Work Period			
	P1	P2	P3	P4
W1	WL1	WL1	-	-
W2	WL2	WL3	WL2	WL2
W3	WL3	WL2	WL3	WL3
W4	WL4	WL4	WL4	WL4
W5	-	-	WL1	WL1

Since the above example is an *unbalanced* work assignment problem (m = 5, n = 4), we can simply add worker location 5 (*WL5*) as a dummy location to convert it to a *balanced* problem (see Table 5.2). When any worker is assigned to *WL5*, he/she will be idle in that work period. (In practice, the worker may be assigned to a low-noise area.)

Table 5.2 A *balanced* work assignment problem with a dummy worker location *WL5* (m = 5, n = 5)

Worker	Work Period			
	P1	P2	P3	P4
W1	WL1	WL1	*WL5*	*WL5*
W2	WL2	WL3	WL2	WL2
W3	WL3	WL2	WL3	WL3
W4	WL4	WL4	WL4	WL4
W5	*WL5*	*WL5*	WL1	WL1

Fig. 5.1 shows a chromosome representation of the work assignment problem as a string. The chromosome string is divided into p segments, where each segment represents a work period. In each segment, there are n genes, where each gene represents a worker location. For our example, the chromosome consists of four segments, with five genes in

each segment. The first five genes show the work assignments for the five workers in work period P1, the next five genes for the work assignments in work period P2, and so on. It should be noted that in each period, the order of assignment is WLl, WL2, WL3, WL4, and *WL5*. This chromosome representation allows each worker to attend only one worker location in one work period, and each worker location has only one worker to attend in one work period as well.

It is also observed that the length of the chromosome string is equal to $m \times p$.

Period 1					Period 2					Period 3					Period 4				
W1	W2	W3	W4	W5	W1	W3	W2	W4	W5	W5	W2	W3	W4	W1	W5	W2	W3	W4	W1

Fig. 5.1 Chromosome encoding

There are a constant number of chromosomes in the population as denoted by *Popsize*. The initial population is obtained from the hybrid procedure.

5.3.2 Crossover

For chromosomes that are coded using permutation representation, there are several applicable crossover operators, namely, partially matched crossover (PMX), cycle crossover, order crossover, position-based crossover, heuristic crossover, and so on (Gen, 1997). The number of chromosomes involved in the crossover operation (or crossover rate) must also be determined. Usually, a crossover rate is defined as a percentage of the total number of chromosomes. Note that the number of chromosomes in the population remains unchanged.

A heuristic crossover is developed from the concept of the classical partially matched crossover (PMX). It consists of two stages: *Crossover* and *Improvement*. In the *Crossover* stage, the procedure is similar to that of the PMX crossover. A work period called *selected work period* is randomly selected. After exchanging chromosome segments between the parents to generate a pair of offspring, the segment on the right side of the cut position is emptied for both offspring. In other words, all workers are unassigned. Then, randomly select a work period called *compared work period* that is next to the selected work period. In the *Improvement* stage, workers are then reassigned to worker locations in order to minimize the total worker-location changeover. The improvement attempt starts with an objective to assign the worker to the worker location where he/she currently works in the *compared work period*. To obtain a feasible work assignment solution, the PMX concept is used to map some genes.

The heuristic crossover algorithm can be described as follows.

1. Randomly select pairs of parent chromosomes from the population. The number of pairs is determined from the given crossover rate.
2. For each pair of chromosomes, randomly select a *selected work period*. The selected period is the same work period for both parent chromosomes.
3. For each pair of chromosomes, randomly select a cut position. This cut position is also the same position for both parent chromosomes.
4. After cutting the chromosomes, exchange the right-hand sides of both parents to generate new offspring.
5. For each pair of offspring, randomly select a *compared work period* which is next to the *selected work period*.
6. For offspring No. 1, in *selected work period*, empty the genes after the cut position in the corresponding period (i.e., leave the workers unassigned). Then, reassign

workers to the same worker locations where they used to be assigned in the *compared work period*.

7. For unassigned workers in Step 6, assign the worker to the worker location in the previous order before the crossover operation. Repeat this step with the remaining workers and worker locations until the chromosome (of the offspring) is completed. Verify that all offspring are legal. If necessary, legalize illegal offspring according to the mapping relationship (see detail in PMX crossover.)

8. Repeat Steps 6 - 7 for offspring No.2.

The crossover probability used in this study is 0.40. (That is, the crossover rate is 40%.)

5.3.3 Mutation

Mutation is a genetic operation which makes random alterations to various chromosomes. The rate of mutation is defined as a percentage of the total number of genes in the population that are allowed to be changed. Random mutation changes a small number of genes in chromosomes depending on a mutation probability *Pm*.

The heuristic mutation algorithm is adapted from the swap mutation and can be described as follows.

1. Randomly choose a chromosome and a work period (called *selected work period*) in which mutation will occur.
2. Randomly choose a worker location (or a worker) in the *selected work period*.
3. Randomly select a *compared work period* which is next to the *selected work period* and find the worker location where the worker in the *selected work period* (in Step 2) works in the *compared work period*.
4. Within the *selected work period*, swap the two workers working in both worker locations found in Steps 2 and 3.

The mutation rate used in this study is 5%.

5.3.4 Fitness, Penalty, and Evaluation Function Definition

An evaluation function is used to evaluate the quality of chromosomes in each generation. The chromosome receiving a high evaluation value will potentially be selected for inclusion in the next generation. To obtain the evaluation function, a fitness function and a penalty coefficient have to be defined.

- *Fitness Function*

Here, a fitness value of the work assignment model (described by Eqs. and Inequality (3.27) - (3.32)) is defined as the total worker-location changeover *F*. Thus, strong chromosomes are those chromosomes that have low fitness values. A fitness function of chromosome *k*, $f_k(v_k)$, can be written as

$$f_k(v_k) \quad = \quad F \qquad (5.3)$$

- *Penalty Function*

Since this problem has an upper bounded constraint, i.e., each daily noise exposure must not exceed 90 dB*A* (or the sum of noise loads per workday of each worker must not exceed 1), a penalty term is added to the fitness function so that any chromosome that falls in infeasible space will have a lesser chance to be selected for inclusion in the next generation than others. The penalty coefficient of chromosome *k*, p_k, is proportional to the amount of extra daily noise load of all workers and can be determined using the following function:

$$p_k = \begin{cases} 0, & \text{if constraint (3.28) is satisfied} \\ \theta \cdot \sum_{i=1}^{m} V_i \cdot gen & \text{otherwise} \end{cases} \qquad (5.4)$$

$$\text{where} \quad W_i = \sum_{j=1}^{n}\sum_{k=1}^{p} w_j x_{ijk}\,, \qquad i = 1, \ldots, m \qquad (5.5)$$

$$V_i = \begin{cases} 0, & \text{if } W_i - 1 \le 0 \\ W_i - 1, & \text{otherwise} \end{cases} \qquad (5.6)$$

θ = a positive value.

To protect from early rejecting an infeasible chromosome that may give good offspring after the GA operations, the penalty function is proportional to the generation number.

- *Evaluation Function*

A function to evaluate the fitness of chromosomes in the current population and of new offspring is a function of the fitness function and penalty coefficient. An evaluation function value of chromosome k, $eval(v_k)$ can be defined as

$$eval(v_k) = \frac{1}{f_k(v_k) + p_k} \qquad k = 1, 2, \ldots, Popsize \qquad (5.7)$$

5.3.5 Selection Procedure

The selection procedure involves two basic issues, namely, sampling space and sampling mechanism.

- *Sampling Space*

The proposed procedure uses *enlarged sampling space* in the GA operation. This method keeps both parents and offspring in the sampling space called enlarged sampling space. Therefore, the size of the sampling space is equal to *Popsize* + (*cross_pair* × 2) + *mut_no*. For this method, the chances that parents and offspring will be selected for inclusion in the next generation depend on their evaluation function values.

- *Sampling Mechanism*

A sampling mechanism involves how to select chromosomes from the sampling space to be the new population. In this study, *roulette wheel selection* is employed. Roulette wheel selection is an elitist approach in which the best chromosome has the highest probability to be selected for inclusion in the next generation. In the selection process, the best chromosome is firstly selected to the next generation. After the selection procedure, the next generation has the same population size as the current one.

5.3.6 Termination Rule

The GA procedure is terminated when the iteration hits a maximum generation denoted by *Max_gen*. In addition, the stopping criterion may use both the maximum generation and termination time when the problem size is increased.

5.3.7 Local Improvement

A local improvement involves a procedure for improving the best work assignment solution obtained from each generation. In the proposed GA, the local improvement employs two algorithms developed by Yaoyuenyong and Nanthavanij (2004).

- *Swap Algorithm*

The objective of the swap algorithm is to swap or exchange two workers (from different worker locations) in the same work period so that the total worker-location changeover is decreased, while the daily noise load that each swapped worker receives does not exceed 1. For any p periods, there are p sub-algorithms which will be applied consecutively. The swap algorithm is described below.

r-Period Swap for Decreasing Total Worker-Location Changeover (r = 1 to p/2 where p = 2, 4, or 8)

1. Randomly choose a worker location j^* to which worker i^* is currently assigned.
2. Find all C_r^p possible combinations of r periods. Let **S** be a set of s_u such that **S** = {s_u: u = 1,…, C_r^p }, where each s_u represents each combination of r periods.
3. For each combination s_u, consider all periods k_a where $a \in s_u$.
4. Find any worker location j_o (to which worker i_o is assigned) where $j_o \neq j^*$ such that [$f_{j=j^*} + f_{j=j_o}$] can be reduced (after swapping) and $W_{i=i^*}$ and $W_{i=i_o}$ are less than or equal to 1.
5. If there exists such worker location j_o, then swap worker i_o and worker i^* between worker locations j_o and j^* in all period k_a where $a \in s_u$.
6. Repeat Steps 3 – 5 for $\forall s_u \in$ **S**.

- *Multi-start Algorithm*

A multi-start algorithm is employed to repeat the swap algorithm. The current best work assignment solution from the previous step will be *shaken* and will re-enter the swap algorithm. The process of *shaking* is to randomly select *one* pair of workers in the same work period and swap their worker locations. Then, the resulting work assignments will also be shaken and subsequently improved by the swap algorithm again. It is expected that this technique can move the current solution to a better neighborhood.

5.4 Numerical Examples and Results

The following parameters are used in the demonstration of the proposed GA procedure. The population size is set at 50 chromosomes. The maximum generation depends on the size of the problem. The heuristic crossover and heuristic mutation are used, with Pc and Pm being 0.40 and 0.05, respectively. A constant value of the penalty function θ is 10. Additionally, the number of times that the work assignment solution is shaken (and improved) is set to 15 based on our computational experiment.

Three *unbalanced* work assignment problems are examined: (1) "M = 5 and n = 4" problem, (2) "M = 8 and n = 6" problem, and (3) "M = 12 and n = 10" problem. The number of work periods per workday for the three problems is four periods (p = 4). The maximum generations for the three problems are 2,500, 4,000 and 30,000 generations, respectively. The termination time is set at 1,000 seconds for all three problems. Each problem is solved 10 times.

All three problems are solved using the proposed heuristic GA, which is written in Visual Basic. For the first two problems, an optimization software program called LINGO is also utilized to obtain the work assignment solution with the minimum number of workers and the minimum total worker-location changeover. The third problem, however, is too large for LINGO to find the optimal solution.

5.4.1 Problem 1 (*M* = 5 and *n* = 4)

Consider the facility where there are four worker locations (WLl, WL2, WL3, and WL4) and five workers available for job rotation. It is assumed that noise loads per work period measured at the four worker locations are 0.3830, 0.3120, 0.2510, and 0.1850, respectively. The two work assignment models described in sections 3.3.2 are solved to obtain the optimal work assignment solution with *m** and *F** (see Table 5.3). Then, the proposed heuristic GA is applied to solve this problem. Table 5.4 shows the *initial* work assignment solution (from the hybrid procedure) and the *final* work assignment solution (from the heuristic GA).

Table 5.3 "Optimal" daily work assignments for the five workers (Problem 1)

Worker	Work Period				Daily Noise Exposure (dB*A*)
	1	2	3	4	
W1	WL4	WL4	WL2	WL2	89.96
W2	WL2	WL2	WL4	WL4	89.96
W3	WL3	WL3	-	WL1	89.12
W4	WL1	-	WL3	WL3	89.12
W5	-	WL1	WL1	-	88.08

Note: *m** = 5; *F** = 5.

Table 5.4 "GA-based" daily work assignments for the five workers (Problem 1)

(a) The *initial* solution

Worker	Work Period				Daily Noise Exposure (dB*A*)
	1	2	3	4	
W1	WL4	WL4	WL2	-	89.08
W2	-	WL1	WL4	WL2	89.08
W3	WL2	-	WL1	WL3	89.60
W4	WL3	WL3	-	WL1	89.12
W5	WL4	WL2	WL3	WL4	89.50

Note: *m** = 5; *F* = 11.

(b) The *final* solution

Worker	Work Period				Daily Noise Exposure (dB*A*)
	1	2	3	4	
W1	WL4	WL4	WL4	WL2	88.97
W2	WL2	WL2	WL2	-	89.52
W3	-	WL1	WL1	WL4	89.64
W4	WL3	WL3	-	WL1	89.12
W5	WL1	-	WL3	WL3	89.12

Note: *m** = 5; *F** = 5.

5.4.2 Problem 2 (*M* = 8 and *n* = 6)

Next, we consider another facility where there are six worker locations (WLl, WL2, WL3, WL4, WL5, and WL6) and eight workers for job rotation (Wl, W2, W3, W4, W5, W6, W7, and W8). Noise loads per work period at the six worker locations are assumed to be 0.3550, 0.3000, 0.2460, 0.2250, 0.1550, and 0.1200, respectively. Tables 5.5 and 5.6 show the "optimal" work assignment solution (m^* = 6 and F^* = 4) and the "GA-based" work assignment solution (m^* = 6 and F^* = 4) obtained form LINGO and the heuristic GA, respectively.

Table 5.5 "Optimal" daily work assignments for the six workers (Problem 2)

Worker	Work Period				Daily Noise Exposure (dB*A*)
	1	2	3	4	
W1	WL5	WL5	WL2	WL2	89.32
W2	WL2	WL2	WL5	WL5	89.32
W3	WL1	WL1	WL6	WL6	89.63
W4	WL4	WL4	WL4	WL4	89.24
W5	WL6	WL6	WL1	WL1	89.63
W6	WL3	WL3	WL3	WL3	89.88

Note: m^* = 6; F^* = 4.

Table 5.6 "GA-based" daily work assignments for the six workers (Problem 2)

(a) The *initial* solution

Worker	Work Period				Daily Noise Exposure (dB*A*)
	1	2	3	4	
W1	WL1	WL3	WL4	WL6	89.60
W2	WL3	WL1	WL6	WL4	89.60
W3	WL5	WL6	WL1	WL3	89.04
W4	WL6	WL5	WL3	WL1	89.04
W5	WL2	WL4	WL2	WL5	89.85
W6	WL4	WL2	WL5	WL2	89.85

Note: m^* = 6; F = 18.

(b) The *final* solution

Worker	Work Period				Daily Noise Exposure (dB*A*)
	1	2	3	4	
W1	WL4	WL4	WL4	WL4	89.24
W2	WL1	WL1	WL6	WL6	89.63
W3	WL3	WL3	WL3	WL3	89.88
W4	WL6	WL6	WL1	WL1	89.63
W5	WL2	WL2	WL5	WL5	89.32
W6	WL5	WL5	WL2	WL2	89.32

Note: m^* = 6; F^* = 4.

5.4.3 Problem 3 (M = 12 and n = 10)

Problem 3 assumes that there are 12 workers available for job rotation (W1, W2, …, W12). These workers are to be assigned to 10 worker locations (WL1, WL2, …, WL10). Noise loads per work period at the 10 worker locations are assumed to be 0.4002, 0.35717, 0.3333, 0.2711, 0.2506, 0.2222, 0.2003, 0.1999, 0.1555, and 0.1500, respectively. Due to its relatively large size, only the heuristic GA is applied to solve Problem 3. The hybrid procedure yields the initial work assignment solution with m = 11 and F = 33. Then, the heuristic GA is able to reduce the total worker-location changeover to 9 times (F = 9). The resulting work assignment solutions are shown in Table 5.7.

Table 5.7 "GA-based" daily work assignments for the eleven workers (Problem 3)

(a) The *initial* solution

Worker	Work Period				Daily Noise Exposure (dB*A*)
	1	2	3	4	
W1	WL1	-	WL5	WL8	88.83
W2	-	WL1	WL8	WL5	88.83
W3	WL6	WL10	WL1	WL7	89.80
W4	WL9	WL6	WL7	WL1	89.84
W5	WL2	WL8	WL4	-	88.64
W6	WL8	WL2	-	WL4	88.64
W7	WL5	WL7	WL2	WL9	89.73
W8	WL7	WL5	WL9	WL2	89.73
W9	WL3	WL9	WL10	WL3	89.80
W10	WL4	WL3	WL6	WL10	89.83
W11	WL10	WL4	WL3	WL6	89.83

Note: m = 11; F = 33.

(b) The *final* solution

Worker	Work Period				Daily Noise Exposure (dB*A*)
	1	2	3	4	
W1	WL1	WL8	WL8	WL8	90.00
W2	-	WL1	WL1	WL9	89.67
W3	WL8	WL5	WL5	WL5	89.64
W4	WL7	WL7	WL7	WL7	88.40
W5	WL4	WL4	WL4	-	88.51
W6	WL2	WL2	-	WL4	89.89
W7	WL5	-	WL2	WL2	89.74
W8	WL9	WL9	WL9	WL1	88.97
W9	WL10	WL10	WL3	WL3	89.75
W10	WL3	WL3	WL10	WL10	89,75
W11	WL6	WL6	WL6	WL6	89.15

Note: m = 11; F = 9.

5.4.4 Comparisons of Work Assignment Solutions between LINGO and Heuristic GA

To evaluate both the efficiency and effectiveness of the heuristic GA, the following three indices are used: (1) number of workers involved in job rotation, (2) total worker-location changeover, and (3) computation time. In terms of the computation time of the heuristic GA, we consider an average hit time as computed from the 10 replicates. Readers should note that the hit time is the time mark at which the best solution is found by the GA. Table 5.8 shows the comparisons of the three indices between LINGO and the heuristic GA.

Table 5.8 Comparisons between LINGO and Heuristic GA

Problem	Solution Approach	Number of Workers	Total Worker-Location Changeover	Average Hit Time (second)
Problem 1	Optimization (LINGO)	5	5	5,454.00
	Heuristic GA	5	5	0.20
Problem 2	Optimization (LINGO)	6	4	767.00
	Heuristic GA	6	4	0.50
Problem 3	Optimization (LINGO)*	-	-	-
	Heuristic GA	11	9	178.40

*LINGO was terminated after running for 8 hours and not being able to find a feasible solution.

In problems 1 and 2 where LINGO can determine the work assignment solutions with the minimum number of workers, it is seen that the heuristic GA is also able to yield the solutions with the same minimum numbers of workers. Problem 2 results in the balanced work assignments; while, in problem 1 and 3, the heuristic GA yields the unbalanced work assignments. Although there is no minimum solution from LINGO to compare with, it is believed that the heuristic GA is able to determine the minimum number of workers for job rotation. From daily noise exposures in Table 5.7, one can easily see that it is unlikely that the safety work assignments can be obtained when the number of workers is less than eleven.

When evaluating the total worker-location changeover, the heuristic GA is as effective as LINGO in yielding the work assignment solution with the minimum total worker-location changeover. However, the initial solution generated by the hybrid procedure still has many worker-location changeovers. It is the heuristic GA that significantly improves the initial solution such that the final work assignment solution has the minimum total worker-location changeover.

The computation time comparison shows that the heuristic GA is very efficient when comparing with LINGO. In problems 1 and 2, the heuristic GA is able to generate a feasible solution that satisfies all constraints and in relatively short computation time. It is perhaps attributed to an ability of the hybrid procedure in finding the lower bound that is the same or very close to the minimum number of workers for job rotation. This ability helps to shorten the computation time of the heuristic GA since it does not have to do multiple tasks.

Three workplace noise problems are presented as examples to compare the solutions obtained from two solution approaches, i.e., optimization and heuristic GA. Each problem is solved for ten times and using the maximum generation or the termination time as the stopping condition. The results from the examples show that the heuristic GA can find the optimal solution for small-sized ($n \leq 6$) problems. The heuristic GA matches the optimization program (LINGO) with respect to the quality of the solution (as judged from m^* and F^*). In terms of the average hit time, the heuristic GA is significantly superior to LINGO.