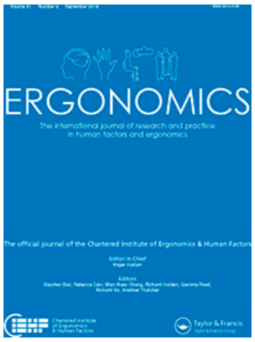


Ergonomics

ISSN: 0014-0139 (Print) 1366-5847 (Online) Journal homepage: https://www.tandfonline.com/loi/terg20

Designing safe job rotation schedules using optimization and heuristic search

Brian J. Carnahan , Mark S. Redfern & Bryan Norman

To cite this article: Brian J. Carnahan , Mark S. Redfern & Bryan Norman (2000) Designing safe job rotation schedules using optimization and heuristic search, Ergonomics, 43:4, 543-560, DOI: 10.1080/001401300184404

To link to this article: https://doi.org/10.1080/001401300184404

ERGONOMICS, 2000, VOL. 43, NO. 4, 543–560

Technical Note

Designing safe job rotation schedules using optimization and heuristic search

BRIAN J. CARNAHAN†*, MARK S. REDFERN‡ and BRYAN NORMAN§

†Department of Industrial and Systems Engineering, 207 Dunstan Hall, Auburn University, Auburn, AL 36849-5346, USA

‡Department of Bioengineering, The University of Pittsburgh, PA, USA

§Department of Industrial Engineering, The University of Pittsburgh, PA, USA

Keywords: Job rotation scheduling; Integer Programming; Genetic algorithms.

Job rotation is one method that is sometimes used to reduce exposure to strenuous materials handling; however, developing effective rotation schedules can be complex in even moderate sized facilities. The purpose of this research is to develop methods of incorporating safety criteria into scheduling algorithms to produce job rotation schedules that reduce the potential for injury. Integer programming and a genetic algorithm were used to construct job rotation schedules. Schedules were comprised of lifting tasks whose potential for causing injury was assessed with the Job Severity Index. Each method was used to design four job rotation schedules that met specified safety criteria in a working environment where the object weight, horizontal distance and repetition rate varied over time. Each rotation was assigned to a specific gender/lifting capacity group. Five versions of the integer programming search method were applied to this problem. Each version generated one job rotation schedule. The genetic algorithm model was able to create a population of 437 feasible solutions to the rotation problem. Utilizing cluster analysis, a rule set was derived from the genetic algorithm generated solutions. These rules provided guidelines for designing safe job rotation schedules without the use of a computer. The advantages and limitations of these approaches in developing administrative controls for the prevention of back injury are discussed.

1. Introduction

In a manufacturing or service environment, job scheduling is an essential component in maintaining the profitability of the enterprise. Within a schedule, the timing and sequencing of operations that humans and machines perform must strike a balance between productivity demands (i.e. delivery of output of a certain type at a certain time and place) and the safety concerns of the personnel involved in meeting these demands. When these safety requirements are disregarded, the profit made by the operation can be substantially diminished due to increased operating costs incurred by worker injury (Snook 1978). To adequately address this issue, schedules for certain human-machine operations can be developed that include

---

*Author for correspondence. e-mail: carnahan@ eng.auburn.edu

Ergonomics ISSN 0014-0139 print/ISSN 1366-5847 online © 2000 Taylor & Francis Ltd
http://www.tandf.co.uk/journals/tf/00140139.html

systematic variations in task exposure hours of the involved employees. These variations are designed to ensure that the demands of the job do not exceed the capabilities of the workers. Such scheduling of operations to ensure worker safety is known as job rotation.

A planner may encounter several challenges in creating effective job rotation schedules for tasks involving lifting. One such difficulty is that in realistic industrial operations the demands of each task (i.e. the weight of the load or the rate of lifting) may not be constant. Trying to establish a balance between work and employee capacity can prove extremely difficult in situations where work demands are variable. Obstacles may also arise in trying to establish a job rotation plan for a diverse working population. The same sequence of tasks that is relatively safe for one individual, given a high lifting capacity (i.e. 90th percentile male), may place a physically weaker individual (i.e. 25th percentile male) at substantial risk of back injury. Another impediment is identifying the 'best' job rotation plan out of all possible solutions that can exist, given a set of circumstances.

Job rotation plan development would be facilitated if scheduling rules could be utilized by the planner. Such rules, if they existed, would have to provide guidance as to the maximum number of hours per day a worker could safely perform an industrial task. The rules would have to take the following factors into account: (1) the physical demands of the task; (2) the gender of the worker; (3) the lifting capacities of the worker; and (4) the demands and time spent performing other tasks. The last factor listed is critical in that these rules would interact with one another as a *rule set*. Without these rules, and under production constraints in which each task must be undertaken throughout the workday, the number of potential job rotation schedules even for small groups of workers can become very large.

The potential number of possible solutions to a job rotation problem and the need for developing computational methods can be best illustrated by the following example. The objective in this example is to develop a rotation schedule for a manufacturing cell for a 5-day work week where all tasks are completed and the risk of back injury to the employees involved is minimal. The cell is comprised of four operations (A, B, C, D). Each operation involves performing several lifting tasks for 50 min out of every hour. The weight of load lifted, the rate of lifting, and the horizontal distance of the load's centre of mass from the ankles all vary within each hour and vary over the course of an 8-h shift. In order to provide an example of how the job demands may vary over time, the subtask demand descriptions for Task A are defined in table 1. It is assumed that the job characteristics vary uniformly across their range of values. For example, the weights of objects in subtask FK of Task A vary uniformly between 35 and 37 kg during the time interval from 08:00 h to 12:00 h. Four gender capacity groups staff the cell (table 2). Each operation must be undertaken by a single employee throughout the workday. Finally, an employee's shift can be broken down into eight, 1-h time periods. An example of a job rotation schedule for this manufacturing cell is presented in table 3.

Given that the assignments of tasks to employees within each time period are independent of one another, the total number of potential job rotation schedules available to the planner is quite large (110, 075, 314, 200). With over 110 billion possible solutions, using enumeration to evaluate all possible job rotation schedules is impractical. The problem of assigning the tasks to employees falls within the category of problems known

Table 1. The subtask demand descriptions for Lifting Task A.

	Object weight (kg)		Repetition rate (lifts/min)			Distance (cm)	
Subtask	08:00 to 12:00h	13:00 to 17:00h	08:00 to 11:00h	11:00 to 14:00h	14:00 to 17:00h	08:00 to 12:00h	13:00 to 17:00h
FK_A	35 – 37	25 – 37	0.1 – 2.0	1.0 – 3.0	0.1 – 1.0	30 – 40	18 – 20
SO_A	23 – 27	27 – 31	7.0 – 11.0	3.0 – 6.0	4.0 – 7.0	15 – 20	17 – 22
KS_A	18 – 22	13 – 16	2.0 – 6.0	5.0 – 8.0	2.0 – 4.0	20 – 27	10 – 20

FK_A = lifting from floor to knuckle height.
SO_A = lifting from shoulder height to overhead.
KS_A = lifting from knuckle height to shoulder height.

Table 2. Gender and lifting capacity of employees in manufacturing cell.

Employee	Gender	Lifting capacity
1	Male	50th Percentile
2	Female	90th Percentile
3	Male	25th Percentile
4	Female	50th Percentile

Table 3. Example of a job rotation schedule for Lifting Tasks A, B, C and D.

Time period	50th Percentile male	90th Percentile female	25th Percentile male	50th Percentile female
08:00 – 09:00	Task D	Task A	Task C	Task B
09:00 – 10:00	Task C	Task A	Task B	Task D
10:00 – 11:00	Task C	Task B	Task D	Task A
11:00 – 12:00	Task A	Task D	Task C	Task B
13:00 – 14:00	Task D	Task C	Task A	Task B
14:00 – 15:00	Task A	Task D	Task C	Task B
15:00 – 16:00	Task A	Task C	Task D	Task B
16:00 – 17:00	Task C	Task D	Task B	Task A

as combinatorial optimization problems. Traditionally, these types of problems have been solved using integer programming (Pinedo 1995). However, as will be discussed more completely in §2, it is difficult to use integer programming to develop job rotation schedules when there are dynamic task characteristics. Fortunately, there exists a set of powerful heuristic computational procedures, known as genetic algorithms (GAs), that are designed to search and find answers to complex problems in situations where the number of possible alternative solutions is vast and the problem environments are dynamic. GAs, as described by Holland (1992: 54 – 58), utilize the principles of natural selection and survival of the fittest to 'evolve' a population of good solutions to a particular problem or set of problems. Developing numerous good job rotation schedules benefits the planner in two distinctive ways. First, multiple solutions provides one with greater flexibility for planning work over long periods of time and can take into account situations where there are last minute changes in operations or personnel staffing. A second distinct advantage lies in identifying the commonalties that exist between the good solutions that are collected by the GA. These common traits can be used to derive a set of rules (i.e. guidelines) that reasonably explain these solutions. The purpose of this study was to design a genetic algorithm to evolve a wide variety of solutions to the previously described job rotation problem. These solutions would then be compared to those generated by using integer programming. Once the GA generated solutions were obtained, inductive interference via cluster analysis would be carried out to develop sets of scheduling rules. These rules act as guidelines for determining the task exposure hours for both male and female workers of various lifting capacities. The efficacy of the generated rules was then tested by using them to create a new job rotation schedule without the use of a GA or a computer. This schedule was then evaluated in regard to the potential risk of back injury.

The Job Severity Index (JSI) as describd by Liles (1986) was used to assess the potential for back injury. The JSI measure is sensitive to the lifting capacity of the individual performing the tasks, as well as the weight of the object lifted, rate of lifting and horizontal distance of the object from the ankles at the start. The JSI is a unitless ratio of weight lifted to worker capacity. As JSI increases, the risk of back injury increases. Research by Ayoub and Selan (1983) and Liles *et al.* (1984) suggest that jobs with JSIs that exceeded 1.5 were associated within increased incidence and severity of back injuries when compared to jobs where the JSI was 1.5 or less.

2. Method

Three methods of generating job rotation schedules that incorporate the JSI criteria are described in this section. The first is integer programming (IP), which is the method that has been traditionally used for job rotation problems. The second is a genetic algorithm that more readily captures the dynamic nature of the problem. The third is a clustering method that is used to develop job rotation rules from the GA results.

2.1. *Integer programming approaches to job rotation*

An integer programme (IP) to develop job rotation schedules can be developed using a set of decision variables denoted X_{ijt},

$$\text{where } X_{ijt} = \begin{cases} 1 \text{ if worker } i \text{ performs} \\ \text{task } j \text{ during time period } t \\ 0 \text{ otherwise} \end{cases}$$

In this study all of the IPs were comprised of 4 (gender capacity groups) × 4 (lifting tasks) × 8 (time periods) = 128 decision variables. The purpose of each integer programme was to find values for the 128 decision variables so that the maximum JSI found across the schedules of each of the four gender capacity groups was minimized. The general formulation for the IP is shown in appendix 1.

The cost variable c_{ijt} is a real number, which represents the cost, in terms of JSI, of worker i performing lifting task j during time period t. The first set of constraints ensures that each lifting task is manned by a single worker across each of the eight time periods. The second set of constraints ensures that each worker performs only one lifting task in each time period. The third set of constraints considers the overall task demands placed upon each gender capacity group based on the job assignments and calculates the JSI value for each gender capacity group. The minimization objective forces the IP to find the set of X_{ijt} values corresponding to the job rotation schedule that minimizes the maximum JSI found among the four gender capacity groups.

A limitation of integer programming is that it is designed to optimize deterministic problems or problems where the data is known with certainty. Unfortunately, the problem described in §1 and demonstrated in tables 1, 2 and 3 does not have this characteristic; because the object weights, distances and lifting rates are stochastic it is not possible to determine the coefficient values necessary to formulate the integer programme. One way to handle this difficulty is to use the expected values for each of the task characteristics. This permits the development of the necessary coefficients to construct the IP. Using this type of methodology five IPs were applied to the job rotation problem. In each of the five variations of the IP the value of c_{ijt} was determined in a different manner. In the first variation (VAR50), the cost variable was a partial JSI calculation obtained by setting the ranges of object weight, horizontal distance of the object from the ankles at the start of the lift, and lifting rate to their 50th percentile or expected values. The next three variations (VAR75, VAR90, VAR100) were developed in a similar manner by setting the task demand variables to their 75th, 90th and 100th (maximum) percentile values, respectively. The fifth IP variation (VARWTAVG) attempts to find a job rotation schedule that minimizes the maximum JSI found under all four task demand conditions used by VAR50, VAR75, VAR90 and VAR100. This is done by using the c_{ijt} values from VAR50, VAR75, VAR90 and VAR100 giving each equal weighting. Thus, this IP determines the solution that is best across all four scenarios. Each IP variation was designed to generate a single job schedule. The five variations were evaluated using CPLEX (ILOG Inc., Cambridge, MA) software and run on a UNIX (Sun Microsystems, Palo Alto, CA) system.

2.2. *The genetic algorithm approaches to job rotation*

The following is a brief description of the GA stages used to acquire multiple solutions to the job rotation problem. The development of the GA procedure used here is based on the general method proposed by Davis (1991: 1–22).

2.2.1. *Stage 1—creating the environment*: The environment in which the solutions exist is dynamic in the sense that it is based on the fluctuation of subtask characteristics that comprise the four major lifting operations of the manufacturing cell. The three parameters that were allowed to vary over time were weight of the object lifted, rate of lifting, and horizontal distance of the object from the ankles at the start of the lift (recall the example for Task A in table 1). For all four operations, the genetic algorithm utilizes a random number generator to create the weight, rate and distance values assuming a uniform probability distribution over the ranges.

2.2.2. *Stage 2—establish the initial population of solutions*: Within this stage, the GA generates 195 initial job rotation solutions. Initial generation entails randomly assigning one of the four operations to a single employee across each of the eight time periods. Each solution is represented as a linear array of 32 characters (figure 1). The array is divided into eight sections with each corresponding to a single time period during a shift. Within each section there are four subsections (table 4).

In the initial job rotation solution illustrated in figure 1 and table 4, the 08:00 to 09:00 h time-period contains: the 50th percentile male performing Task D, the 90th percentile female performing Task A, the 25th percentile male performing Task C, and the 50th percentile female performing Task B. In

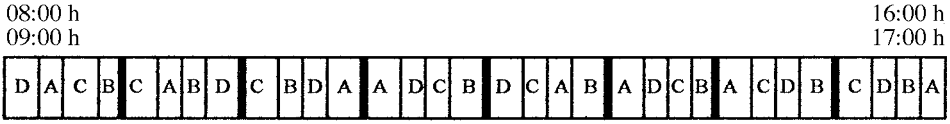


Figure 1. A 32-character array representing a job rotation schedule solution.

Table 4. Worker assignment to lifting tasks within the job rotation schedule solution.

Subsection	Represention
1	Task performed by the 50th percentile male
2	Task performed by the 90th percentile female
3	Task performed by the 25th percentile male
4	Task performed by the 50th percentile female

order to maintain the production constraints of the manufacturing cell each section of the representation contains tasks A, B, C and D in a specific sequence. In order to complete Stage 2, a genetic tag (an integer equal to 1, 2 or 3) is randomly assigned to each of the 195 solutions. Each tag is a marker representing one of three potential 'species' of solutions and will play an integral part in the fitness and selection operation of Stage 3.

2.2.3. *Stage 3—fitness assessment and selection*: In the third stage of the algorithm the 'fitness' of each solution is evaluated. Within the context of the job rotation problem, a solution's fitness is based on the potential risk of back injury encountered by each of the four employees. Each solution possesses three levels of fitness. The following is a description of the fitness measurements used.

1. *Standardized fitness*. A separate Job Severity Index (JSI) is calculated for each of the four types of employees within the job rotation schedule. Owing to the variability of the task demands the JSI will fluctuate over evaluations. Therefore, a sampling procedure was implemented in order to calculate the standardized fitness. The first step was to calculate the standardized fitness for each employee type by taking a sample of 10 JSI measures based on randomly generating each sample from the task characteristic distributions. Then the squared Euclidean distance between these samples and a constant of 1.5 (1.5 was selected because it is the JSI cut-off value to minimize the likelihood of back injury) was determined for each worker. This distance was then scaled between 0 and 1.0, with a value of 1.0 indicating that the JSI samples were very close to 1.5, using the expression below where JSI_k is the JSI value for sample k for worker i. JSI-Diff_i is the standardized fitness for each worker/employee i.

$$\text{JSI-Diff}_i = \frac{1}{1 + \sum_{k=1}^{10} (1.5 - \mathrm{JSI}_k)^2}$$

The second step was to calculate the fitness between the four employee types in order to obtain a single measure of standardized fitness. This was accomplished by determining the squared

Euclidean distance between the standardized fitness of the four employee types and a constant of 1. This measure is also scaled to be between 0 and 1, with 0 indicating that the rotation schedule results in high (and low) JSI values for one or more employees and 1 indicating that the rotation schedule results in JSI measures very close to 1.5 for all involved employees. The scaling equation is represented by the expression below.

$$\text{Standardized-Fitness} = \frac{1}{1 + \sum_{i=1}^{4}(1.0 - \text{JSI-Diff}_i)^2}$$

2. *Species fitness*. The standardized fitness of each individual solution divided by the total number of solutions bearing the same marker (i.e. genetic tag). An increase in the number of solutions with the same tag will result in a linear decrease in the individual fitness of each solution bearing that tag. This procedure allows the algorithm to search for multiple solutions in parallel (Spears 1994).

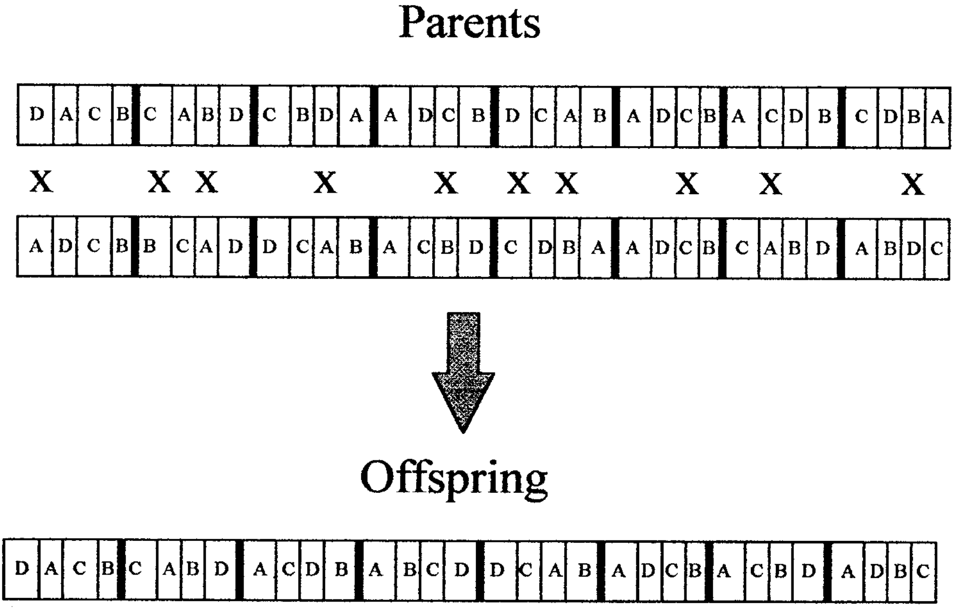
3. *Relative fitness*. The species fitness of a single solution divided by the sum total of the species fitness for all 195 solutions.

Selection was conducted using fitness-based stochastic selection with replacement (i.e. roulette wheel selection) (Goldberg 1989: 2–7) utilizing the relative fitness of each solution. Using this selection procedure, the most fit schedules have the highest probability of being selected for survival in the next generation of solutions.

2.2.4. *Stage 4—reproduction*: Within this stage, pairs of selected solutions are randomly chosen to undergo crossover, a procedure in which two schedules will combine and/or exchange task assignments to produce two 'offspring' solutions. An example of crossover is shown in figure 2. Under this type of crossover, referred to as partially mapped crossover (Michaelwicz 1994: 105–110), offspring can be produced that have sequence (i.e. task assignment) characteristics of both parents. This mechanism is used to generate two rotation schedules that replace their parents within the population. Under mutation, solutions within the population are randomly chosen. A selection along the representation (a task assignment within a particular time period) is selected, again at random, and the characters in two subsections will switch positions, altering the task assignment within that particular time period. As with crossover, the mutated replaces its parent within the population as well.

Within a single run of the GA, Stages 3 and 4 are executed in a cyclic fashion. The completion of a single cycle represents a generation. The GA performed 10 separate runs, allowing the development of 300 generations per run. Within every generation, three schedules (each representing the highest standardized fitness score of each species) were recorded to a single file. Once all ten runs were completed this file was scanned and all unique solutions (i.e. no duplicate schedules), with a standardized fitness greater than or equal to 0.98, were saved to a separate second file. The cut-off value of 0.98 was selected based on preliminary test results. The GA used in this particular application was written in C++ and run on an 486 IBM PC.

Once collected, the solutions found by the GA and the five integer programmes were compared to one another under the following workplace scenarios: (1) all task demand variables are set to their 50th percentile values across all time periods; (2) all task demand vari-



Parents

DACB CABD CBDA ADCB DCAB ADCB ACDB CDBA

X X X X X X X X X X

ADCB BCAD DCAB ACBD CDBA ADCB CABD ABDC

Offspring

DACB CABD ACDB ABCD DCAB ADCB ACBD ADBC

Figure 2. Reproduction in which two solutions create an offspring solution (X= crossover site randomly assigned between two parents).

ables are set to their 75th percentile values across all time periods; (3) all task demand variables are set to their 90th percentile values across all time periods; and (4) all task demand variables are set to their 100th percentile or maximum values across all time periods. Each job rotation schedule was evaluated by recording the maximum JSI value found among the four gender capacity groups. This measure was then used to compare the GA-generated solutions to the IP-generated solutions across each of the four workplace scenarios.

2.3. *Cluster analysis of GA-generated job rotation solutions*

Cluster analysis was used to create clusters such that each cluster would contain a set of task assignment hours that were similar in regard to duration. Determination of these clusters facilitated the derivation of a general set of rules governing task exposure for each gender capacity group.

2.3.1. *Exposure hour rules—the Kohonen self-organizing map*: In order to effectively cluster the unique GA-generated solutions based upon task exposure hours, the current representations had to be altered in form. Specifically, the 32-character linear array had to be changed to a continuous-valued 16-dimensional linear array (figure 3). The 16 numbers within this array represent the hours that each of the four employee types spent in Tasks A, B, C and D. The example illustrated in figure 3 represents a single job rotation schedule where the 50th percentile male works 3 h in Task A, no hours in Task B, 3 h in Task C, and only 2 h in Task D. All unique solutions were converted to this new representation. The exposure hours for each of the four employee types were saved in four separate data files.

To cluster the real valued vectors, representing the task exposure hours for a gender/capacity group, an artificial neural network was employed. This

	50th percentile Male				90th percentile Female				25th percentile Male				50th percentile Female			
	3	0	3	2	2	1	2	3	1	2	3	2	2	5	0	1
Tasks:	**A**	**B**	**C**	**D**												

Figure 3. Sixteen-dimensional array representing the task exposure hours for a job rotation solution.

type of network is known as a Kohonen Self-Organizing Map. Fausett (1994: 169 – 187) describes the network's architecture as being used to cluster continuous valued vectors into a pre-established set of cluster nodes. This is accomplished by minimizing the Euclidean distance between the input signal and the input weight vector of each cluster. The result is that each cluster would contain a set of task assignment hours that were similar in regard to duration.

2.3.2. *Solution templates—nearest-neighbour clustering algorithm*: Clustering the unique solutions based upon task sequence required the use of a modified nearest-neighbour clustering algorithm. According to Jain and Dubes (1988: 128 – 129), the nearest-neighbour clustering algorithm can group together (i.e. cluster) similar signals or patterns without initial information regarding the possible number of clusters present in the data. The nearest-neighbour clustering algorithm used in this experiment was designed to cluster together those job rotation schedules that shared at least 11 (one-third) or more of the same task assignments within the same time periods. The nearest-neighbour clustering algorithm was carried out using the four steps shown in appendix 2. Each of the unique job rotation solutions was subjected to clustering under the nearest-neighbour clustering algorithm. The purpose of this analysis was to obtain partial solutions (i.e. solution templates) whose task assignments were common among a group of rotation schedules. Solution templates were coupled with the 'rules' governing task exposure hours in an attempt to find new good solutions without the use of a genetic algorithm or computer.

3. Results

3.1. *Performance of the genetic algorithm*

The genetic algorithm was able to discover 437 unique solutions to the job rotation scheduling problem. A plot of the mean fitness of the sub-populations across generations revealed that the algorithm was successful in using simulated evolution to develop increasingly better solutions (figure 4). The algorithm was able to develop highly fit job rotation schedules under conditions in which certain task characteristics were random variables. This evidence supports the contention that job demands do not have to be static in order for the GA to obtain design solutions. An example of a highly fit job rotation schedule is presented in table 5.

Over all ten runs the GA evaluated 1 194 662 job rotation schedules. This sample is approximately 0.001% of the total number of solutions available.

3.2. *Task exposure hour rules*

The Kohonen Self-Organizing Map was successful in clustering the task exposure hour vectors for all four gender capacity groups. Rule sets were derived from the data clusters for all four gender/lifting capacity groups. Figure 5 displays the 21 unique task exposure hour vectors for

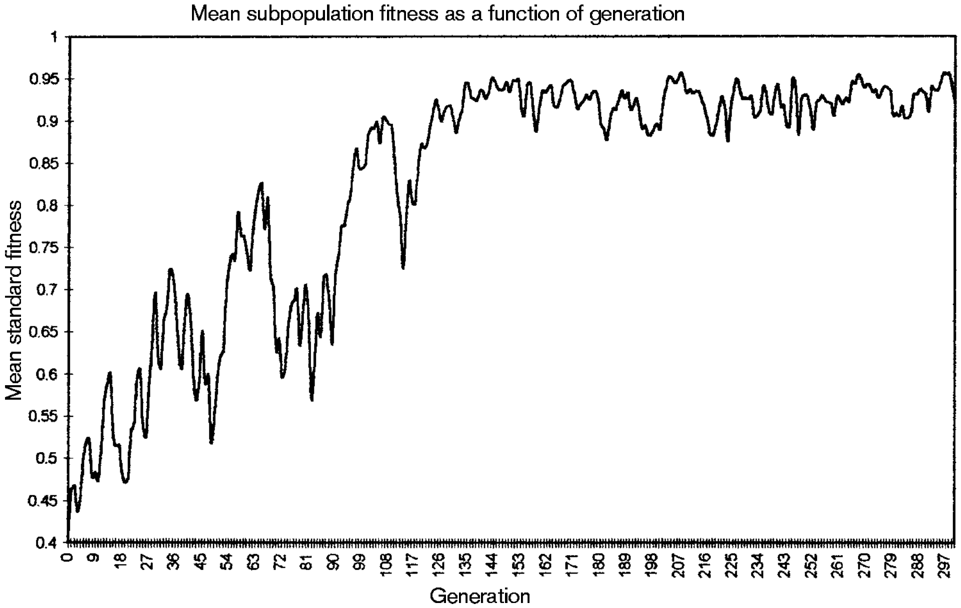


Figure 4. The mean fitness for a sub-population of solutions across generations.

Table 5. A highly fit job rotation solution.

Time period	50th Percentile male	90th Percentile female	25th Percentile male	50th Percentile female
08:00 – 09:00	Task C	Task A	Task D	Task B
09:00 – 10:00	Task C	Task A	Task D	Task B
10:00 – 11:00	Task A	Task B	Task C	Task D
11:00 – 12:00	Task D	Task B	Task C	Task A
13:00 – 14:00	Task C	Task A	Task D	Task B
14:00 – 15:00	Task C	Task A	Task D	Task B
15:00 – 16:00	Task C	Task A	Task D	Task B
16:00 – 17:00	Taso D	Task C	Task B	Task A
Mean JSI	1.54	1.53	1.49	1.49
SD JSI	0.05	0.05	0.07	0.05

males with 50th percentile lifting capacity.

The rules that apply to the task exposure hour vectors in figure 5 are listed below.

- No more than 4 h spent in *Task A*.
- No more than 1 h spent in *Task B*. If = 0 h, then no more than 3 h spent in Task A, no less than 1 h in Task D.

  If = 1 h, then no more than 2 h spent in Task D.
- No less than 2 and no more than 7 h spent in *Task C*.
- No more than 3 h spent in *Task D*. If = 0 h, then 1 h spent in Task B, and at least 3 h spent in Task C.
- The combined hours of Tasks A and C should be between 5 and 7.

Task			
A	**B**	**C**	**D**
2	1	5	0
3	0	3	2
1	1	6	0
3	1	4	0
3	0	2	3
1	1	5	1
2	0	4	2

Task			
A	**B**	**C**	**D**
1	0	5	2
0	1	7	0
2	0	5	1
1	0	4	3
2	0	3	3
2	1	4	1
0	0	6	2

Task			
A	**B**	**C**	**D**
0	0	5	3
4	1	2	1
3	1	3	1
0	0	7	1
3	1	2	2
4	1	3	0
1	0	6	1

Figure 5. Task exposure hour vectors: males with 50th percentile lifting capacity.

3.3. *Task assignments over time*

The 437 job rotation schedules were broken down into specific task sequences for each of the four gender capacity groups. For each group, the unique task sequences were counted and collected. Frequency distributions, showing the number of sequences that had a specific task being performed at a given time, were established for all eight time periods. In the 437 solutions collected by the GA there were 244 unique task sequences for the 90th percentile female and frequency distributions for the 90th percentile female are presented in table 6. A review of the distributions indicate that Task A is most frequently performed by the 90th percentile female. For Task B, the scheduling preference appears to be for the 13:00 to 14:00 h and the 16:00 to 17:00 h time periods. For Task C, the scheduling preference appears to be for the 10:00 to 12:00 h time periods. For Task D the scheduling preference appears to be for the 08:00 to 09:00 and 11:00 to 12:00 h time periods.

3.4. *Task sequence patterns*

When developing a job rotation schedule, assigning a given task to a gender capacity group within a preferred time period gives the planner the greatest number of options for scheduling in subsequent time periods. The tree diagram in figure 6 that clusters the task sequences for the 50th percentile female illustrates this point. The tree diagram shows that the greatest flexibility in scheduling the 50th percentile female group (i.e. the largest number of sequence options) is achieved when assigning Task B to the 09:00 to 10:00 h time period.

3.5. *Discovering new solutions without a genetic algorithm*

The nearest-neighbour clustering algorithm was successful in forming 58 clusters of job rotation schedules that had at least 11 identical task assignments at the same time periods. These clusters represent partial job rotation solutions as seen in the following examples. Using the cluster in table 7 as a template, a new rotation schedule can be formed. Although there are 663 552 possible solutions that contain this template, not all of them may be safe in regard to the prevention of back injury. In this instance, task assignments were made in conjunction with two primary considerations. The first was based on the exposure hour rules for the four gender capacity groups that were previously discussed. The second consideration was to base assignments on the scheduling preferences taken from the frequency

Table 6. Task sequences across time periods for the 90th percentile female.

Time period	Task A	Task B	Task C	Task D	Total
08:00 – 09:00	**121**	48	33	**42**	244
09:00 – 10:00	**145**	47	42	10	244
10:00 – 11:00	**112**	37	**58**	37	244
11:00 – 12:00	**88**	39	**72**	**45**	244
13:00 – 14:00	**110**	**59**	47	28	244
14:00 – 15:00	**141**	43	37	23	244
15:00 – 16:00	**134**	47	51	12	244
16:00 – 17:00	**111**	**69**	39	25	244

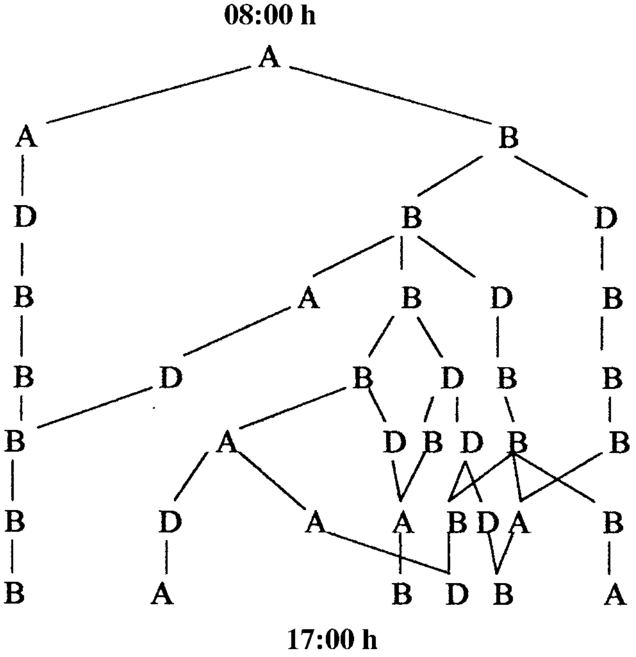


Figure 6. Sequences for the 50th percentile female beginning with Task A.

distributions of the task sequences. Using these two considerations the job rotation solution of table 8 was found without the use of a computer. Once constructed, the new job rotation solution was evaluated taking 100 JSI samples for each of the four gender capacity groups and calculating the mean and the standard deviation (table 8). This new solution is reasonable since the JSIs are not excessive for any of the four gender capacity groups. Additional solutions could be found by following the rules established through cluster analysis and conducting a local search of the solution space using the template as a base.

3.6. *Comparison of GA- and IP-generated job rotation solutions*

Each of the five IPs found a single job rotation solution. These solutions were compared with the maximum JSI values found across the four worker capacity groups in the solutions discovered by the GA. The maximum JSI values were compared under the four workplace scenarios discussed previously (table 9). As expected, the lowest maximum JSI values for each of the four workplace scenarios was discovered by the IP whose cost variables were based on that particular scenario (see values in bold type in table 9). The solution found by the VAR50 integer programme, however, is somewhat inflexible in that it resulted in JSI values well above the lowest maximum values found for the 75th, 90th and 100th percentile workplace scenarios. This same type of inflexibility was also seen when the solutions generated by VAR75, VAR90

Table 7. Partial job rotation solution—10 solutions in cluster.

Time period	50th percentile male	90th percentile female	25th percentile male	50th percentile female
08:00 – 09:00	Task *	Task B	Task *	Task A
09:00 – 10:00	Task *	Task A	Task *	Task B
10:00 – 11:00	Task *	Task *	Task *	Task B
11:00 – 12:00	Task *	Task A	Task *	Task B
13:00 – 14:00	Task *	Task *	Task *	Task *
14:00 – 15:00	Task C	Task A	Task B	Task D
15:00 – 16:00	Task *	Task *	Task *	Task *
16:00 – 17:00	Task *	Task *	Task *	Task *

Task *—this task assignment varied throughout the cluster.

Table 8. New job rotation solution created without the GA.

Time period	50th percentile male	90th percentile female	25th percentile male	50th percentile female
08:00 – 09:00	Task C	Task B	Task D	Task A
09:00 – 10:00	Task C	Task A	Task D	Task B
10:00 – 11:00	Task C	Task A	Task D	Task B
11:00 – 12:00	Task C	Task A	Task D	Task B
13:00 – 14:00	Task C	Task A	Task D	Task B
14:00 – 15:00	Task C	Task A	Task B	Task D
15:00 – 16:00	Task B	Task A	Task C	Task D
16:00 – 17:00	Task A	Task B	Task C	Task D
Mean JSI	1.50	1.55	1.50	1.56
SD JSI	0.098	0.038	0.13	0.102

Table 9. Maximum JSI values found in job rotation solutions discovered by GA and IP methods.

Search method	50th Percentile	75th Percentile	90th Percentile	100th Percentile
VAR50	**1.54**	1.91	2.17	2.36
VAR75	1.62	**1.82**	2.06	2.24
VAR90	1.70	1.88	**2.00**	2.16
VAR100	1.77	1.92	2.02	**2.12**
VARWTAVG	1.59	1.85	2.02	2.15
GA	1.58	1.85	2.05	2.21

Workplace scenario: all task demand variables set to percentile value.

and VAR100 were applied across the different workplace scenarios. The GA-generated solutions were robust—demonstrating maximum JSI values that were close to the optimal values across the four workplace scenarios. The IP with the weighted average cost function (VARWTAVG) also generated solutions that were robust. However, recall that the VARWTAVG IP was specifically designed for these four scenarios that it was tested on. The GA was evaluating many more solutions and therefore developing job rotation schedules that are robust across the entire range of task characteristic scenarios.

4. Discussion

This study investigated methods to incorporate safety criteria into scheduling algorithms to produce job rotation schedules that reduce the potential for injury. In this example, low-back injury was the safety concern with criteria being developed from the JSI. The proposed methods successfully found schedules that maintained productivity while controlling for exposure to musculo-skeletal strain as determined by the JSI. The IP methods found single job solutions. The GA, on the other hand, found over 400 unique solutions to the job scheduling problem. Thus, the GA was able to develop alternative solutions to the problem that could be used in further analysis. A cluster analysis of the GA solutions was shown to be effective in developing generalizable rules that could be used to develop job rotation schedules with sufficiently low JSI levels. These rules could be used in a practical sense to develop job rotation schedules without using complex computational methods.

Although in some ways successful, the application of integer programming to the job rotation problem is not without limitations. First, this particular application represents a relatively small problem (4 groups × 4 lifting tasks) but still required 128 decision variables and 68 constraints for the IP formulation. Even this relatively small problem required 10 to 80 min of CPU time to solve the resulting IP. This should be contrasted with the CPU time for the GAs, which is only a matter of minutes. Doubling the size of the problem (8 groups × 8 lifting tasks) would require 512 decision variables and the construction of 136 constraints for the IP to function properly. Preliminary testing on IPs of this size indicated that solutions could not be determined with 48 h of computation time. In addition, the variation in the solution time across different problem instances will increase as the problem size increases. Thus, with increasing problem size, IP formulation/implementation becomes too time consuming for engineers to apply in real world situations. However, the larger problems can be readily solved using the genetic algorithm with an increase in CPU time that varies roughly linearly with the increase in workers and tasks. Another limitation is the IP's tendency to fixate on a single solution to the job rotation problem. If one is interested in finding multiple solutions to the problem using IP, the programme must run through its branch-and-bound search procedure, find an optimal solution, create/add a new constraint to the formulation that makes the current solution infeasible, and then repeat the search procedure. To generate over 400 unique solutions would require the IP to execute over 400 branch-and-bound searches. The time required to conduct such a search would exceed the response time needs of the engineer for even very small problems. Finally, all of the versions of the IP required one to set the object weight, repetition rate, and horizontal distance of all lifting tasks to be static (i.e. fixed) numerical values. Law and Kelton (1991: 1 – 21) have pointed out that replacing various probability distributions with fixed values in simulation models is a dangerous practice. The authors suggested that the means *and* variances of the input probability distributions are required in simulation models to provide realistic output measures of what was actually found in the workplace. This guideline is especially important if the parameter values used in the simulation are correlated. Correlation between task demand parameters is not considered in the IP formulations of the job rotation problem.

The results of this study suggest that a genetic algorithm approach to the scheduling of work may have possible benefits to the ergonomics design process; however, a listing of the current model's limitations would also be prudent. Many of these limitations also apply to IP or other methods as well. First, the model assumes that an employee works an 8-h schedule for 5 days a week. The current model may be inadequate in situations where the days worked per week vary between individuals or the work-force is subject to relatively high job turnover rate. Second, the schedules were developed based upon a single fitness measure, the JSI, with the objective of reducing the risk of back injury. Other concerns, such as risk factors leading to upper extremity disorders (Silverstein *et al.* 1986) were not considered. The model assumes that the workers are paid the same rate regardless of which task is performed. If this assumption is not true, the GA may develop schedules that result in a marked pay imbalance between groups of workers. The model also assumes that the task demand parameters (represented by random variables) are uncorrelated. Such an assumption may result in an under-estimation (or over-estimation) of the variance associated with the JSI values associated with each job rotation. Finally, implementation of this algorithm could place additional costs on the employer by requiring periodic strength testing of their employees. Union rules and legislation such as the Americans with Disabilities Act (1990) may prevent enacting this type of assignment process. Many of the above weaknesses in the current GA model can be addressed by further development of a more complex model in future versions.

While the GA model reported here does have limitations, the method does show promise for helping to develop effective job rotation schedules. Employing a GA to evolve safe job rotation schedules has a major advantage over other optimization methods in that the planner is left with a population of good solutions to exploit in the redesign process. Given that the GA is an adaptive algorithm, the model has the ability to adjust to changes in staffing. The gender/capacity groups, used in the current model, could be replaced with the lifting capacity profiles of individuals. This change would alter the 'fitness landscape'; however, the GA process (selection, crossover, mutation, and replacement) would still drive the search to find the best solutions (i.e. most fit) within the new landscape.

Future versions of the GA model could be used to develop a simple software package for ergonomics specialists. This package would allow the specialist to choose: (1) the number of workers/tasks; (2) worker capacities; (3) the task demands; (4) the number and duration of time periods; (5) the fitness criteria for schedule evaluation; and (6) the level of solution fitness required to save the solutions specifications to a file. Non-specialists (i.e. supervisors, safety engineers) however, would rely primarily upon the population of solutions generated by the GA model. Such a population, generated by a central facility, could be incorporated into a database that would allow the non-specialist to generate a list of potential job rotation schedules that had specific attributes (i.e. certain employees performing particular operations at specified times). The list would give the non-specialist a relatively narrow venue of options, balancing specific planning requirements with a variety of scheduling options.

As demonstrated in this study, the population of good solutions can be clustered into templates (i.e. partial solutions) and rules governing the hours of task exposure can be developed. These rules/templates would provide the specialist/non-specialist with both guidance and decision flexibility when developing schedules for unforeseen circumstances (e.g. worker absences due to illness or vacation). In addition, these rules could be used in the creation of an expert system where they would act as constraints preventing the scheduler from designing a job rotation plan that placed workers at risk of back injury. The rules could also be used as a preliminary screening tool in the evaluation of similar operations with pre-existing job rotation schedules. The scheduler would measure the hours spent in specific tasks of an established schedule and then compare them to the rule sets. Those schedules that violate the rule sets may indicate a percentage of the population (male or female) at risk for potential back injury, warranting a more detailed investigation.

Future research in this area should focus on continued development of the job rotation problem as well as the GA and IP approaches to this problem. The job rotation problem could be extended to the scheduling of individual workers, who are by nature a composite of various lifting capacities (depending on the type of lifts involved). Using different probability distributions for the task demand parameters and incorporating correlation between these parameters may make for more interesting (and challenging) problems. Enhanced realism within the job rotation problem may be possible by adding constraints regarding pay balance and union-enforced restrictions concerning the duration/type of task each worker performs. A more thorough investigation of larger problems (more lifting tasks, workers, or groups of workers) may shed additional light on the performance differences between IP and GA approaches to the job rotation problem.

More sophisticated versions of both the IP and GA models may want to account for productivity-based criteria and other safety concerns such as risk factors for upper extremity injuries. In addition to the JSI, other measures of low-back injury risk may be incorporated into the objective function of the IP or fitness function of future GA models. Studying the differences in solutions found by algorithms incorporating different objective/fitness criteria would be yet another area of fruitful research. Finally, once a large number of solutions have been collected, the effectiveness of various data-mining techniques could be studied in the hope of developing systematic methods of constructing scheduling rules that enhance worker safety.

References

AYOUB, M. M. and SELAN, J. L. 1983, An ergonomics approach for the design of manual materials-handling tasks, *Human Factors*, **25,** 507–515.

DAVIS, L. (ed) 1991, *Handbook of Genetic Algorithms* (New York: Van Nostrand Reinhold).

FAUSETT, L. 1994, *Fundamentals of Neural Networks* (Englewood Cliffs, NJ: Prentice Hall).

GOLDBERG, D. E. 1989, *Genetic Algorithms in Search, Optimization, and Machine Learning* (New York: Addison Wesley).

HOLLAND, J. H. 1992, *Adaptation in Natural and Artificial Systems* (Cambridge, MA: MIT Press).

JAIN, A. K. and DUBES, R. C. 1988, *Algorithms for Clustering Data* (Englewood Cliffs, NJ: Prentice Hall).

LAW, A. M. and KELTON, W. D. 1991, *Simulation Modeling and Analysis* (New York: McGraw-Hill).

LILES, D. H. 1986, The application of the job severity index to job design for the control of manual materials handling injury, *Ergonomics*, **29,** 65 – 76.

LILES, D. H., DEIVANAYAGAM, S., AYOUB, M. M. and MAHAJAN, P. 1984, A job severity index for the evaluation and control of lifting injury, *Human Factors*, **26,** 683 – 693.

MICHALEWICZ, Z. 1994, *Genetic Algorithms+ Data Structures= Evolutionary Programs*, 2nd ed. (New York: Springer Verlag).

PINEDO, M. 1995, *Scheduling Theory, Algorithms, and Systems* (New York: Prentice Hall).

SILVERSTEIN, B. A., FINE, L. J. and ARMSTRONG, T. J. 1986, Carpal tunnel syndrome: causes and a preventative strategy, *Seminars in Occupational Medicine*, **1,** 213 – 221.

SNOOK, S. H. 1978, The design of manual handling tasks, *Ergonomics*, **12,** 963 – 985.

SPEARS, W. M. 1994, Simple subpopulation schemes, *Proceedings of the 3rd Conference on Evolutionary Computation* (River Edge, NJ: World Scientific), 296 – 307

United States Dept. of Justice 1990, *Americans With Disabilities Act* (Washington D.C.: Civil Rights Division).

Appendix 1. 0–1 integer programming formulation for job rotation problem.

Minimize : JSI

Subject to :

$$\text{(Constraint set 1)} \sum_{i=1}^{4} X_{ijt} = 1 \qquad \forall j = 1, 2, 3, 4 \;\; \forall t = 1, 2, \ldots 8$$

$$\text{(Constraint set 2)} \sum_{j=1}^{4} X_{ijt} = 1 \qquad \forall i = 1, 2, 3, 4 \;\; \forall t = 1, 2, \ldots 8$$

$$\text{(Constraint set 3)} \sum_{j=1}^{4} \sum_{t=1}^{8} c_{ijt} X_{ijt} \leq \text{JSI} \;\; \forall i = 1, 2, 3, 4$$

$X_{ijt} \in (0, 1) \qquad c_{ijt} \in \Re$ (real number)

Appendix 2. Four-step procedure for the nearest neighbour clustering algorithm.

Step 0. Let i be the job rotation schedule index counter, let x_i represent job rotation schedule i, k the cluster index counter, and n the total number of clusters.

Step 1. Set $i = 1$, $k = 1$, and $n = 1$. Assign job rotation schedule x_1 to Cluster C_1. Set $i = i + 1$.

Step 2. Find the nearest neighbour of x_i among the job rotation schedules already assigned to C_k. Let S_k denote the number of task assignments that job rotation schedule x_i has in common with its most similar job rotation schedule in C_k.

Step 3. If $S_k \geqslant 11$, then assign x_i to C_k. Otherwise, set $k = k + 1$. If $k > n$ assign x_i to cluster C_{n+1}, set $n = n + 1$, and go to **Step 4**. Else go to **Step 2**.

Step 4. If every pattern has been assigned to a cluster, stop. Otherwise, set $i = i + 1$, $k = 1$, and go to **Step 2.**