PERGAMON

Computers & Industrial Engineering 39 (2001) 159–171

computers & industrial engineering

www.elsevier.com/locate/dsw

Project surface regions: a decision support methodology for multitasking workers assignment in JIT systems

H.-C. Horng[a,*], J.K. Cochran[b]

[a]Department of Industrial Engineering and Management, Chaoyang University of Technology, Taichung, Taiwan
[b]Department of Industrial Engineering, Arizona State University, Tempe, AZ 85287, USA

Abstract

This research develops a decision support methodology called project surface regions (PSR) via the use of discrete-event simulation and response surface methodology (RSM). It is used to assist production managers in determining the appropriate number of multitasking workers and the corresponding dynamic dispatching rule-pair when a JIT system's behavior changes. A set of dynamic dispatching rule-pairs, including a new invented rule-pair, is considered to demonstrate the application of PSR. © 2001 Elsevier Science Ltd. All rights reserved.

Keywords: Project surface regions; Response surface methodology; Discrete-event simulation

1. Introduction

The concept of multitasking workers plays a major role in the success of Just-in-Time (JIT) implementation. It has been pointed out that one key element of a JIT system is its ability to use multitasking workers to react to unbalanced work loads where the bottleneck location changes from period to period (Hall, 1983; Treleven, 1987). A standard set of dynamic dispatching rule-pairs has been evaluated for push-type production systems in the literature (Treleven, 1987; Park & Study, 1990; Malhortra & Kher, 1994). However, there are no studies on the evaluation of dynamic dispatching rule-pairs of multitasking workers in JIT production systems. In this paper, we present a decision support methodology called project surface regions (PSR), which utilizes discrete-event simulation and response surface methodology (RSM), and how PSR can be used to assist production managers in determining the appropriate number of multitasking workers and the corresponding dynamic dispatching rule-pair when a JIT system's behavior changes.

A set of rule-pairs has been considered for the dynamic dispatching of multitasking workers and

* Corresponding author.

applied to dual resource constrained (DRC) job-shop environments (Treleven, 1987; Park & Study, 1990; Malhortra & Kher, 1994). In a DRC job-shop, the number of machines is typically less than the number of workers, who may be cross-trained to perform a variety of tasks (Treleven, 1989). This is the same as the situation of multitasking workers in JIT assembly systems, where the performance of each station is increased as the number of worker assigned to that station increases. Morris and Tersine (1994) use simulation to compare three different worker assignment rules in DRC environment so that a better layout can be obtained. They found out that process layout has better performance (in WIPs and throughput) than cellular layout no matter which rule is selected. Shenoy and Bhadury (1993), on the other hand, construct a MRSRP model using simulation language SLAM II to help the maintenance planner to schedule maintenance resources effectively and ensure improvement in the availability of the machines.

Each dynamic dispatching rule-pair is composed of two rules from two separate categories, ‘when’ and ‘where’. In the ‘when’ category, two rules have been considered in the literature (Treleven, 1987; Nelson, 1967; Fryer, 1973; Gunther, 1979, 1981). The first rule, *the decentralized rule*, allows multitasking workers to transfer to another station only when they become idle due to an empty queue at the current station. This rule has been recommended for DRC environments with transfer delays (Treleven, 1987). Under the second rule, *the centralized rule,* multitasking workers are allowed to transfer to another station after the completion of every job. As compared to the previous rule, the efficiency of this rule is significantly affected by transfer delays. Therefore, it can be inappropriate to apply this rule to DRC shops with considerable transfer costs.

There are many different rules that have been considered in the literature under the ‘where’ category. The only one this study considered is the ‘first arrived in the system, first served’ (FASFS): this rule assigns the multitasking worker to a station whose queue contains a job that has been in the station for the longest time. It is shown that this rule works well in DRC shops with homogeneous workers and instant transfer (Weeks & Fryer, 1976). However, the performance of this rule is unknown in JIT production systems with multitasking workers.

These rule-pairs can also be applied to the assignment of multitasking workers in JIT production systems. However, these rule-pairs do not take advantage of the special characteristic of JIT production environments. JIT production systems use kanban as a tool to realize pull-type production and control total WIPs in the entire production system. If the total number of production kanban (PK) is k, then any station with $k - 1$ PK waiting to be fulfilled is a bottleneck station at that moment. Under the circumstances, the machine or the operator is performing task on one of the jobs while a full load of parts in the input part buffer is waiting. This bottleneck situation causes two problems. First, the succeeding station will be idle since there will be no parts to process. Second, the station before the bottleneck station is idle because of the shortage of PK (no authorization to process incoming parts). This situation never occurs in non-JIT production systems unless there are limitations on the size of inter-station buffers.

This rule falls into the category of a ‘when’ rule and is called *the bottleneck rule*. The bottleneck rule allows the transfer to be made once the number of unsatisfied PK of the station drops to $k - 2$ after a job is completed, that is, the station is no longer a bottleneck to the system and it is safe to transfer the multitasking worker to other needed stations.

In summary, three rule-pairs are considered in this study. R_1 allows the transfer of the multitasking worker when the current station has an empty queue. R_2 allows transfer of the multitasking worker when he/she completes the current job. R_3 allows transfer of the multitasking worker when the current station

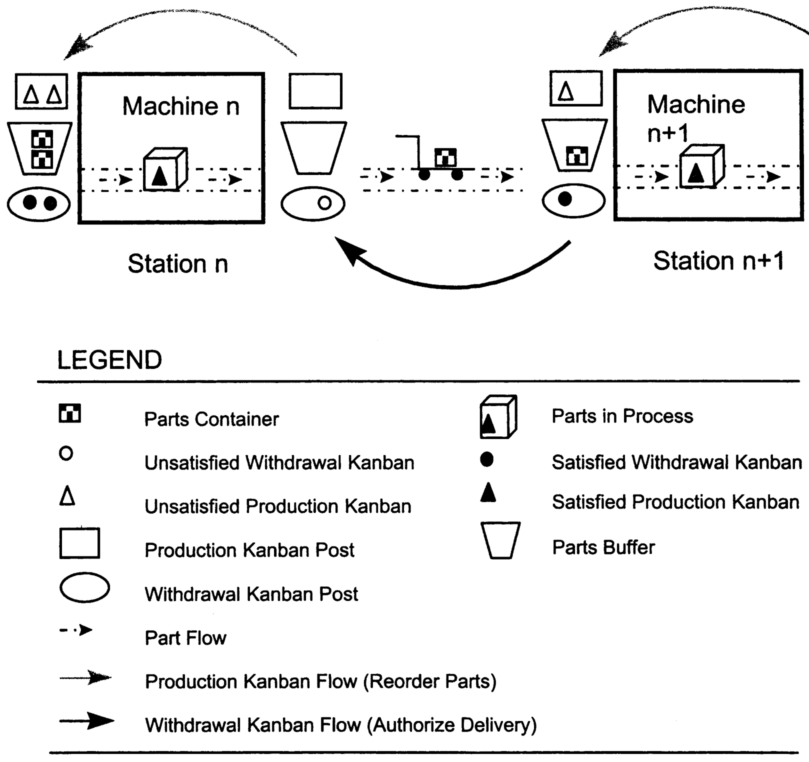

Machine n
Machine n+1
Station n
Station n+1
LEGEND
Parts Container
Unsatisfied Withdrawal Kanban
Unsatisfied Production Kanban
Production Kanban Post
Withdrawal Kanban Post
Part Flow
Production Kanban Flow (Reorder Parts)
Withdrawal Kanban Flow (Authorize Delivery)
Parts in Process
Satisfied Withdrawal Kanban
Satisfied Production Kanban
Parts Buffer

Fig. 1. The dual Kanban-controlled system.

is no longer a bottleneck station. All three rule-pairs assign the multitasking worker to a station whose queue contains a job that has been in the station for the longest time.

2. Dual-kanban controlled JIT production systems

Two types of kanban-controlled mechanism, single and dual, are used in JIT production systems. In the dual kanban-controlled production system, two different types of kanban called the PK and the withdrawal kanban (WK) are utilized. PK is used to authorize the production of a certain quantity of a specific type of part in one station. WK is then used to authorize the transportation of a certain quantity of a specific type of part from one station to another. The pull concept of a JIT production system is thus realized via the implementation of these two different kanbans. On the other hand, in a single kanban-controlled production system, each kanban serves the purpose of authorizing both production and transportation of certain parts. One can easily construct a simulator for the single kanban production ystem just by making slight modifications to the dual kanban-controlled simulation models. This study does not attempt to compare the differences between single- and dual-kanban controlled production systems, but to demonstrate the application of PSR. Therefore, we only focus on the

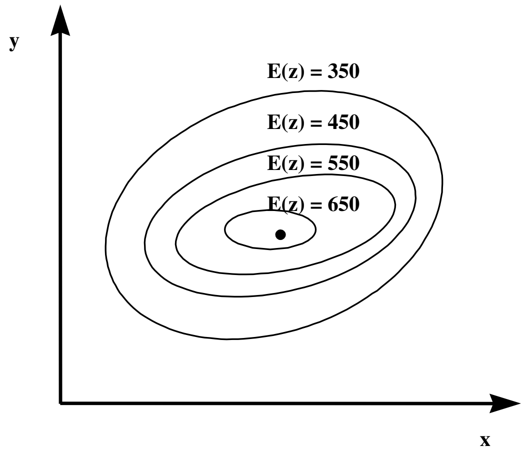


Fig. 2. A typical response surface.

dual kanban-controlled production system. The most common form of a dual kanban-controlled production system is described by (Monden, 1983; Schonberger, 1982).

Fig. 1 illustrates the behavior of these two types of kanban. In the figure, parts are produced in batches in stations n and $n + 1$. Each station contains one machine, two part buffers (PB), two production kanban posts (PKP), and two withdrawal kanban posts (WKP). Only when there is an unsatisfied (the amount specified on the kanban have not been fulfilled) production kanban in the input PKP, can the parts in the container of the input PB be processed by the machine. The container with processed parts in the output PB of station n must wait for an unsatisfied WK from the next station to be delivered to the input PB of the next station.

PK is transferred between the input PKP and the output PKP of the same station. For example, the unsatisfied PK of station n is put on the input PKP to wait for a batch of parts (a parts container). The PK becomes satisfied when machine n becomes idle and a parts container shows up in the input PB. The satisfied PK then follows the parts container during the machining process. After the machining process is finished, the satisfied PK is then put on the output PKP to wait for the next reorder point. At the reorder point, all PK on the output PKP is transferred to the input PKP and becomes unsatisfied.

WK, on the other hand, is transferred between stations. For example, the unsatisfied WK is put on the output WKP of station n. When a parts container shows up in the output PB of station n after the machining process, the WK is satisfied and can be delivered with the parts container to station $n + 1$. The satisfied WK stays on the input WKP of station $n + 1$ as long as the associated parts container has not been processed by machine $n + 1$. Once the parts container is put into processing, the associated WK is then transferred to the output WKP of station n and becomes unsatisfied again.

3. RSM and simulation

When the objective of a simulation study is to optimize a system's performance, RSM is often employed (Montgomery & Bettencourt, 1977; Gaston & Walton, 1994; Donohue, Houck & Myers, 1995). The major benefit of using RSM in a simulation study is the significant reduction in the number of simulation runs needed. In this study, we are dealing with the optimization of a system's throughput regarding to different dynamic dispatching rule-pairs. Since there are three rule-pairs to be compared based on their performance (*system's throughput*) over a certain range of independent variables (for example, *utilization factor and number of multitasking worker*), large number of simulation runs are needed. The application of RSM can also accurately generate a good approximation of the true response surface of the system's throughput. Thus, the combination of simulation and RSM is suitable for this study.

For each rule-pair in this study, RSM can be used to generate the response surface function for the system's throughput over a certain range of independent variables. The observed dependent variable or response *throughput* (z) is a function of *utilization factor* (y) and *number of kanbans* (x), i.e.

$$z = f(x, y) + \varepsilon,$$

where ε is a random error. If the expected response is $E(z)$, then the surface $\varepsilon(z) = f(x, y)$ is called a response surface.

Fig. 2 is graphical representation of a typical response surface. As the true response surface is usually unknown, finding a suitable approximation becomes important to the experimenter. Polynomials are the most widely used type of approximating function. Among all polynomials, the first-order model is often employed with the method of steepest ascent at the initial stages of a response surface study to find the region where the optimum is located. Once the region is found, a second-order polynomial is then applied to provide a more satisfactory approximation to the true response surface (Montgomery & Bettencourt, 1977; Montgomery, 1991). Since the interest region has been predetermined in this study, the second-order polynomial becomes the basic approximation for this study.

Given a set of N responses, $Y_u(u = 1, \ldots, N)$, the mathematical expression (or equation) for the second-order model is

$$y_u = \beta_{00} + \sum_{i=1}^{k} \beta_i x_{iu} + \sum_{i=1}^{k} \beta_{ii} x_{iu}^2 + \sum_{i<j}\sum \beta_{ij} x_{iu} x_{ju} + \varepsilon_u$$

or, in matrix form,

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

where

$$\mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1k} & x_{11}^2 & \cdots & x_{1k}^2 & x_{11}x_{12} & \cdots & x_{1(k-1)}x_{1k} \\ 1 & x_{21} & \cdots & x_{2k} & x_{21}^2 & \cdots & x_{2k}^2 & x_{21}x_{22} & \cdots & x_{2(k-1)}x_{2k} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 1 & x_{N1} & \cdots & x_{Nk} & x_{N1}^2 & \cdots & x_{Nk}^2 & x_{N1}x_{N2} & \cdots & x_{N(k-1)}x_{Nk} \end{bmatrix}, \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_N \end{bmatrix},$$

and $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_k, \beta_{11}, \beta_{22}, \ldots, \beta_{kk}, \beta_{12}, \beta_{13}, \ldots, \beta_{k-1,k})'$.

In the equation above, $\mathbf{X}$ is a

$$\left(N \times \left(1 + 2k + \binom{k}{2}\right)\right)$$

matrix of the level of the k independent variables ($x_{ju}, j = 1, \ldots, k$), Y is a ($N \times 1$) vector of the responses (y_u, $u = 1, \ldots, N$), $\boldsymbol{\beta}$ is a

$$\left(\left(1 + 2k + \binom{k}{2}\right) \times 1\right)$$

vector of the regression coefficients, and $\boldsymbol{\varepsilon}$ is an ($N \times 1$) vector of random errors (ε_u, $u = 1, \ldots, N$). The least squares estimator of $\boldsymbol{\beta}$ is

$$\check{\boldsymbol{\beta}} = \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{X}$$

In general, the more data gathered the better the prediction. However, more data means more experimental runs are needed and thus, it is more expensive to conduct the experiment. In order to generate a second-order model efficiently, a special design is necessary. The purpose of an RSM design is to give a guideline for gathering enough data about the true response surface while keeping the number of runs to a minimum. However, it is helpful to have a design criterion so that an appropriate design can be selected from the many designs available. For the first-order model, *orthogonality* (the cross products of the columns of the $\mathbf{X}$ matrix sum to zero) is an optimal design property as it minimizes the variance of the regression coefficients. For the second-order model, *rotability* (measuring how much the variance of the predicted response $\mathbf{y}$ at some point $\mathbf{x}$ is affected by the direction of the point from the design center) becomes important since it requires the variance of a predicted value remain constant at points that are equidistant from the design center. Therefore, a 100% rotatable design provides equal precision of estimation in all directions, which is very important for locating the optimum. Note that any first-order orthogonal design is rotatable.

Two RSM designs are important to the study, the *central composite design* (CCD) and the *face-centered composite design* (FCC). The PSR procedure that will be presented later only allows at maximum two independent variables (k). Under that circumstance, the CCD is 100% rotatable with $\alpha = 1.414$, that is, the CCD will have four axial points as well as four factorial points. In addition, with four center points, this design is also orthogonal and of uniform-precision. A uniform-precision design affords more protection against bias in the regression coefficients than others do (Montgomery, 1991).

The FCC is chosen when one of the independent variables (for example, *number of multitasking workers*) is a discrete variable. The rotability of a FCC design is 93.08% for $k = 2$ and reduces significantly as k increases. This is considered a disadvantage. Therefore, it is only used when it is difficult to change factor levels.

4. Verification of SimJIT via analytical model

The SimJIT simulator is an object-oriented simulation software developed by the author (Horng-Chyi Horng, 1996) for the purpose of simulating dual-kanban controlled JIT systems with multitasking workers. Since there is no analytical model available for the verification of the SimJIT simulator with multitasking worker and their dynamic dispatching rule-pairs, the only practical approach in this section to verify the SimJIT simulator is when there is no multitasking worker involved.

A three-station dual-kanban controlled system with the number of kanban equal to three (for both production and withdrawal kanban) and processing time equal to Exponential(1) has been studied

Table 1
Comparison of SimJIT and Markov-numerical results

	SimJIT results	Markov-numerical results	Error (%)	*t*-value
Utilization				
Online operator 1	0.83081	0.83076	0.0060	0.0432
Onlineoperator 2	0.83043	0.83076	0.0397	−0.3233
Onlineoperator 3	0.82994	0.83076	0.0987	−0.8652
Average number in queue				
Input part buffer 1	1.19392	1.19485	0.0778	−0.1367
Input part buffer 2	2.15168	2.15130	0.0177	0.0785
Input part buffer 3	1.60698	1.60986	0.1789	−0.5737
Output part buffer 1	0.97529	0.97439	0.0924	0.1019
Output part buffer 2	0.55374	0.55379	0.0090	−0.0415
Input production Kanban buffer 2	1.61710	1.61545	0.1021	0.4785
Input production Kanban buffer 3	2.17008	2.16924	0.0387	0.6749
Output withdrawal Kanban buffer 1	0.84834	0.84870	0.0424	−0.0784
Output withdrawal Kanban buffer 2	1.39304	1.39014	0.2086	0.7984

(Berkley, 1991) via the Markov-chain approach. Table 1 is the comparison of the simulation results from the SimJIT simulator and the numerical results from the Markov chain. The maximum error calculated is only 0.2% and the maximum absolute *t*-value for hypothesis testing is 0.8652. This means that the SimJIT simulator produces almost the same results as the Markov-numerical approach published in the literature for systems where no multitasking workers are involved. The development of an analytical model of JIT production systems with the capacity of handling the dynamic dispatching of multitasking workers is a potential future research topic.

5. The PSR procedure

5.1. Description of the hypothetical system

In order to demonstrate the PSR procedure, a case study is selected. The nature of the selected hypothetical system is the same as the system illustrated in Fig. 1. There are 10 stations in series in this system. Parts are processed and delivered in terms of a container, that is, all parts in a container are processed and delivered together. Several assumptions are added as below:

1. Only one type of part is processed at all stations.
2. The incoming material are always available when a new order is placed, which means there is no delay on the incoming material for the first station.
3. The processing times of all parts in a container at all stations are assumed to be the same, which means the production line is perfectly balanced.

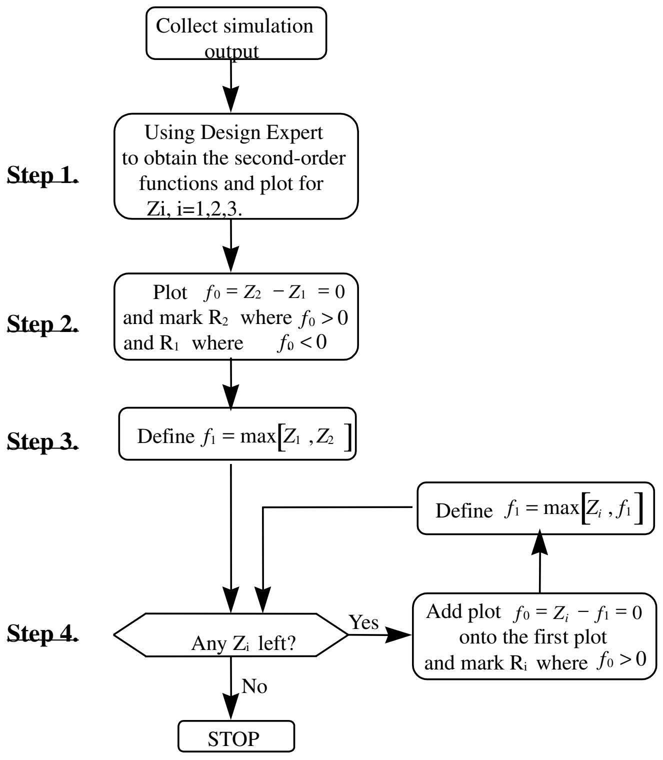

Collect simulation output
Step 1.
Using Design Expert to obtain the second-order functions and plot for Zi, i=1,2,3.
Step 2.
Plot $f_0 = Z_2 - Z_1 = 0$ and mark R_2 where $f_0 > 0$ and R_1 where $f_0 < 0$
Step 3.
Define $f_1 = \max[Z_1, Z_2]$
Define $f_1 = \max[Z_i, f_1]$
Step 4.
Any Z_i left?
Yes
Add plot $f_0 = Z_i - f_1 = 0$ onto the first plot and mark R_i where $f_0 > 0$
No
STOP

Fig. 3. The flow chart of the procedure.

4. The finished goods are delivered to customers immediately, which means an output parts buffer is unnecessary for the last station.

The reorder point for the PK at every station is one, that is, once the number of PK in the input PK buffer is equal to or less than one, all PK in the output PK buffer are issued (put back into the input PK buffer) to reorder parts from the previous station. There are originally five kanbans at each station and the processing time at each station is exponentially distributed with a mean service time of five.

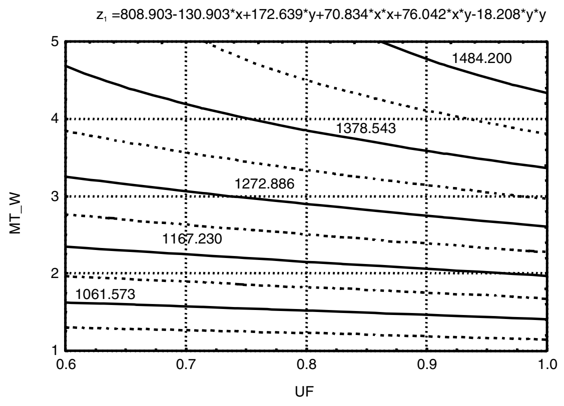
Two independent variables, utilization factor and number of multitasking workers, are considered. The simulation is run for 10,000 h. With several pilot runs, we found the system reaches steady-state around 1500 h, therefore the truncate point (when warm-up period ends and previous statistics are discarded) is set to 2000 h. FCC is the appropriate RSM design for this analysis since the number of multitasking workers is a discrete variable. Three replications of the simulation are conducted. With three rule-pairs and 12 runs for FCC, the total simulation runs are $3 \times 12 \times 3 = 108$ for each configuration. The computer system used in this study is a Pentium 100 MHz with 24M RAM running Windows 95. The CPU time required to complete 108 simulation runs ranges from 20 to 30 min.

5.2. Illustration of the procedure

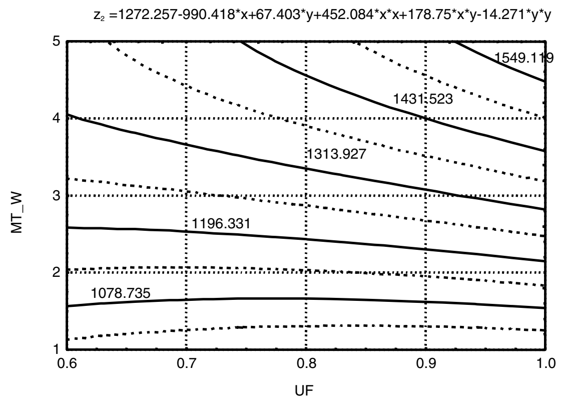
The procedure for generating plots of the project surface regions is shown in Fig. 3. The major concern

Table 2
Simulation output for the case study (MTW, multitasking workers; UF, utilization factor; Throughput, average number of completed batches of parts)

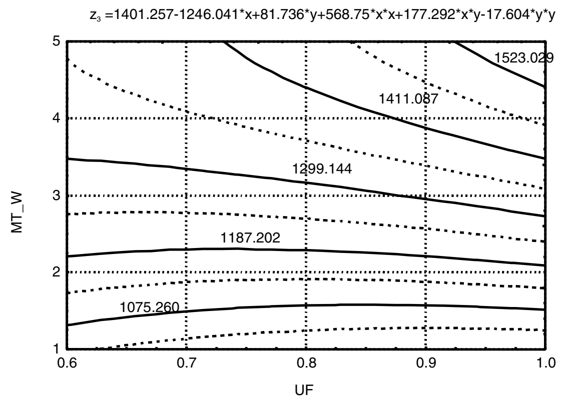
Number of MTWs (coded)	UF (coded)	Number of MTWs (actual)	UF (actual)	Throughput (R_1)	Throughput (R_2)	Throughput (R_3)
−1	−1	1	0.55	996	985.33	984.67
−1	0	1	0.75	970	972.67	970.67
−1	1	1	0.95	934	974.33	1013.33
1	−1	5	0.55	1571	1667	1620
1	0	5	0.75	1432.33	1392.67	1397.67
1	1	5	0.95	1387.33	1370	1365.67
0	−1	3	0.55	1280.33	1265.33	1281.67
0	0	3	0.75	1305.67	1293.67	1282.67
0	0	3	0.75	1306	1272.33	1313.33
0	0	3	0.75	1293.33	1261	1297
0	0	3	0.75	1263.33	1325.67	1282
0	1	3	0.95	1273.33	1250.33	1273



3D Contour Plot (Throughput of R_1)



3D Contour Plot (Throughput of R_2)



3D Contour Plot (Throughput of R_3)

Fig. 4. Second-order functions and contour plots of the three rule-pairs.

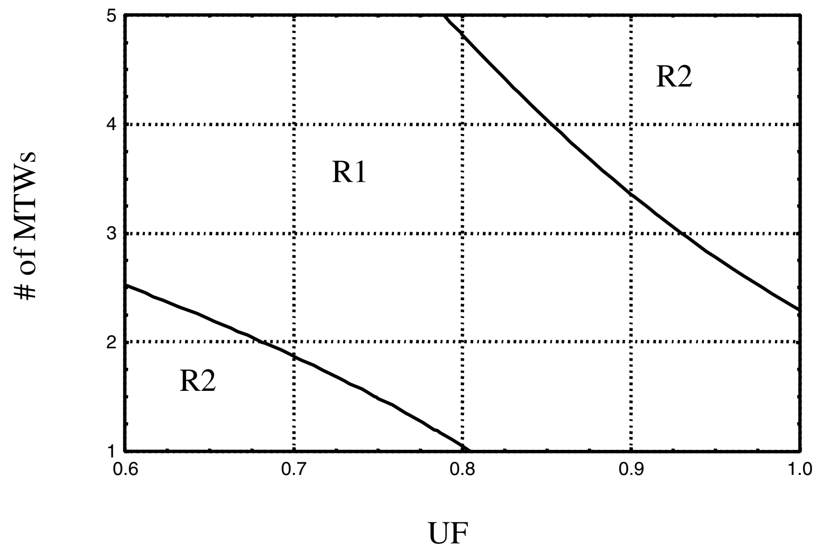


Fig. 5. Project surface regions plot of R_1 and R_2.

of this example is assumed to be the system throughput with respect to the utilization factor (from 0.6 to 1.0) and the number of multitasking workers (from 1 to 5). The procedure for generating the project surface regions of R_1, R_2, and R_3 is:

STEP 0.1 Construct a simulation model for each rule.

STEP 0.2 Use RSM to generate the total number of runs required and initial conditions for each

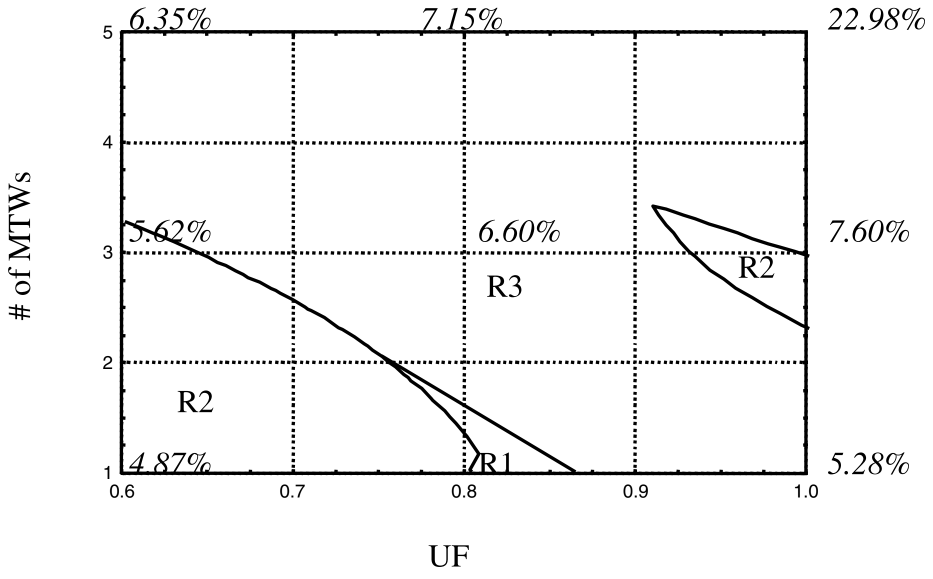


Fig. 6. The PSR plot of the case study.

model. First two columns in Table 2 show an FCC design. The corresponding value of two variables, the number of multitasking workers and the utilization factor are shown in column 3 and 4, respectively.
STEP 0.3 Run the simulation model according to the initial conditions and required number of runs specified in STEP 0.2. Last three columns in Table 2 list the SimJIT simulation output for each rule-pair.

STEP 1. Use response surface software such as Design Expert™ to generate a second-order function for each rule. The second-order function would look like:

$$Z_i = b_0 + b_1x + b_2y + b_{11}x^2 + b_{12}xy + b_{22}y^2,$$

where Z_i is the throughput using rule R_i and x as well as y are the utilization factor and the number of multitasking workers, respectively. Represented in Fig. 4 are second-order functions and contour plots.
STEP 2. Let $f_0 = Z_{12} - Z_{11}$, plot $f_0 = 0$, and mark R_1 on the region where $f_0 < 0$ and R_2 on the region where $f_0 > 0$. Fig. 5 illustrates the plot after STEP 2.
STEP 3. Define $f_1 = \max[Z_{11}, Z_{12}]$.
STEP 4. Perform the following recursive procedure:
  STEP 4.1. STOP if there is no second-order function left.
  STEP 4.2. Plot $f_0 = z - f_1$ on the plot illustrated in Fig. 5 (superimpose), where z_i is one of the remaining second-order functions. Mark R_i on the region where $f_0 > 0$. In the example, mark R_3.
  STEP 4.3. Assign $f_1 = \max[z, f_1]$ and go to STEP 4.1.

The final plot for the example after STEP 4 is illustrated in Fig. 6. This is a project surface region plot. The percentage of gain on the system throughput over no multitasking workers is shown in the figure at some points. The percentage of gain on the system throughput increases when UF is high and more multitasking workers are involved. This is because the system is exponentially distributed in both processing time and order arrival, and when UF is high, bottleneck problems are going to happen more often. By increasing number of multitasking workers, the likelihood of bottleneck problems is reduced and thus increasing number of multitasking workers increases the system throughput. However, increasing number of multitasking workers does not always produce reasonable results. In Fig. 6, when UF is low (0.6), increasing number of multitasking workers can only increase the system throughput by less than 1%. Whether to add another multitasking worker is a decision of the production managers. All in all, the production manager can use this plot to determine which rule to use and how many multi-tasking workers are needed once he/she has the information about the number of multitasking workers and the utilization factor.

6. Summary and conclusion

A decision support tool called PSR is developed for the purpose of assisting production managers in deciding whether to add multitasking workers to the JIT production line and which dynamic dispatching rule-pairs these multitasking workers should follow. By integrating discrete-event simulation and RSM, the PSR provides insight about the system throughput when system behavior changes and multitasking workers are involved.

This study also supports that by adding multitasking workers, the system throughput can be improved significantly. The impact of multitasking workers increases when UF is high and the number of kanban assigned to each station is not enough to offset the variation in processing time and order arrives more often. In some situations, adding another multitasking worker cannot improve the system throughput significantly, especially when UF is low and fewer bottleneck problems occurs. Therefore, it is up to the production manager whether to add another multitasking worker.

The new invented bottleneck rule appears to have better performance on the system throughput in most situations. It incorporates the special characteristic of JIT production systems into the worker assignment rule-pair. The main design purpose of this bottleneck rule is to prevent stations in JIT production systems from becoming a bottleneck station. Thus, it greatly improves the system throughput and works better in the JIT production environment.

References

Berkley, B. J. (1991). Tandem queues and kanban-controlled lines. *International Journal of Production Research*, *29* (10), 2057–2081.

Donohue, J. M., Houck, E. C., & Myers, R. H. (1995). Simulation designs for the estimation of quadratic response surface gradients in the presence of model misspecification. *Management Science*, *41* (2), 244–262.

Fryer, J. S. (1973). Operating policies in multiechelon dual-constrained job shops. *Management Science*, *19* (9), 1001–1012.

Gaston, G. J., & Walton, A. J. (1994). The integration of simulation and response surface methodology for the optimization of IC processes. *IEEE Transactions on Semiconductor Manufacturing*, *7* (1), 22–33.

Gunther, R. E. (1979). Server transfer in a dual resource constrained parallel queueing system. *Management Science*, *25* (12), 1245–1257.

Gunther, R. E. (1981). Dual-response parallel queues with server transfer and information access delays. *Decision Sciences*, *12* (1), 97–111.

Hall, W. R. (1983). *Zero inventory*, Homowood, IL: Dow Jones-Irwin.

H. -C. Horng. (1996). *Optimizing the multitasking of workers in Just-in-Time systems.* PhD dissertation, Department of Industrial and Management Systems Engineering, Arizona State University, Tempe.

Malhortra, M. K., & Kher, H. V. (1994). An evaluation of worker assignment policies in dual resource-constrained job shops with heterogeneous resources and worker transfer delays. *International Journal of Production Research*, *32* (5), 1087–1103.

Monden, Y. (1983). *Toyota production system*, Norcross: Industrial Engineering and Management Press.

Montgomery, D. C., & Bettencourt, V. M. (1977). Multiple response surface methods in computer simulation. *Simulation*, *October*, 113–121.

Montgomery, D. C. (1991). *Design and analysis of experiments* (3rd ed.), New York: Wiley.

Morris, J. S., & Tersine, R. J. (1994). A simulation comparison of process and cellular layouts in a dual resource constrained environment. *Computers and Industrial Engineering*, *26* (4), 733–741.

Nelson, R. T. (1967). Labour and machine limited production systems. *Management Science*, *13* (9), 648–671.

Park, P. S. (1990). A study of labor assignment rules with bottlenecks. *Omega*, *18* (3), 247–257.

Schonberger, R. J. (1982). *Japanese manufacturing techniques*, New York: Free Press.

Shenoy, D., & Bhadury, B. (1993). MRSRP — a tool for manpower resources and spares requirements planning. *Computers and Industrial Engineering*, *24* (3), 421–430.

Treleven, M. (1987). The timing of labor transfers in dual resource-constrained systems: push vs. pull rules. *Decision Sciences*, *18*, 73–88.

Treleven, M. (1989). A review of dual resource constrained system research. *IIE Transactions*, *21* (3), 279–287.

Weeks, J. K., & Fryer, J. S. (1976). A simulation study of operating policies in a hypothetical dual-constrained job shop. *Management Science*, *22* (12), 1362–1371.