Vol. 37 No. 8 Transactions of Information Processing Society of Japan Aug. 1996

Regular Paper

Solution of a Difficult Workforce Scheduling Problem by a Genetic Algorithm

JULIO TANOMARU†

This paper has two major goals: (1) to define a staff scheduling problem for a heterogeneous workforce with many realistic constraints extracted from the real world, and (2) to investigate its solution using a customized genetic algorithm featuring a group of operators which combine stochastic behavior and heuristics. After formulating the problem, schedules for the whole workforce are represented by integer chromosomes of fixed dimension. Violations of constraints and problem requirements are reflected by cost increases, and genetic operators act stochastically but tend to decrease such costs. Although the operators interact with each other, they were designed in an independent way for the sake of simplicity and modularity. Overall, the action of these stochastic-heuristic operators resembles a sophisticated mutation operator biased to improve schedules by reducing the costs of constraint violations. Experiments show that high-quality workforce schedules can be obtained in reasonable time even for large problems.

1. Introduction

Companies or organizations whose workers are allowed to have irregular working schedules have to cope with the problem of assigning working shifts to all the workers in order to attend a predicted work demand over a planning horizon with minimum cost, while considering a number of restrictions concerning labor constraints, individual preferences of each worker, and so on. Hospitals, hotels, telephone companies, departments stores, etc., are examples of situations in which workers with irregular schedules are usually present ☆.

The planning horizon may consist of a few hours, days, or even weeks or longer periods of time over which the demand of work can be predicted with reasonable accuracy. Ideally, the predictions would match the actual demand perfectly, and all the workers would be scheduled in such a way that no worker would either be idle or have to work overtime, and the total production would match the demand as well as possible. However, in real situations particularly found in the service industry, the schedules of all workers are usually determined by one or more human experts, who have learned through experience how to simplify constraints of the problem and generate schedules that, if not optimal, are considered to be acceptable.

In terms of costs, anytime the service ability exceeds the demand, there will be idle or sub-utilized workers, implying low productivity and increased costs. Conversely, if the demand cannot be satisfied, the quality of the service may deteriorate as workers may have to work more than the recommended time. Furthermore, the company would lose potential profits and would risk losing market share to competitors. Therefore, scheduling the workforce in such a way to match the demand without wasting labor and overloading the workers is a crucial and omnipresent problem in the service industry.

The determination of a sub-optimal workforce scheduling for the situations above constitutes a multivariable, multimodal, combinatorial optimization problem involving many constraints. Such characteristics discourage the application of conventional optimization methods, which are likely to be trapped in local optima. On the other hand, recent years have witnessed the emergence of evolutionary algorithms, notably genetic algorithms (GAs)[12),15),21),24)], which have been successfully applied to a number of problems with some of the characteristics mentioned above.

The structure of the remainder of this paper is as follows: Related work concerning employee staffing and scheduling and GAs for scheduling are reviewed in Section 2. A constrained workforce scheduling problem is formally defined in Section 3 for the case when the workers

† Faculty of Engineering, The University of Tokushima

☆ In the remaining of this paper, without loss of generality, the term *company* is used to designate the scheduling context, while the words *worker* and *employee* are considered synonyms.

are classified into several groups or categories. A customized GA for the solution of the proposed problem is introduced in Sections 4 and 5; the former explains the representation approach and the computation of cost parameters associated with the violation of problem constraints, while the latter describes the heuristic genetic operators conceived to decrease the values of the violation costs. Simulation results that confirm the applicability of the proposed method to *real-world* workforce scheduling problems are shown in Section 6, while Section 7 concludes the paper.

2. Related Work

2.1 Employee Staffing and Scheduling

Probably the first formulation of the workforce scheduling problem was accomplished by Dantzig[6], who in 1954 defined a simplified problem for the scheduling of a homogeneous (only one category of workers) workforce during a single day. Much later, in 1982 Mabert and Watts defined the *tour scheduling problem*, in which decisions concerning the scheduling of working shifts and days-off for an extended planning horizon were included[20]. In that problem, the workers are assumed to work in tours, defined as working shifts of fixed length and position in the planning horizon, and the objective is to assign an integer number of employees to each possible tour in such a way that the staffing requirements for all the planning horizon are satisfied at minimum cost.

As computers evolved, researchers began to consider the challenging case in which the workforce is heterogeneous, that is, composed of workers with different qualifications, wages and other features. This is a more realistic situation, found everywhere from production lines to hospitals.

Decision problems concerning the scheduling of a heterogeneous workforce were proved to be NP-complete by Bartholdi[2], and several heuristic approaches have been proposed for this important class of problems[3),9),10),19], all involving simple gradient procedures. Such procedures start from a feasible solution and investigate a few neighbors generated by disturbing a few variables, and replace the solution by a neighbor whenever the latter has better performance than the former.

Even though the heterogeneous tour scheduling problem has been the subject of active research, in practice there are many cases in which actual workforce scheduling problems do not fit its formulation. For example, in many practical situations working shifts are of variable length and may be located with some freedom within the planning period. Furthermore, the conventional tour problem does not consider individual differences among workers, and is only interested in the number of workers of each category assigned to each tour. In the same way, labor considerations such as overtime work, desirable number of working shifts per worker, and special time allowances are totally ignored, although in any practical workforce scheduling problem such factors must always be taken into account.

Such considerations motivated us to propose a new formulation for a heterogeneous workforce scheduling problem with working shifts of variable length and position, workers with individualized time preferences, and a series of labor considerations, described in detail in the following sections of this paper.

2.2 GA for Scheduling Problems

One may consider that the gradient-based, heuristic procedures proposed for the conventional tour scheduling problem could be modified to fit the formulation of a new workforce scheduling problem featuring flexible working shifts and a number of new constraints. However, it is well known that gradient methods only work well when the search space is convex and unimodal, and this is not the case with this class of scheduling problems[2),8)].

Where gradient optimization methods fail, evolutionary methods such as GAs[12),15),21),24)] may succeed. An impressive number of representative combinatorial optimization problems has been solved using GAs since the late 1980s, including the traveling salesman problem[14),21)], the transportation problem[21),25)], clustering[4], and applications to process control[18),22)], among many others.

A glance at the recent GA literature also shows a considerable number of papers concerning the application of GAs to scheduling problems[1),5),7),8),11),16)]. However, although they all use the word *scheduling* as a keyword, most of them focus on the job shop scheduling problem[1),5),7),16)], a problem in which the objective is to schedule the usage of a set of work stations (machines) of a job shop in such a way that the job contracts are fulfilled with maximum profits. This is a problem only slightly related to workforce scheduling, since there is a

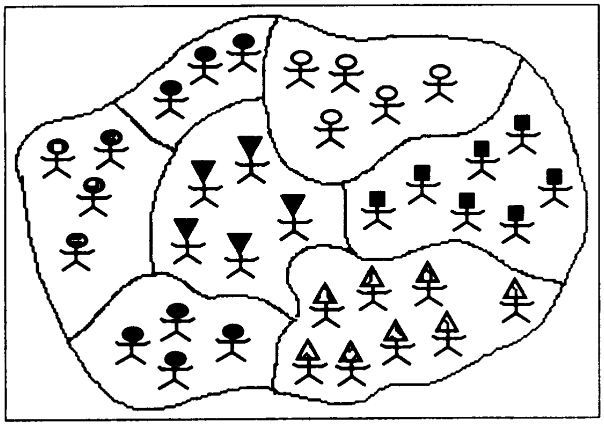

Fig. 1 Pictorial depiction of a heterogeneous workforce of 36 employees divided into 7 groups or categories.

relatively small number of variables to consider. An original problem, the scheduling of university tests by a GA has also been considered[11], but in practice such a problem does not have a large search space, and can be solved well using heuristic rules. Workforce scheduling using a distributed GA was the subject of investigation by Easton and Mansour[8], but they aimed their approach at the tour scheduling problem, which has the limitations explained above.

3. Problem Statement

This section introduces the formulation of a new workforce scheduling problem. Assume that there is an company with a heterogeneous workforce classified into several employee groups or categories as shown in **Fig. 1**.

The proposed constrained heterogeneous workforce scheduling problem (CHWSP) consists of determining the working schedule of all the workers during a given planning horizon in order to minimize the total costs, while satisfying a number of constraints. Such constraints may be general (e.g., available number of employees), specific to a group (e.g., maximum and average duration of a working shift), or specific to an employee (e.g., time-preferences of a worker or desired number of working shifts).

The planning horizon consists of discrete, consecutive, and uniform (same duration) time intervals t, for $t = 1, 2, 3, \ldots, T$. Each employee works in *shifts*, each shift consisting of an integer number of time intervals. Also, let S denote the maximum number of working shifts of any employee during the planning horizon.

3.1 Workforce Requirements

The basic requirements of the CHWSP include R_t and R_t^g, respectively the total required number of employees and the minimum number of employees of group g required at the interval t, for $t = 1, 2, \ldots, T$. Additionally, there is a parameter γ such that γR_t gives the minimum acceptable total number of employees during a time interval.

For example, consider the time $t = 2$ for a simple CHWSP with only two groups of employees, and let the workforce requirements for that time be given by $R_2 = 10$, $R_2^1 = 5$, $R_2^2 = 2$, and $\gamma = 0.8$. In this case, any schedule with total number of employees at least equal to $0.8 \times 10 = 8$ is acceptable, although 10 is the ideal number. Also, there should be at least 5 workers of the first group, and at least 2 of the second group.

3.2 Employee Groups

In CHWSP, the employees are assumed to belong to different groups or categories, each one labeled by a super-index g, $g = 1, 2, \ldots, G$. Group characteristics include: α^g, a binary flag indicating whether time allowances should be considered (1 = yes, 0 = no); β^g, the overtime allowance coefficient for group g expressed as a fraction of the main wages; $0 < \mu^g \leq 1$, the group's productivity coefficient; σ_m^g, σ_s^g, and σ_M^g, respectively the minimum, standard, and maximum number of consecutive time intervals per working shift for any member of the group; σ_r^g, the minimum number of consecutive time intervals between shifts (rest); ω_m^g and ω_M^g, respectively the minimum and maximum desired number of working shifts during the planning horizon; H^g, the minimum number of days-off during the planning horizon; and N^g, the number of employees available in group g.

3.3 Employee's Specifications

An index e, $e = 1, 2, 3, \ldots, E$ is assigned to each employee. The individual characteristics of each employee include: g_e, the group to which the employee e belongs; W_e, the basic wage of employee e for one time interval; an array $\boldsymbol{P}^e = (\pi_t^e)$, for $t = 1, 2, \ldots, T$, indicating the working time preferences of employee e for all time intervals; each element of the array can assume only three values: 1 (preferable), 0 (acceptable), or −1 (unacceptable).

3.4 Other General Data

A day-off is defined as an integer number H of consecutive time intervals in which an employee is not scheduled to work. Furthermore, for each time t of the planning horizon the company specifies a special time allowance coefficient A_t, which affects only those groups of employees for which $\alpha^g = 1$. These coefficients allow for variation of actual wages according to time, which is a realistic assumption, since many companies reward certain groups of employees for work done on holidays or at night, for example, while some other groups are not eligible for the special allowances.

3.5 Wages Policy

All the wages are assumed to correspond to one time interval and, as stated above, any employee works in shifts made up integer numbers of such intervals. For an employee e, the basic labor costs corresponding to a working shift between time intervals t_1 and t_2 is given by

$$c^e_{t_1,t_2} = \sum_{t=t_1}^{t_2}(1 + \alpha^{e_g} A_t)W_e \qquad (1)$$

where the second term enclosed in the parentheses stand for special time allowances. For example, a given group of employees may be entitled to 30% over the normal wages for work done on Christmas.

3.6 Goals of CHWSP

A CHWSP is specified by the values T, S, G, R_t, R^g_t, γ, α^g, β^g, μ^g, σ^g_m, σ^g_s, σ^g_M, σ^g_r, ω^g_m, ω^g_M, H^g, N^g, E, g_e, W_e, π^e_t, H, A_t, for $t = 1,2,\ldots,T$, $g = 1,2,\ldots,G$, and $e = 1,2,\ldots,E$. For instance, for a problem with $T = 200$, $E = 100$, and $G = 10$, there is nothing less than 22,715 numbers to consider, making it a very difficult problem to be solved by conventional methods.

A schedule for the CHWSP is defined as a sequence of numbers specifying the working times for all the employees of the workforce during the planning horizon. Let $\boldsymbol{s}$ be a given schedule for all employees for the whole planning horizon, and let $C(\boldsymbol{s})$ represent the schedule's associated cost, which includes labor costs and penalties corresponding to violation of the constraints of the problem. The objective is to find a schedule $\boldsymbol{s}$ such that $C(\boldsymbol{s})$ is as small as possible. As stated, CHWSP is a multimodal optimization problem with a search space which increases with the number of employees, the length of the planning horizon, and so on. Such characteristics lead one to speculate that a GA conveniently customized to CHWSP should outperform conventional optimization methods based on simple local gradient procedures.

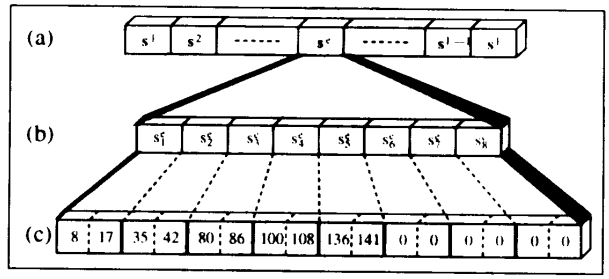

Fig. 2 Example of a chromosome (candidate solution) for a CHWSP with E employees and maximum number of shifts $S = 8$. (a) The whole chromosome; (b) Schedule of the e-th employee; (c) Further zooming-in shows that the employee has 5 active and 3 inactive working shifts.

4. Customized GA to CHWSP

4.1 Chromosome Representation

The schedule of each employee is represented as a list of working shifts, where each shift is denoted by a pair of integers corresponding to the first and last time intervals of the shift. A candidate solution (chromosome) is a set of working schedules for each employee, and populations of such candidate solutions evolve as in conventional GAs. For an employee e, the i-th working shift can be represented as

$$s^e_i = (s^e_{i1}, s^e_{i2}) \qquad (2)$$

where $0 \le s^e_{i1} \le s^e_{i2} \le T$ denote, respectively, the first and the last time intervals of the shift. Inactive shifts are represented by $(0, 0)$. Case when $s^e_{i1} = s^e_{i2}$ correspond to working shifts consisting of a single time interval. The total schedule for employee e is just the concatenated list of the corresponding working shifts. Since it is assumed that any employee can have a maximum of S working shifts during the planning horizon, a simple approach is to represent the whole schedule for an employee as an ordered array

$$\boldsymbol{s}_e = [\ s^e_1 \quad s^e_2 \quad s^e_3 \quad \cdots \quad s^e_{S-1} \quad s^e_S\ ], \qquad (3)$$

so that the schedule of all workers becomes

$$\boldsymbol{s} = [\ \boldsymbol{s}^1 \quad \boldsymbol{s}^2 \quad \boldsymbol{s}^3 \quad \ldots \quad \boldsymbol{s}^E\ ], \qquad (4)$$

which denotes a complete candidate solution for the CHWSP and consists of $2 \times S \times E$ integers between 0 and T.

An example of a chromosome (candidate solution) is shown in **Fig. 2** for the case in which $S = 8$.

4.2 Basic Costs

The wage costs of an employee during all the planning horizon is given by

$$c^e(\boldsymbol{s}) = \sum_{i=1}^{n_s^e} c_i^e(\boldsymbol{s}) \quad (5)$$

where $c_i^e(\boldsymbol{s})$ is the cost of the i-th shift and n_s^e is the number of active working shifts of employee e. For convenience, the n_s^e active shifts are always translated to the first positions, while the remaining $S - n_s^e$ (idle) shifts occupy the last positions of $\boldsymbol{s}^e$, as indicated in Fig. 2. For each candidate schedule $\boldsymbol{s}$, the following values can be easily computed: n_h^e, the number of days off; n_t^g, the number of employees of group g scheduled to work at time interval t; n_t, the total number of employees at time t; n_t^*, the effective staff scheduled to time t; and $c(\boldsymbol{s})$, the total wage cost of the schedule, where

$$n_t^* = \sum_{g=1}^{G} \mu^g n_t^g \quad (6)$$

and

$$c(\boldsymbol{s}) = \sum_{e=1}^{E} c^e(\boldsymbol{s}). \quad (7)$$

4.3 Costs of Constraint Violations

A given candidate solution may violate certain constraints by, for example, scheduling some employees to work overtime. Such violations are mandatory in many cases, depending on staff requirements and employees' data. In this paper, violations are penalized by increasing the scheduling cost according to the degree of the violation. Sometimes penalizing a candidate solution due to the violation of soft constraints directly reflects the increase in labor costs as, for example, in the overtime work case. However, there are penalties that not necessarily mean that any real cost will be incurred, and such penalties are artificially set. The current system distinguishes a total of 9 types of constraint violations, and associate penalties to all of them. Some of the violations are applied to each active working shift of an employee, while others are computed only with respect to the whole schedule of an employee, and others only make sense for the whole workforce schedule.

In order to facilitate the definition of the costs due to violation of constraints, a positive-identity function $I^+(\cdot)$ is introduced as

$$I^+(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

4.4 Individual Working Shifts

a) Employee group requirement σ_m^g not fulfilled for working shift s_i^e: In this case the cost of the working shift is increased by 50% of the difference between the minimum specified number of time intervals for a working shift of the employee, and the number of time intervals actually worked, times the basic employee's wages. The penalty corresponding to the i-th working shift of employee e is given by

$$v_1(e_i) = I^+[0.5(\sigma_m^{e_g} - (s_{i,2}^e - s_{i,1}^e + 1))W_e] \quad (9)$$

where $e = 1, 2, \ldots, E$ and $i = 1, 2, \ldots, n_s^e$.

b) Employee group requirement σ_s^e not fulfilled (overtime work): The corresponding penalty for the i-th shift of employee e is given by

$$v_2(e_i) = I^+[(s_{i,2}^e - s_{i,1}^e + 1 - \sigma_s^{e_g})W_e], \quad (10)$$

meaning that any working shift exceeding the standard shift duration specified for the corresponding employee group is considered as overtime work.

c) Employee group requirement s_r^g not fulfilled: This implies a cost increase proportional to the difference between the minimum specified number of idle time intervals between consecutive shifts and the actual number of resting intervals. The penalty is defined as

$$v_3(e_i) = I^+[0.5(\sigma_r^{e_g} - (s_{i+1,1}^e - s_{i,2}^e - 1))W_e] \quad (11)$$

for $e = 1, 2, \ldots, E$ and $i = 1, 2, \ldots, n_s^e - 1$.

d) Employee's working time preferences not fulfilled: Recalling that the working time preferences of an employee are expressed as a tri-state array in which each element can assume the values -1, 0, or +1, a simple way to penalize schedules which violate an employee's time preferences table is given by the non-negative quantity defined below.

$$v_4(e_i) = 0.5 \sum_{t=s_{i,1}^e}^{s_{i,2}^e} (1 - \pi_t^e)W_e. \quad (12)$$

The penalty values defined in Eqs. (9)-(12) can be summed up for all the active working shifts of a given employee e, resulting in the non-negative values $v_1(e)$, $v_2(e)$, $v_3(e)$, and $v_4(e)$, respectively.

4.5 Total Schedule of an Employee

a) Employee group requirement h^g not fulfilled: This situation is severely penalized by adding to the employee's cost the wages corresponding to a minimum working shift.

$$v_5(e) = I^+[(H^{e_g} - n_h^e)\sigma_s^{e_g} W_e]. \quad (13)$$

b) Employee group's required number of shifts not fulfilled: An effort is done in order to try to schedule each employee to a number of working shifts in the specified range of the employee's group. The following penalties have been adopted:

$$v_6(e) = I^+[0.5(\omega_m^{e_g} - n_s^e)\sigma_m^{e_g} W_e] \quad (14)$$

$$v_7(e) = I^+[0.5(n_s^e - \omega_m^{e_g})\sigma_s^{e_g} W_e]. \quad (15)$$

4.6 Total Workforce Requirements

Let ν^1 and ν^2 be penalty coefficients specified in cost units. Then the following violation costs are defined:

a) Minimum workforce requirement R_t not fulfilled:

$$v_8(t) = \begin{cases} 2(R_t - n_t^*)\nu^1 & \text{if } n_t^* < \gamma R_t \\ I^+[(R_t - n_t^*)\nu^1] & \text{otherwise} \end{cases} . \quad (16)$$

b) Workforce requirement R_t^g not fulfilled:

$$v_9(t) = I^+[(R_t^g - n_t^g)\nu^2]. \quad (17)$$

The equations above define simple penalties proportional to the deficit of workers for each time interval. Penalties are doubled when the number of effective workers is below the minimum acceptable number of workers γR_t. Conversely, when the number of scheduled workers exceeds the requirements, it is expected that the corresponding costs will eventually result in the *death* of the schedule due to selective pressure of the GA mechanisms.

4.7 Total Cost

The total cost of a candidate solution is given by the sum of its basic labor costs and the penalties corresponding to violations of soft constraints and workforce requirements, resulting in

$$C(\boldsymbol{s}) = c(\boldsymbol{s}) + v_1(\boldsymbol{s}) + \beta^{e_g} v_2(\boldsymbol{s}) + \sum_{i=3}^{9} v_i(\boldsymbol{s}). \quad (18)$$

All the terms in the equation above are nonnegative, so that simple summation suffices to specify the cost of a schedule. The third-term on the right-hand side of the equation is weighted by the overtime allowance coefficient β^{e_g}.

5. Genetic Operators

5.1 Reproduction and Crossover

As in conventional GAs, starting from a population of candidate schedules, the first step to produce a new generation is reproduction based on the total cost values, after fitness scaling is carried out to help prevent premature convergence. An elitist approach[12] is taken, since the main target is not the population average, but the population's best. Following reproduction, two-point crossover is performed on a "per-employee" basis, in such a way to allow whole schedules for pairs of employees to be exchanged.

5.2 Heuristic Operators

After reproduction and crossover, stochastic-heuristic operators are applied to decrease the penalty values. Such operators are divided into three groups: those concerning only isolated working shifts, operators Op_1–Op_4; those concerning the whole schedule of a single employee, operators Op_5–Op_7; and those concerning a whole candidate solution, operators Op_8–Op_9. For each violation v_i, an operator Op_i has been devised by combining heuristics and probability. Although the constraints are mutually related, the operators were devised in an independent way for the sake of simplicity and modularity, making easy the addition of new operators.

To conceive the operators, the idea was to start from a deterministic operation and add to it a random component heuristically biased. This approach was first proposed by Davis[7], who argued that a system based on such a procedure should be able to outperform the deterministic predecessor in the same environmental niche.

Furthermore, the overall action of the proposed operators was designed in such a way that all the generated schedules are, in principle, feasible. For example, schedules in which a worker is assigned to overlapping working shifts will never occur. Had a binary representation and conventional 2-point crossover and mutation operators, as in the popular simple GA[12], be chosen, much probably a large proportion of schedules without meaningful interpretation would result, and computationally expensive heuristic correction procedures would be necessary. Exploring unfeasible regions of the search space translates to a waste of computation resources, so the approach of incorporating problem-specific knowledge to the representation and overall mechanism was adopted[13),21].

Note, however, that saying that unfeasible solutions are not considered does not mean that, for example, a worker cannot be scheduled to work 100 hours in a row. Such situations are theoretically possible, but since the associated penalty values will be large, it is expected that the GA selection process will *kill* such a poor-quality schedule in a few generations. The heuristic operators conceived so far are described bellow, together with an operator selection procedure that controls their application.

a) Operator Op_1

(1) Delete shift with probability

$$Prob^1_{\text{del}} = 1 - \frac{s_{i,2} - s_{i,1} - 1}{\sigma_s^{e_g}}, \quad (19)$$

meaning that too short shifts have high probability of being deleted (canceled).

(2) If the shift survives deletion, extend it by one time interval to the left or right in the planning horizon, according to the employee's time preferences of the time interval to be appended, that is, the probability of stretching the shift to the left is given by

$Prob^1_{\text{left}} \Rightarrow$

	$\pi^e_{s^e_{i,2}+1}$: −1	0	1
$\pi^e_{s^e_{i,1}-1}$: −1	0.50	0.25	0.10
0	0.75	0.50	0.25
1	0.90	0.75	0.50

while the probability of extending the working shift to the right is given by

$$Prob^1_{\text{right}} = 1 - Prob^1_{\text{left}}. \quad (20)$$

Putting into words, when the time preferences are the same for both the neighboring intervals of the working shift to be stretched, there is a 50% probability of extending the shift in each direction. When the worker preferences favor one of the time intervals slightly (1 vs 0 or 0 vs −1), the preferred interval has a 75% probability of being included in the extended shift, while the other interval is given a 25% probability. When the worker prefers one of the intervals strongly (1 vs −1), the probabilities are set to 90% and 10%, respectively.

b) Operator Op_2

(1) Prune the shift by one time interval from the left or right, according to the employee's time preferences of the time interval to be deleted, that is, the probability of shortening the shift from the left is given by

$Prob^2_{\text{left}} \Rightarrow$

	$\pi^e_{s^e_{i,2}}$: −1	0	1
$\pi^e_{s^e_{i,1}}$: −1	0.50	0.75	0.90
0	0.25	0.50	0.75
1	0.10	0.25	0.50

and the probability of shortening the shift from the right (finishing the shift earlier) is just the complementary value.

The reasoning here is analogous to in the case of operator Op_1: When a working shift is to be shortened, the time preferences of the first and last intervals of the shift are compared and given deletion probabilities in such a way to favor the worker's preferences, but also allow mutation in the opposite direction.

c) Operator Op_3

(1) Move shifts apart by one time interval by moving either the left-hand working shift or only the right-hand one according to the employee's time preferences. The probability of moving the left-hand shift is given by

$$Prob^3_{\text{left}} = \frac{(\pi^e_{s^e_{i,1}-1} - \pi^e s^e_{i,2})}{10} - \frac{(\pi^e_{s^e_{i+1,2}+1} - \pi^e_{s^e_{i+1,1}})}{10} + 0.5. \quad (21)$$

d) Operator Op_4

(1) Delete the working shift with probability based on how poorly the shift satisfies the employee's time preferences, that is

$$Prob^4_{\text{del}} = \frac{\sum_{t=s_{i,1}}^{s_{i,2}} (1 - \pi^e_t)}{2(s_{i,2} - s_{i,1} + 1)}. \quad (22)$$

e) Operator Op_5

(1) For every two consecutive shifts, compute the quantity

$$(s_{i+1,1} - s_{i,2} - 1) \bmod H. \quad (23)$$

(2) Use the resulting values in a roulette-wheel fashion, selecting one rest period.

(3) Move the corresponding shifts apart according to $Prob^3_{\text{left}}$, making it easy to create a new day-off.

f) Operator Op_6

(1) For idle periods (t_1, t_2) of length $\geq \sigma_s^{e_g}$, compute

$$\frac{1}{t_2 - t_1 + 1} \sum_{t=t_1}^{t_2} (1 + \pi^e_t). \quad (24)$$

(2) Select a period using the roulette method.

```
PROCEDURE HeuristicOperators

BEGIN
  FOR employee := 1 TO E DO
  BEGIN
    FOR ALL shifts DO
      Operator (SelectOne (v1, v2, v3, v4));
    Operator (SelectOne (v5, v6, v7));
  END
  Operator (8);
  Operator (9);
END ;
```

Fig. 3 Application of the heuristic operators. Overall, the procedure can be thought of as a sophisticated mutation operator.

(3) Create a shift of length $\sigma_s^{e_g}$.

g) Operator Op_7

(1) For all active shifts, determine how badly each working shift matches the employee's time preferences by computing the non-negative quantity

$$\sum_{t=s_{i1}}^{s_{i2}} (1 - \pi_t^e). \qquad (25)$$

(2) Select a shift by the roulette-wheel method.

(3) Delete it.

h) Operator Op_8

(1) For all $t = 1, 2, \ldots, T$, and every g, compute

$$p_1(t) = (\mathrm{int})(\gamma R_t - n_t^* + 0.5) \quad (26)$$

$$p_2(t) = (\mathrm{int})(R_t - n_t^* + 0.5) \quad (27)$$

$$p_3^g(t) = (\mathrm{int})(\gamma R_t^g - n_t^g + 0.5). \quad (28)$$

(2) Select one interval where $p_1 > 0$ or $p_2 > 0$.

(3) Depending on the interval length and p_3^g, extend neighbor shift or create a new one.

i) Operator Op_9: analogous to Op_8.

The usage of the heuristic operators can be summarized as illustrated in **Fig. 3**. First, for each active working shift of each employee, the non-negative penalty values $v_1(e_i)$, $v_2(e_i)$, $v_3(e_i)$, and $v_4(e_i)$ are used to construct a roulette wheel, from which one of the violations is selected randomly, and then the corresponding operator is executed. After repeating the action for all the working shifts of a given employee, the penalty values $v_5(e)$, $v_6(e)$, and $v_7(e)$ are used to generate another roulette wheel, which is then spun once, and the corresponding operator is put into effect. After repeating the action for all the employees, finally the operators Op_8 and Op_9, which act on the whole workforce schedule, are applied in sequence. Overall, the procedure can be thought of as a sophisticated mutation operator, with random behavior biased by heuristic rules in such a way to improve the schedule's quality.

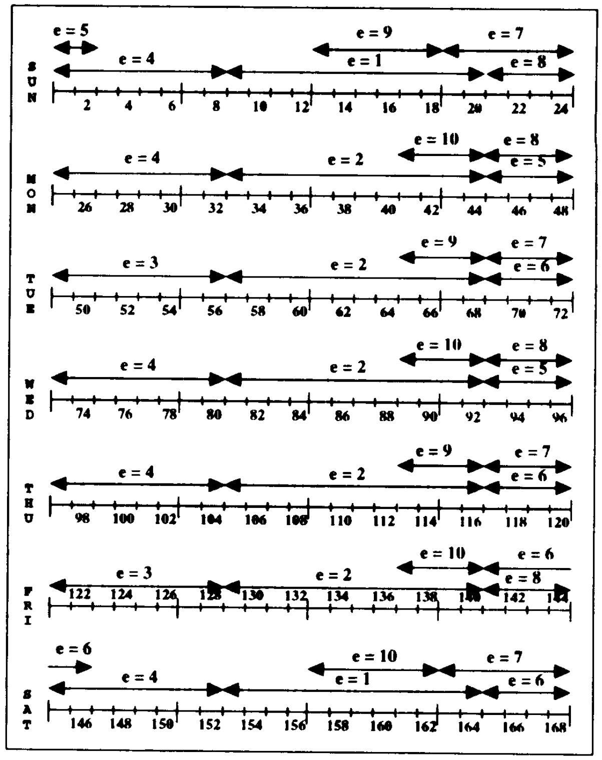

Fig. 4 Workforce schedule obtained by the proposed GA for a small CHWSP consisting of 10 employees of 3 groups over a planning horizon of 168 time intervals.

6. Experimental Results

The proposed GA for CHWSP was implemented as a C-language program in a workstation environment, and several simulations with realistic data were carried out. In all the simulations, selection took place using the stochastic remainder method[12], the crossover probability was 0.7, and the 10% top individuals were always forced to appear in the next generation, in an elitist strategy. The cost parameters ν^1 and ν^2 used in the computation of the violation costs v_8 and v_9 were set to the maximum wage corresponding to a single time interval. Note that in the proposed method there is no need to specify a mutation probability, since the probabilities are built in the heuristic operators.

First, the resulting workforce scheduling for a small CHWSP with $E = 10$ employees divided into $G = 3$ groups, $T = 168$ (this is equivalent to a whole week divided in one-hour intervals), and $S = 7$ is shown in **Fig. 4**. Since

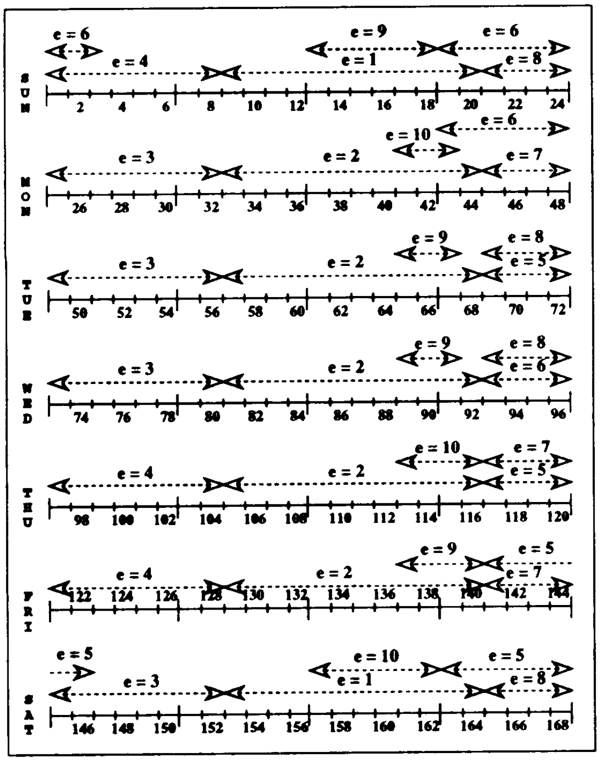

Fig. 5 Optimal workforce schedule obtained by hand. In terms of cost, both schedules are equivalent, indicating that the customized GA was able to find an optimal solution.

this is a small CHWSP, it can be solved optimally by a human, whose solution is shown in **Fig. 5**. Although there seem to be a few differences between the workforce schedules in Fig. 4 and Fig. 5, in fact both schedules turned out to be equivalent in cost terms, indicating that the solution obtained by GA is the optimal one☆, and was obtained after 50 generations of a population of 100 randomly-initialized chromosomes.

The problem defined as CHWSP was designed based on the workforce scheduling needs of actual companies with flexible working shifts. In this situation, the workforce requirements are only short-term forecasts made by company experts. Once forecasts are made and the corresponding workforce schedule is determined, there may be dynamical changes in the workforce requirements, depending on the particular problem.

To verify the robustness of the proposed method, the personnel requirements were changed after convergence to the optimal schedule had been achieved, and the results are shown in **Fig. 6**. The horizontal axes shows the number of generations, while the schedule costs are shown in the vertical. The curves represent the best of 10 runs with different initial values, and employ an elitist approach. The first part of the graph shows the convergence to the optimal schedule, which was obtained as early as in the 43rd generation in the best case. After the 60th generation, two independent disturbances were applied to the problem, corresponding to an increase and a decrease in the required workforce during the planning horizon, and the results are shown in Fig. 6 (a) and Fig. 6 (b), respectively. After an initial transient, both cases quickly converged to steady states with optimal costs☆☆, demonstrating the robustness of the proposed approach.

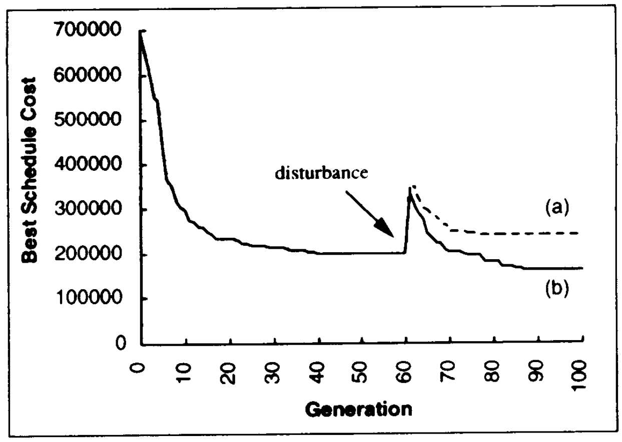

Fig. 6 Reaction to dynamic changes in the workforce requirements after convergence. After the 60th generation the workforce requirements were increased (a) or decreased (b) abruptly, but the system succeeded in converging to the new optima after a brief transient.

The performance of the proposed approach to solve the CHWSP was also compared with other methods in the literature, and the results obtained are shown in **Table 1**, where RW, BAB, SA, and GA stand for random walk, branch-and-bound[23], simulated annealing[17], and the proposed GA method. For the BAB method, schedules were generated randomly and all the neighboring possibilities were investigated until the total cost exceeded the cost of the best schedule obtained so far. For the SA, mutations were generated in a range proportional to the temperature, and 50 levels of temperature from 5 to 0.1 were used for anneal-

☆ The actual data consisted of more than 2,500 numbers, too many to be reproduced in this paper.

☆☆ The schedules obtained were confirmed to correspond to the optimal ones using an exhaustive method.

Table 1 Comparison among RW (random walk), BAB (branch-and-bound), SA (simulated annealing), and GA (proposed method) applied to the solution of a small-scale CHWSP. While both RW and BAB failed to find the optimal solution, SA and GA succeeded. The proposed method was by far the fastest one.

Search Method	Convergence to Optimal		Cost of the Best Schedule After 10,000s
	Time (s)	No. of Schedules	
RW	—	—	223,085
BAB	—	—	201,475
SA	482	24,613	197,900
GA	72	4,213	197,900

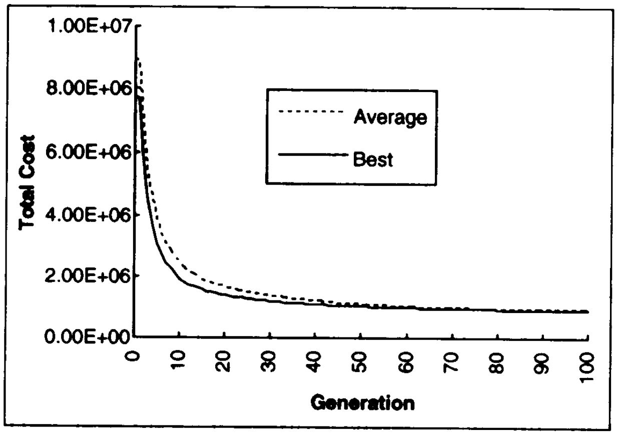

Fig. 7 Decrease of the total schedule cost for a large CHWSP with 50 employees. Further analysis indicated that high-quality solutions were obtained.

ing, together with the Metropolis criterion.

For 10 runs, the second column in Table 1 shows the average time at which the optimal solution was first determined, while the third column shows the number of workforce schedules which were evaluated until finding the optimal. Finally, the last column of the table gives the cost of the best solution found in 10,000 seconds of processing time. Both RW and BAB failed to find the optimal solution, while SA and GA succeeded. From the results, it is clear that the proposed approach was considerably faster than SA, and in average needed only about 1/6 of the number of schedule evaluations SA required to find the minimum.

Simulations were also carried out for a larger problem with $E = 50$ employees divided into $G = 4$ groups, planning horizon $T = 336$, and maximum number of working shifts per employee $S = 12$. This problem involves an enormous amount of data (almost 19,000 numbers), so in practice sub-optimal solutions are acceptable. The search space is also very large, with chromosomes made up of 1200 integers in the interval [0, 336]. The decrease of the overall schedule cost is shown in **Fig. 7** for a population of 100 chromosomes. The graph shows the average of 10 runs with different initial values. In all the experiments convergence was achieved in less than 200 generations. Although it is not practical to verify whether the best solution obtained after 200 generations corresponds to the absolute optimal, the values of all the penalty values decreased uniformly. Of course, this convergence alone does not guarantee that the solution obtained is good, but analysis of the resulting schedule and penalty values by a human confirmed the high quality of the obtained solution. The corresponding graphs were omitted here due to page restrictions, but allowed us to conclude that the customized GA succeeded in generating high-performance workforce schedules for the problem, confirming the effectiveness of the proposed approach. Concerning the execution time, each run of 200 generations took approximately 10 minutes on a workstation computer.

7. Concluding Remarks

A difficult workforce scheduling problem (CHWSP) with *real-world* constraints was defined, and an approach to solve it by a customized genetic algorithm was proposed. The problem involves a great number of variables, rather complex data structures for program implementation, and several probabilistic-heuristic operators described in this paper. In comparison with similar problems proposed in the literature, CHWSP is more general, using the concept of flexible working shifts and data individualized on a per-employee basis.

The customized GA is characterized by a sophisticated mutation operator composed by a series of operators which compete with each other to improve the quality of a workforce schedule. The GA operators were designed independently, in such a way to facilitate the addition of new operators. Therefore, the design of more sophisticated operators is a natural avenue for improvement of the system.

The system has been implemented and experimental results have shown that, for problems of moderate size which usually require hours or days of a human expert, good schedules can be obtained in a few minutes. For very large problems, however, computation time may become a problem. A parallel implementation using the idea of population partition among several processors of a parallel computer is currently under investigation. Additionally, further specialization of CHWSP for dealing with more particular situations as, for example, working schedules of nurses or aircraft crews, is being considered.

Several parameters are built in the definitions of the operators. As a matter of fact, proper parameter setting is a problem with all stochastic optimization methods, and there is no general theory indicating how to set the probabilities of crossover and mutation in GAs for optimal results. All the operator probabilities were set using heuristic reasoning. A small batch of experiments performed with slightly different values for the parameters did not produce significantly different results for the two problems illustrated in the paper, suggesting that the performance of the proposed method is not highly sensitive to the parameter values. However, this is only an experimental indication, and a complete analysis of the sensitivity of GAs with respect to the probabilities of crossover and mutation is out of the scope of this paper.

References

1) Bagchi, S., Uckun, S., Miyabe, Y. and Kawamura, K.: Exploring Problem-Specific Recombination Operators for Job Shop Scheduling, *Proc. Fourth Int. Conf. on Genetic Algorithms*, pp.10–17 (1991).
2) Bartholdi, J.: A Guaranteed-Accuracy Round-Off Algorithm for Cyclic Scheduling and Set Covering, *Operations Research*, Vol.29, No.3, pp.501–510 (1981).
3) Bechtold, S., Brusco, M. and Showalter, M.: A Comparative Evaluation of Labor Tour Scheduling Methods, *Decision Sciences*, Vol.22, pp.683–699 (1991).
4) Bhuyan, J.N., Raghavan, V.V. and Elayavalli, V.K.: Genetic Algorithm for Clustering with an Ordered Representation, *Proc. Fourth Int. Conf. on Genetic Algorithms*, pp.408–415 (1991).
5) Cleveland, G.A. and Smith, S.F.: Using Genetic Algorithms to Schedule Flow Shop Releases, *Proc. Third Int. Conf. on Genetic Algorithms*, pp.160–169 (1989).
6) Dantzig, G.: A Comment on Edie's Traffic Delay at Toll Booths, *Operations Research*, Vol.2, No.3, pp.339–341 (1954).
7) Davis, L.: Job Shop Scheduling with Genetic Algorithms, *Proc. First Int. Conf. on Genetic Algorithms*, pp.136–140 (1985).
8) Easton, F. and Mansour, N.: A Distributed Genetic Algorithm for Employee Staffing and Scheduling Problems, *Proc. Fifth Int. Conf. on Genetic Algorithms*, Morgan-Kaufmann, pp.360–367 (1993).
9) Easton, F. and Rossin, D.: Equivalent Alternate Solutions for the Tour Scheduling Problem, *Decision Sciences*, Vol.22, pp.985–1007 (1991).
10) Easton, F. and Rossin, D.: Sufficient Working Subsets for the Tour Scheduling Problem, *Management Sciences*, Vol.37, No.11, pp.1441–1451 (1991).
11) Fang, H.L.: Investigating Genetic Algorithms for Scheduling, Master's Thesis, Univ. of Edinburg (1992).
12) Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley (1989).
13) Grefenstette, J.J.: Incorporating Problem Specific Knowledge in Genetic Algorithms, *Genetic Algorithms and Simulated Annealing*, Davis, L. (Ed.), Morgan Kaufmann, pp.42–57 (1987).
14) Grefenstette, J.J., Gopal, R., Rosmaita, B. and Gucht, D.V.: Genetic Algorithms for the Traveling Salesman Problem, *Proc. First Int. Conf. on Genetic Algorithms*, pp.160–168 (1985).
15) Holland, J.H.: *Adaptation in Natural and Artificial Systems*, Univ.of Michigan Press (1975).
16) Husbands, P. and Mill, F.: Simulated Co-Evolution as the Mechanism for Emergent Planning and Scheduling, *Proc. Fourth Int. Conf. on Genetic Algorithms*, pp.264–270 (1991).
17) Kirkpatrick, S., Gellat, Jr., C.D. and Vecchi, M.P.: Optimization by Simulated Annealing, *Science*, Vol.220, pp.671–680 (1983).
18) Krishnakumar, K. and Goldberg, D.E.: Control System Optimization Using Genetic Algorithms, *Journal of Guidance, Control, and Dynamics*, Vol.15, No.3, pp.735–740 (1992).
19) Li, C., Robinson, E. and Mabert, V.: An Evaluation of Tour Scheduling Heuristics with Differences in Employee Productivity and Cost, *Decision Sciences*, Vol.22, pp.700–718 (1991).
20) Mabert, V. and Watts, C.: A Simulation Analysis of Tour Shift Construction Procedures,

Management Sciences, Vol.28, No.5, pp.520–532 (1982).

21) Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag (1992).

22) Nordvik, J.P. and Renders, J.M.: Genetic Algorithms and Their Potential for Use in Process Control: A Case Study, *Proc. Fourth Int. Conf. on Genetic Algorithms*, pp.480–486 (1991).

23) Rich, E.: *Artificial Intelligence*, McGraw-Hill (1983).

24) Srinivas, M. and Patnaik, L.M.: Genetic Algorithms: A Survey, *Computer*, Vol.27, No.6, pp.17–26 (1994).

25) Vignaux, G.A. and Michalewicz, Z.: A Genetic Algorithm for the Linear Transportation Problem, *IEEE Trans. Systems, Man, and Cybernetics*, Vol.21, No.2, pp.445–452 (1991).

(Received September 14, 1995)
(Accepted May 10, 1996)

Julio Tanomaru graduated in Electronics Engineering from the Aeronautics Institute of Technology (ITA), Brazil, in 1986, worked for two years in a major Brazilian telecommunications company, and received the M.E. degree in computer sciences from the University of Tokushima, Japan, in 1992. Currently, he is a research associate at the University of Tokushima, Department of Information Science and Intelligent Systems, where he is also pursuing his doctoral degree. His current research interests include parallel processing, applications of evolutionary algorithms, neural networks for pattern recognition and adaptive control, and optimization problems. He is a member of the IEEE Control and SMC societies, ISCA, IPS of Japan, SICE, and iSCIe.