

Association Rules Discovery in Workforce Schedule Database

Jihong Yan¹ and David Nembhard²

1: Department of Industrial Engineering, Mailbox 204, Harbin Institute of Technology, China, 150001, jyan@hit.edu.cn

2: Department of Industrial & Manufacturing Engineering, Pennsylvania State University, 310 Leonhard Building, University Park, PA 16802

Abstract—Discovering hidden patterns in large sets of workforce schedules to gain insight into the potential knowledge in workforce schedules are crucial to better understanding the workforce dispatching decision making process, thereby improve workforce allocation and optimization. In this paper, a conceptual framework of the scheduling pattern discovery system is proposed. Association rule extraction methodologies are applied to explore the patterns in workforce schedules generated by a genetic algorithm (GA) based method through maximizing system throughput and machine utilization in a parallel production environment. A rule set scheduler is developed which approximates the genetic algorithm's functionality furthermore yields problem solutions by means of rules of thumb. Numerical examples illustrate that the discovered scheduling patterns can unveil the relationships existing between the characteristics of workers and machine operations, facilitate managers to enhance workforce assignment and predict system production.

Keywords: Data mining; workforce scheduling; cross-training; genetic algorithms

1. INTRODUCTION

THE decision-making requirements for scheduling multi-skilled workforce has been an intriguing research topic in recent years and the goal of achieving optimum workforce schedules which will ensure that production objectives are met is an on-going desire of most manufacturing companies. Unfortunately, because of its inherent complexity, workforce scheduling is a dominant issue that can affect whether or not production targets are achieved. In addition to the problems posed by scheduling, the productivity rate of workers may alter at any given instant in time. Therefore, dynamic changes to worker productivity rate add considerably to the already difficult scheduling.

Research on genetic algorithms (GAs) in scheduling shows that they have been successfully used in learning systems to discover permutations of sets of jobs and resources(e.g., [1]-[3]). Genetic algorithms (GAs) are stochastic search algorithms often provide fast solutions to workforce

scheduling problems. However, GAs do not demonstrate repeatability or provide an explanation of how a solution is developed [4]. Therefore, exploring the patterns among optimal or near optimal schedules derived by GA is of significance to discovering unknown knowledge existed in the schedules, and better understanding of the process. The development of data mining methodologies that is capable of intelligently discovering effective scheduling patterns is a major step beyond the scheduling methodologies that exist today. While it may be difficult to discover a general pattern that performs best in many different environments under many different operating conditions, it may be possible to discover the best pattern from a particular production environment. This rule mining process would greatly ease the task of those trying to construct and manually evaluate rules in different problem environments, both in academia and industry. Recently, data mining has been applied to extract due date assignment rules in Job-shop environment [5], but for workforce scheduling pattern recognition problem, few result has been achieved.

In this research, we develop an approach that has the ability, using a general presentation, to automatically learn effective scheduling patterns for a given production environment – the parallel workshop. The extracted patterns answer the questions such as which workers work as experts on what tasks? Which workers operate more than one task, which tasks? When should these workers rotate on tasks? By the extracted patterns, the scheduling derived by GA could be duplicated and system analysis could be conducted easily by running simulation using these rules.

2. THE WORKFORCE SCHEDULING PROBLEM

We consider workforce scheduling problem in a learning and forgetting (L/F) environment.

A. Problem Formulation

We make the following assumptions in order to illustrate the construction of a worker-task assignment model using a GA based approach.

- There are n workers available to perform the tasks
- There are m workstations operated in parallel
- Time is separated into t_p time periods, where a worker performs only one task during each time period.

- The rate of learning and forgetting is a function of how much time the worker has been performing a particular task
- Workers are heterogeneous with respect to initial productivity p_i , learning, and forgetting rates (r_i, α_i), and maximum productivity rate $k_i, i = 1, 2, \dots, K, n$
- $x_{i,j}$ x units of cumulative work for worker i performing task j
- $y_{x_{i,j}}$ Productivity rate corresponding to $x_{i,j}$

The objective function presented below in Eqn. (1) calculates which workers have to be trained for which tasks so that throughput of the machines can be maximized and meanwhile, the minimum utilization of machines can be as much as possible.

$$O = \text{Max} \left[\sum_{i=1}^n \sum_{j=1}^m x_{i,j} / c + \min(u_j) \right] \quad (1)$$

where c is a coefficient to normalize the overall system throughput, u_j ($j = 1, \dots, m$) indicates machine utilization rate, which is defined in Eqn. (2)

$$u_j = \frac{\text{machine } j \text{ scheduled hours}}{\text{machine } j \text{ available hours}} \quad (2)$$

As discussed in [6] and [7], job rotation (cross-training) may result in lower productivity as workers become less specialized, and yet increases workforce flexibility to meet the requirements of high flexibility manufacturing or services. Therefore, the objective function trades off productivity and cross-training. The first term of equation (1) determines total productivity of the system, and cross-training is ensured by maximizing minimal machine utilization rate when the number of workers is less than number of workstations. The throughput and L/F models are of the following form for one complete learning and forgetting cycle [8]. Figure 1 illustrates the L/F curve. In this case, after the amount of cumulative work reached 80, productivity rate keeps reducing because of the forgetting effect caused by absenteeism. When the worker was reassigned to the job, another round of learning started with a lower productivity rate $y_{x_{i,j}}$ that s/he left off, and $y_{x_{i,j}}$ gradually converges to the steady state productivity rate $k_{x_{i,j}}$ if worker i spent enough time on task j .

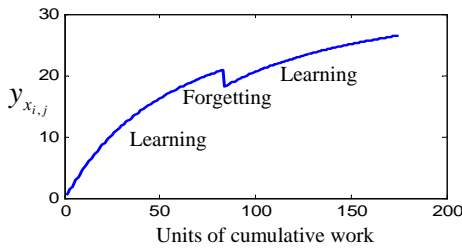


Figure 1 L/F curve

Here, we have assumed that the worker's learning function for a particular task is only related to the worker's initial productivity level for that task and how much time the worker has spent performing that task. Similarly, we assume that the worker's forgetting is only relative to the time spent not performing that task.

B. Worker parameters

More specifically, four parameters are directly related to individual L/F curve and consequently, represent the impact of worker heterogeneity on productivity, such as initial productivity rate p_i , maximum productivity rate by learning process, k_i , as well as learning/forgetting parameters r_i, α_i , which decide the L/F curve profile and represent the heterogeneity of workers, a L/F model proposed in [9] was employed in this study. Workers are grouped into three clusters using K-means clustering method according to individual's capability (IC) represented as high (IC_H), middle (IC_M), low (IC_L) capability correspondingly. The workers in same IC clusters are with similar production capability but not necessarily exactly same L/F profile.

C. Task parameters

The complexity of each task is another factor which has impacts on system productivity and schedules. In this research, task complexity (TC) is classified into three levels representing high, middle and low task complexity (TC_H, TC_M, TC_L). The impact factors of the three complexity levels are 1.2, 1.0 and 0.8 respectively. For example, a worker yields 20% more products on a TC_L task, 20% less products on a TC_H task than the productivity the individual achieves on a TC_M task within the same time length.

D. GA based schedules

To use a genetic algorithm to solve a complex problem, solutions must be encoded in a format that allows for the operations of crossover and mutation. Based on the formulation and assumptions of our problem, we proposed the Chromosome representation format shown in Figure 2:

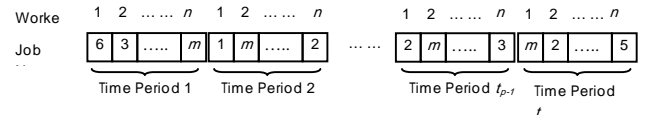


Figure 2. An example of a chromosome with n workers, m jobs, t_p simulation periods

Each worker schedule is represented as a list of working shifts, a candidate solution (chromosome) is a set of working schedules for each individual, and populations of such candidate solutions evolve as in conventional GAs. If number of workers is greater than number of jobs, then there are $n - m$ zeros in each of the periods of the chromosome, namely, $n - m$ workers are idle in the schedule.

According to the fitness of each chromosome, the one is either duplicated or abandoned by roulette wheel method [9] for crossover. Using a mutation probability of 1%, each chromosome selected for mutation had one site for mutation chosen at random.

3. CONCEPTUAL FRAMEWORK OF THE SCHEDULING PATTERN LEARNING SYSTEM

The architecture of the pattern learning system embodies that of the basic learning system model, which consists of two primary components - a reasoning mechanism and a

performance evaluator that represents the problem domain, as shown in Figure 3.

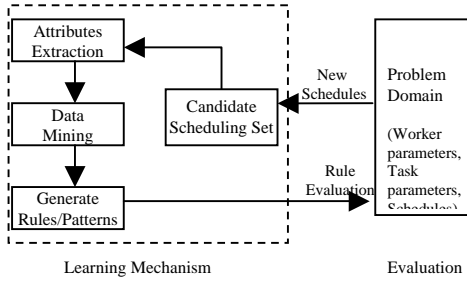


Figure 3. Conceptual model of the scheduling policy learning system

The system starts with a candidate set of schedules that are generated by GA. Schedule attributes such as number of cross-trained workers, rotation time of cross-trained workers as well as worker-task scheduling pattern are extracted from schedules to represent schedule pattern. The extracted rules are assessed through performance evaluation module. A new schedule will be added to the Candidate Scheduling Set in case the schedule pattern derived from GA differs from the extracted rules. It is possible to use such system to construct new rules adaptively as long as new schedules meet the minimum support.

4. IMPLEMENTATION

We used the association rule mining software CBA (v2.1) (Classification Based on Associations), developed at the school of computing, national University of Singapore and available for free download (<http://www.cs.uic.edu/~liub>). CBA uses the well known Apriori algorithm for finding association rules [10]. Since the dataset we use is schedule based, establishing attribution files which can meet the format requirements of CBA is the first step for mining association rules in decision schedules. The association rule extraction flowchart is shown in Figure 4.

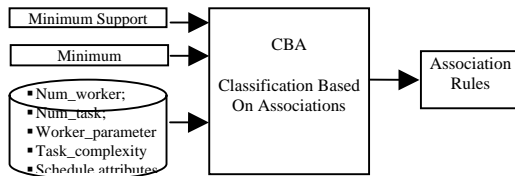


Figure 4. CBA based flowchart for association rule extraction

Rule Structure

A Rule is defined as follows:

Rule(n, m, I, IC, T, TC), which should be interpreted as:

If there are n workers operate m machines, the employee with individual capability IC is scheduled on a task with corresponding task complexity TC for T time periods, where I refers to worker ID.

For a parallel workshop, where each machine is a unit capacity processor, we consider two practical scenarios. First, number of workers is at least no less than number of tasks, in the second scenario, we investigate scheduling patterns of the

workshop condition where number of workers is less than number of tasks.

A. Rules for the cases where number of workers is no less than number of tasks

Consider a team with 6 workers and 6 workstations, the individuals' capability distribution is ($IC_H=2, IC_M=2, IC_L=2$). In this case, two workers are with high capability, two workers are in the middle level, the rest are with low production capability. The task complexity parameters of 6 workstations are: ($TC_H=2, TC_M=2, TC_L=2$).

Simulation time is 40 hours (a week). Support=2%, Confidence=20%, Number of schedules for training is 100. Three features are extracted from the 100 schedules such as number of cross-trained workers, rotation time of cross-trained workers and worker-task assignment pattern to answer the questions: which workers work as experts on what tasks? Which workers operate more than one task, which tasks? When should these workers rotate on tasks?

i. Number of cross-trained workers (NCW)

After applying CBA on the attribute of "number of cross-trained workers" extracted from 100 training schedules, results are derived shown in Table 1. NCW=0 stands for that no workers are cross-trained in the process. Correspondingly, NCW=1 refers to that there is one worker needed to be cross-trained in the process.

Table 1. Number of cross-trained workers of case 1

	NCW 0	1
workers		
$IC_H(1,2)$	100%,100%	
$IC_M(3,4)$	98%,100%	2%,100%
$IC_L(5,6)$	98%,100%	2%,100%

Table 1 could be interpreted as, when the number of workers is not less than number of tasks, the rules of NCW are: workers with higher capability (IC_H) are specialists with the support of 100% and confidence at 100%; most of the workers with middle (IC_M) or low (IC_L) capabilities are specialists with support of 98% and confidence at 100%. Only one of the middle or low capability workers in the training case is cross-trained with the support of 2%.

2) Rotation time of cross-trained workers (RTCW)

The results of the attribute of RTCW are shown as Table 2. Correspondingly, in the training set, only 1 out of the two workers of each set (IC_M, IC_L) is cross-trained with the support of 2%, the rest workers are all specialists.

Table 2. Rotation time of cross-trained worker of case 1

RTCW	None	1 st	2 nd	3 rd	4 th
Workers		10hr	10hr	10hr	10hr
$IC_H(1,2)$	100%, 100%				
$IC_M(3,4)$	98%, 100%			2%,100%	
$IC_L(5,6)$	98%,100%			2%, 100%	

The total simulation time is 40 hours, we separate the simulation period into 4 intervals, table 2 shows that the 2% cross-trained workers among the simulation are rotated at the beginning of the third 10 hours.

3) Worker-task assignment pattern (WAP)

The last question we intend to answer is the pattern of assigning workers to tasks. Table 3 shows the extracted patterns of worker-task scheduling.

Table 3. The worker-task assignment patterns of case 1

Workers \ Jobs	TC_H (1,2)	TC_M(3,4)	TC_L(5,6)
IC_H(1,2)			98%,100%
IC_M(3,4)	21.5%, 100%	74%,100%	
IC_L(5,6)	76%,100%	22%,100%	

Obviously, 98% of high capability workers are assigned to low complexity job at the confidence of 100%. 74% of middle capability workers are scheduled on middle complexity tasks, 21.5% of low capability workers are tasked on high complexity jobs. 76% of the workers with low capability are assigned on high complexity jobs, 22% of the low capability workers are assigned on middle complexity jobs.

In summary, the derived generic rule is: If $n/m \geq 1$, then no cross-training happens, assign high capability workers on low complexity jobs in the whole simulation period as many as possible; assign low capability workers on high complexity jobs, task middle complexity jobs to the workers with middle capabilities.

Specifically, in the first case, $n/m \geq 1$, there are no rotations in the schedules, and 2 high capability workers are assigned to the 2 low complexity jobs, 2 low capability workers are scheduled on the 2 high complexity tasks. This is rational in terms of objective function and type of the system. Since we consider a parallel production environment and the objective value is productivity and minimum machine utilization rate. When the workers work as a specialist, minimum machine utilization rate is 100%, high capability worker can produce more on low complexity tasks.

B. Rules for the cases where number of workers is less than number of tasks

Consider a team with 8 workers, the capability distribution is ($IC_H=2$, $IC_M=4$, $IC_L=2$). Two workers are with high capability, four workers are with middle level, the rest are with low production capability. The parameters of 12 workstations are: ($TC_H=4$, $TC_M=4$, $TC_L=4$). Similarly, the simulation time is 40 hours, number of simulations for this case is 100. Support=2%, Confidence=20%. The derived rules of each attribute are shown in tables 4-6.

Table 4. Number of cross-trained workers of case 2

workers \ NCW	0	1	2	3
IC_H(1,2)	95%,100%	5%,100%		
IC_M(3,4,5,6)			53%,100%	40%,100%
IC_L(7,8)		27%,100%	73%, 100%	

The attribute of NCW of the second case illustrates that 95% high capability workers perform as specialists, only 5% of 1 out of the two IC_H workers is cross-trained. 2 or 3 workers out of the four IC_M workers are rotated with the support of 53% and 40% respectively. 1 or 2 workers out of

the two IC_L workers are rotated with the support of 27% and 73% respectively.

Table 5. Rotation time of cross-trained workers of case 2

workers \ RTCW	None	1 st 10hr	2 nd 10hr	3 rd 10hr	4 th 10hr
IC_H(1,2)	95%,100%			5%,100%	
IC_M(3,4,5,6)	36.5%,100%			56.5%,100%	
IC_L(7,8)	13.5%, 100%			86.5%,100%	

Interestingly, the rotation time of cross-trained workers of case 2 shows as the same pattern as the first case, the workers are rotated at the beginning of the third ten hours. This result is rational since too much cross-training might have a negative impact on system performance [6], a worker is kept on a task as long as possible, only one rotation happens in this case.

The worker-task assignment pattern is similar to the results of case 1, shown as table 6, namely, assigning high capability worker to low complexity task, low capability worker to high complexity task.

Table 6. The worker-task assignment patterns of case 2

workers \ Jobs	TC_H (1,2)	TC_M(3,4)	TC_L(5,6)
IC_H(1,2)			95%,100%
IC_M(3,4,5,6)	21.5%, 100%	74%,100%	
IC_L(7,8)	76%,100%	22%,100%	

Therefore, the general rule could be summarized as: If $n/m < 1$, then cross train low capability workers on high complexity tasks for as much consecutive time as possible before rotating to another task for the rest of time, assign high capability workers on low complexity jobs as specialists as many as possible.

A combination of the extracted rules of cases 1 and 2, will cause the scheduling module to attempt to give employees some consistency in their schedules – that is, they would tend to have several days off in a row, tend to work in the same task, and tend to work on the same task throughout the scheduling period. This is a reasonable result since learning/forgetting effects have been considered during scheduling process, consecutive time spent on the same task might result in high productive rate which contributes to the objective value.

5. EVALUATION

In an effort to determine the effectiveness of the extracted rules, rigorous testing is performed by applying the rules to different scheduling problems in parallel production environment. The aim here is to create a variety of different production scenarios in which discovered scheduling patterns can occur and to evaluate the quality of the rules obtained by data mining method.

The quality of the final rules learned during the training phase is tested by evaluating its predictive power on the test set of instances. The test set is a set of unseen instances to which the best discovered rules is applied. It is comprised of a different set of instances representative of the same desired conditions represented by those in the training set. The

application of the rules to this set of problems gauges the generality and the applicability of the learned rules as a product of the training set. Define Mean_O to evaluate similarity of objective value derived from extracted rules and GA:

$$Mean_O = \sum_{l=1}^{sn} \left(1 - \frac{|O_{rule}(l) - O_{GA}(l)|}{O_{GA}(l)} \right) / sn \quad (3)$$

Where O_{rule} is the objective value derived from extracted rules; O_{GA} is the objective value obtained from GA approach, sn is the number of simulation cases.

Two cases were designed to verify the rule for scenario 1. Test case 1 has 4 workers, 4 tasks with the parameters (IC_H=1, IC_M=2, IC_L=1, TC_H=1, TC_M=2, TC_L=1). The other test case is 10 workers, 8 tasks where (IC_H=5, IC_M=3, IC_L=2, TC_H=2, TC_M=2, TC_L=4). 100 simulation cases were made for each scenario. The comparison results are shown in table 7.

Table 7 Evaluation on the rule of scenario 1:
Number of workers no less than number of tasks

	Mean_O	Std.
Test case 1	99.2%	0.93%
Test case 2	98.53%	0.95%

Table 8. Worker-task assignments

Jobs	TC_H	TC_M	TC_L
workers			
IC_H			100%
IC_M		100%	
IC_L	100%		

Table 8 illustrates that 100% assignment patterns of the established rules match the results from GA. The difference on Mean_O is caused by the heterogeneity of workforce, namely, even though two workers are selected from same cluster (for example, IC_H), the productivity might be a little different within same time length due to the slight differences on worker parameters.

Similarly, two cases were designed to verify the rule for scenario 2. Test case 3 is 2 workers, 4 tasks with the parameters (IC_H=1, IC_M=0, IC_L=1, TC_H=1, TC_M=1, TC_L=2). The test case 4 has 15 workers, 20 tasks where (IC_H=2, IC_M=7, IC_L=6, TC_H=7, TC_M=8, TC_L=5). 100 simulation cases were made for each scenario. The comparison results are shown in table 9.

Table 9 Evaluation on the rule of scenario 2:
Number of workers less than number of tasks

	Mean_O	Std.
Test case 3	99.06%	0.83%
Test case 4	93.25%	2.84%

The rules fit first scenario can give better results than the rules for the second scenario. Since for the second scenario, cross-training happens in the schedule, which adds much complexity in the optimization process, the rules might not be able to present the exact pattern of GA. Rules are of significance for workforce scheduling, especially for large scale scheduling problem, which can deliver prediction on productivity directly in no time, while it might be much more difficult for GA method in terms of computational time.

6. CONCLUSIONS

A significant achievement of this effort is the development of a rule syntax which is generic and powerful. Generally, many types of workforce scheduling procedures can be incorporated into the system via the rule format as it stands. The discovered patterns in workforce schedules provide insights of the manpower assignment process, a predictor and simulator for productivity and bottleneck analysis, which could even help the managers to detect the flaws in workforce allocation, eventually improve workforce optimization and cross-training policies.

Future work would be applying intelligent data mining methods for association rules discovery instead of CBA, in that case, the input format could be defined according to requirements of specific problem. The rule discovering process is applicable for serial production line, the discovered rules will demonstrate the advantages on bottleneck simulation analysis under serial environment. The extracted rules will be different to different objective functions, if larger weighting is given to machine utilization rate, then more rotation/cross-training would be observed, if greater weighting is given to productivity, then some machine might be idle when number of workers is less than number of tasks. In addition, cross-training would be obtained when workforce flexibility index is considered in objective function even though number of workers is greater than number of tasks. Therefore, different objective functions need to be investigated in the future.

REFERENCES

- [1] Dorndorf, U. and Pesch, E., 1995, "Evolution Based Learning in a Job Shop Scheduling Environment", *Computers and Operations Research*, 22,25-40.
- [2] Mattfeld, D.C., 1996, *Evolutionary Search and the Job Shop: Investigations on Genetic Algorithms for Production Scheduling*, Physica-Verlag, Heidelberg, Germany.
- [3] ElMaraghy, H., Patel, V. and Ben Abdallah, I., 2000, "Scheduling of Manufacturing Systems under Dual-Resource Constraints Using Genetic Algorithms", *Journal of Manufacturing Systems*, 19, 186-201.
- [4] Koonce D.A. and Tsai, S.C., "Using data mining to find patterns in genetic algorithm solutions to a job shop schedule," *Computers & Industrial Engineering*, 38, pp. 361-374, 2000.
- [5] Sha D.Y. and Liu C.H., "Using Data Mining for Due Date Assignment in a Dynamic Job Shop Environment", *Int. J. Adv. Manuf. Technol.*, 25: 1164-1174, 2005.
- [6] Nembhard, D.A. and Norman B.A., "Worker efficiency and responsiveness in cross-trained teams," Technical Report, University of Wisconsin-Madison, 2002.
- [7] Nembhard, D.A. and Norman, B.A., "Workforce scheduling with learning and forgetting effects," Presentation, INFORMS, Miami. <http://www.ie.psu.edu/researchlabs/wel/Reprints/INFORMS-Miami-David.pdf>, 2001.
- [8] Nembhard D. A. and Uzumeri M. V. , "Experimental learning and forgetting for manual and cognitive tasks," *International Journal of Industrial Ergonomics*, 25(4), pp. 315-326, 2000.
- [9] Giordana A. and Neri F., "Search-intensive concept induction," *Evolutionary Computation*, 3: pp. 375-416, 1996.
- [10] Agrawal, R.T. and Srikant, "Fast algorithms for mining association rules," *Proc. of the 20th Int'l Conference on Very Large Databases*, Santiago, Chile, Sept., 1994.