IIE Transactions (1999) **31**, 771–781

Controlling shop floor operations in a multi-family, multi-cell manufacturing environment through constant work-in-process

B. GOLANY, E.M. DAR-EL and N. ZEEV

Faculty of Industrial Engineering and Management, The Technion – Israel Institute of Technology, Technion City, Haifa 32000, Israel
E-mail: golany@ie.technion.ac.il

This paper discusses pertinent issues in applying CONstant Work-In-Process (CONWIP) principles to control shop floor operations in a manufacturing environment characterized by several product families processed along different routes in several production cells. The approach we take is to *simultaneously* answer two major questions: (1) what is the best WIP level? and (2) how to arrange the backlog list for a given system? The problem is posed as a mathematical programming model and solved via a simulated annealing heuristic. We design an experiment that captures essential elements of the systems under investigation. We then execute an extensive simulation to evaluate the effectiveness of various control schemes in a multi-cell, multi-family production environment. Specifically, we compare two variants of CONWIP control, one where containers are restricted to stay within given cells all the time and the other where containers are allowed to move through the entire system. We demonstrate the superiority of the latter in all the simulated scenarios.

1. Introduction

This paper investigates CONstant Work In Process (CONWIP) based Shop Floor Control (SFC) of manufacturing environments in which products are grouped into families and machine centers are grouped into cells that are designed to take advantage of the families grouping. The products are assembled into families according to similarities in their production attributes and the machine centers are grouped so as to eliminate, as much as possible, the need to perform costly set-ups when handling the operations within each product family within each group. This essentially describes a Group Technology (GT) design with the possibility that some jobs may be clustered into more than one group. Thus, each product family requires processing in one or more cells where inter-cell flow is done sequentially. Set-ups take place between cells, while intra-cell flow follows a flowshop pattern whose job processing is sequence independent.

Research on GT has mostly focused on the efficiency of the grouping process with respect to some defined criteria. The data input to GT studies include a *complete* description of the jobs together with their respective routings and the outcome of these studies are the groupings into families and subfamilies. Scheduling of operations in a GT manufacturing environment is a relatively recent development [1]. However, most reported GT-related scheduling procedures have thus far assumed given groupings that were determined at an earlier stage. A departure from this practice has recently been reported in Kannan and Ghosh [2] who develop a Virtual Cellular Manufacturing (VCM) model in which virtual cells are dynamically organized around groups of items thus preventing the loss of pooling synergy that happens when machines are permanently assigned to given cells. Scheduling operations in the VCM environment recognizes family affiliation thus yielding setup efficiencies while maintaining pooling synergy. Still, family groupings are done first and scheduling is then done as a second stage. We believe that it may be possible to develop scheduling requirements that will influence the actual GT groupings themselves.

The purpose of GT grouping is to cluster different parts having almost identical routings into families, thus creating flowshop-like manufacturing conditions within each family. A trade-off exists between functional and GT layouts. The partitioning of machine centers in the former, and reinstalling them as 'cells' in the latter, may cause delays and larger WIP sizes if it is not accompanied with significant reduction in setup times and other performance improvement means [3]. Consequently, setup requirements between jobs at each machine in GT settings are short and sequence independent. So, one can safely assume that setups bare no influence on the sequence of jobs through a specific manufacturing cell.

This paper proposes that jobs clustered within each GT family will be scheduled as a CONWIP production system. Using a SFC mechanism proposed by Spearman *et al.* [4], we model the production environment as a closed system where WIP is held at a constant level by allowing only a fixed number of containers (cards) to circulate in the system. Batch sizes, or transfer lot sizes, are predetermined by considering container capacity, setup time and costs on the various machines, etc. The order of loading empty containers with batches is determined by a backlog list. Constructing and maintaining the backlog list is a major control activity in the system. Figure 1 illustrates the operations of a single-cell CONWIP system. The bold arrows represent material flows (i.e., loaded containers) and the plain arrows represent information flows (i.e., empty containers). The loaded containers, each with its particular batch (depicted in the figure by different shapes), form queues in front of the machine centers that are associated with their routing. Setups are generally expected to occur between sequential processes along the single-cell CONWIP system. Each container carries a batch of items through a sequence of machines, designated here as a 'loop'.

The single-cell CONWIP structure contains only a single loop. When work on the batch is completed, the items are removed from the container and placed in the finished goods inventory. The vacant container then returns to the entry point of the system where it joins a queue of containers waiting for subsequent batches to be processed.

The essential system parameters in CONWIP are the size of the WIP (number of containers and their capacity) and the sequencing rules for the backlog list. Very little has been published in this area and the research that has been reported has focused mainly on the issue of WIP size. This was done by analyzing single-item systems characterized with either stochastic processing times or with deterministic processing times and stochastic breakdowns. Duenyas *et al.* [5] have considered a system that produces a single item with a deterministic processing time and random breakdowns. They were able to

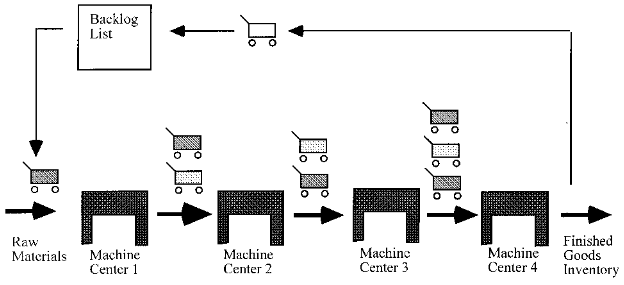

Fig. 1. Schematic CONWIP system.

analytically characterize the mean and standard deviation parameters of the production throughput as a function of the WIP size (measured by the number of containers/cards). They pointed out, however, that the analysis of more complex systems would require resorting to simulation techniques. Duenyas [6], Duenyas and Keblis [7], Dar-El *et al.* [8] and Herer and Masin [9] have shifted the analysis to CONWIP systems with one item and stochastic processing times. Duenyas [6] succeeded in developing an approximation for the system throughput for a general distribution of the processing times and later verified the approximation via simulation. Duenyas and Keblis [7] further explored these approximations and compared them to Kanban control systems in the context of an assembly system.

In multi-loop CONWIP systems, when the batch has finished its processing in a loop (one or several cells), the items are removed from the container and either loaded on a vacant container in a subsequent loop or placed in storage waiting to be picked-up later. Vacant containers returning to the beginning of the loop, are loaded with new batches on a FIFO basis except for the first loop in the system where loading follows the order dictated by the backlog list. In both single-loop and multi-loop systems, some containers may wait a while before being reloaded (e.g., when the first machine in the line has not yet finished processing a previous batch). Thus, actual WIP may in fact be smaller than the number of containers. However, in this paper we shall continue to refer to the number of containers as the WIP level.

Research on multi-stage, multi-item production systems using Kanban and other, more traditional controls, has been widely reported in the literature [10,11]. However, CONWIP related publications have been limited to a single-item (or a single family of items), single-cell production systems. Applying CONWIP to multi-family, multi-cell scenarios has hardly been mentioned until recently. We are aware of only one reference where this matter has been discussed [12, p. 439] where the authors elaborate on some of the difficulties associated with this topic. But, their discussion is qualitative and they do not provide a quantitative method to compute the system parameters.

Another characteristic of the research reported thus far is that the WIP level was handled separately from the issue of arranging the backlog list (most of the previous research simply ignored the latter). Thus, the approach outlined in this paper is innovative in both of the areas just mentioned: it extends the applicability of the CONWIP principles to multi-cell, multi-family systems and it offers a *simultaneous* analysis of the WIP size and the ordering of the backlog list.

The technique developed in this paper allows us to explore some properties of CONWIP SFC in the manufacturing environment being investigated. In particular, we compare the performance of a multi-loop CONWIP system, where each cell is associated with a fixed number of containers, and a single-loop CONWIP system, where the restriction on the number of containers applies only at the system level. The comparison clearly indicates the superiority of the latter system over the former.

The paper is organized as follows. Section 2 describes the manufacturing environment assumed for this research. Section 3 develops a mathematical programming model aimed at finding optimal WIP levels and backlog list arrangements. A simulated annealing procedure is adapted to solve the model. Section 4 provides the details of the experimental design and Section 5 summarizes the results of the simulated experiments. Finally, Section 6 summarizes our findings and offers some conclusions and open questions for further research.

2. Description of the manufacturing environment

The multi-cell systems we model represent realistic GT environments where material flows within and among cells (i.e., some jobs may be clustered into more than one group of machines). The implementation of such systems is partially motivated by their flexibility. Multi-stage, multi-item production systems offer the advantage of scheduling flexibility which induce shifting bottlenecks in each manufacturing cell – a fact that is shown to contribute to lowering the average flow time in the system [12, p. 515], and [13].

Each cell can process items of two kinds:

- End items – these are items that are pulled out of the system (thus, freeing their container) immediately after the last machine in the cell finished their processing.
- Intermediate items – these are items that must be processed in succeeding cells. Hence, they must wait until the first machine in the succeeding cell is ready to start processing them.

It is assumed that there is a steady supply of jobs to the backlog list for the entry cells. Supply of jobs into subsequent cells is dependent on the work being processed through the preceding cells. Work performed within each of the manufacturing cells in essence follows a flowshop pattern with negligible setup times between jobs at each machine.

Two types of manufacturing environments are considered. In the first type, we assume that the setups are fundamentally sequence-independent and add the setup time to the processing time for each job at each machine. Such cases may occur in multi-model assembly systems where components and sub-assemblies are manufactured in the entry cells and later arrive in kits to the assembly cells where there is no need for a specific setup before a new batch is started. Also, as with the entry cells, this does not affect the sequence of jobs through the assembly cells whose setup times within the cell remain zero.

In the second type, we extend the analysis to include setup times that are incurred each time a new job enters a manufacturing cell. This extension makes the analysis more realistic since in many cases there are certain preparations, cleanup operations, re-arrangement of tools, etc. that must be done to accommodate a new family of parts entering a cell.

We assume there are no machine breakdowns, nor any shortages of raw material entering the production system. Processing times are taken as deterministic but these vary between jobs on each machine.

We distinguish between two SFC systems (multi-loop and single-loop CONWIP) that differ in the way information flows are managed in them. In the multi-loop CONWIP system, each cell has a fixed number of containers (that together sum up to a fixed number of containers for the entire system). In the single-loop CONWIP system, there are a fixed number of containers in the system but there is no restriction on their distribution among the cells. Thus, multi-loop CONWIP allows tighter control on the system but, on the other hand, is likely to fall behind single-loop CONWIP in its overall throughput due to situations of 'starvation' or 'blockage' in particular cells during the course of the production schedule.

3. Model development

In this section we formulate the problem of finding optimal WIP levels and arranging the backlog list in a multi-loop CONWIP setting as a mixed-integer linear programming model. The formulation of the single-loop model is simpler since there are fewer variables (no need to formulate the individual number of containers in each cell) and due to space limitation it is omitted. More details are available from the authors upon request.

We employ the following notation:

Parameters

t_{jkc} = time to process item j on machine k in cell c ($j = 1, \ldots, n$; $k = 1, \ldots, m$; $c = 1, \ldots, C$);
C = number of cells;
W = number of containers in the system;
n_c = number of items to be processed in cell c ($c = 1, \ldots, C$);
m_c = number of machines in cell c ($c = 1, \ldots, C$);
$s(j, c)$ = the succeeding cell for item j after finishing its processing in cell c;
$p(j, c)$ = the predecessor cell for item j before starting its processing in cell c;
M = large number.

Variables

y_{jkc} = point in time when item j finishes processing on machine k at cell c;

w_c = number of containers (cards) in cell c ($c = 1, \ldots, C$);
I_{jc} = number of loaded containers in cell c when item j enters the cell;
Z_{ijc} = a zero-one indicator signaling whether job i enters cell c ahead of job j;
R_{ijc} = a zero-one indicator signaling whether job i leaves cell c before job j enters it.

We can formulate the problem as:

$$\text{Min } C_{\max}$$

subject to

$$y_{j1s(j,c)} - t_{j1s(j,c)} \geq y_{jm_c c}, \quad \forall \{(j,c) \mid s(j,c) \neq 0\}, \tag{1}$$

$$y_{j1c} - t_{j1c} \geq 0, \quad \forall \{(j,c) | p(j,c) \neq 0\}, \tag{2}$$

$$y_{jkc} - t_{jkc} + M \times (1 - Z_{ijc}) \geq y_{ikc}, \quad \forall 1 \leq i,j \leq n, 1 \leq c \leq C, 1 \leq k \leq m_c, \tag{3}$$

$$y_{jkc} - t_{ikc} + M \times Z_{ijc} \geq y_{jkc}, \quad \forall 1 \leq i,j \leq n, 1 \leq c \leq C, 1 \leq k \leq m_c, \tag{4}$$

$$y_{jkc} - t_{jkc} \geq y_{j,k-1,c}, \quad \forall 1 \leq j \leq n, 1 \leq c \leq C, 2 \leq k \leq m_c, \tag{5}$$

$$\sum_{c=1}^{C} w_c = W, \tag{6}$$

$$I_{jc} \leq w_c \quad \forall \{(j,c) \mid j \in c\}, \tag{7}$$

$$I_{jc} = \sum_{i=1}^{n} Z_{ijc} - \sum_{i=1}^{n} R_{ijc} \quad \forall \{(j,c) \mid j \in c\}, \tag{8}$$

$$(y_{j1c} - t_{j1c}) - (y_{i1s(i,c)} - t_{i1s(i,c)}) \leq M \times R_{ijc}, \quad \forall \{(i,j,c) \mid i,j \in c, s(i,c) \neq \emptyset\}, \tag{9}$$

$$(y_{i1s(i,c)} - t_{i1s(i,c)}) - (y_{j1c} - t_{j1c}) \leq M \times (1 - R_{ijc}), \quad \forall \{(i,j,c) \mid i,j \in c, s(i,c) \neq \emptyset\}, \tag{10}$$

$$(y_{j1c} - t_{j1c}) - y_{imc} \leq M \times R_{ijc}, \quad \forall \{(i,j,c) \mid i,j \in c, s(i,c) \neq \emptyset\}, \tag{11}$$

$$y_{imc} - (y_{j1c} - t_{j1c}) \leq M \times (1 - R_{ijc}), \quad \forall \{(i,j,c) \mid i,j \in c, s(i,c) \neq \emptyset\}, \tag{12}$$

$$C_{\max} \geq y_{jm_c c} \quad \forall 1 \leq j \leq n_c, \quad 1 \leq c \leq C. \tag{13}$$

Explanation of the model

(1)–(2) No job can be assigned to more than one cell at the same time.
(3)–(4) Starting the processing of job j on machine k in cell c depends on the completion of the previous job on the same machine.
(5) Starting the processing of job j on machine k in cell c depends on the completion of the same job in the previous machine.
(6)–(12) Forcing a constant WIP in each cell (through the definition of the $R_{i,j,c}$ variables. Notice that, by abuse of notation, $j \in c$ implies that job j is processed in cell c.
(13) Defining the overall completion time $C_{\max}$.

The model given above, even though it assumes deterministic processing times, belongs to the class of NP-complete models. This follows the fact that sequencing a flowshop system, a particular case of the CONWIP system, is already NP-complete when the number of cards is greater than the number of items [14]. Thus, solutions must be found via some heuristic mechanism. The heuristic approach we have chosen to employ is Simulated Annealing (SA). This technique attempts to overcome the inherent difficulties encountered by various descent algorithms that stop the search once they have landed on a local optimum. Instead, SA can move out of the neighborhoods of local optima by accepting worse candidates for solution with some probability that changes with the number of iterations. SA has been successfully implemented on a variety of tough problems such as the traveling salesman problem [15], the facility layout problems [16] and, most relevant to our interest, sequencing jobs in flowshop production systems [17–19].

The adaptation of a generic SA procedure to a specific problem, requires several inputs. These include: (1) a description of possible system configurations; (2) a generator of random changes in the system; and (3) an objective function. The minimization of cost in the objective function is analogous to the minimization of energy, in the original annealing procedures, as is discussed by Metropolis *et al.* [20]. A control parameter T (analogous to temperature) together with a "cooling" schedule determine the number of consecutive system changes between changes in T. The performance of SA procedures was shown in previous research [16] to be sensitive to the selection of the elements just mentioned. However, no general technique is known to select the best parameters for any scenario that is investigated and, as a result, customization of the system parameters is necessary in every case. The SA parameters chosen for our analysis are described in the Appendix.

The SA algorithm selection of candidate solutions applies directly to both multi-loop and single-loop CONWIP systems. For the multi-loop CONWIP system, one needs to first select (again, at random) whether the new solution will reflect a change in the distribution of the WIP among the cells or a change in the backlog list. If the change is in the WIP, the algorithm selects at random two cells, i and j. If the number of containers in cell i is larger than one it moves one container to cell j. If the change is in the backlog lists, a cell is selected at random and its backlog list is changed as explained in the Appendix.

Evaluating the value of the objective function in a single-cell CONWIP system is rather straightforward since a loaded container that enters the system must go through the (flowshop) sequence of machines. Therefore, the containers are released in the same order as they enter the system and estimating the objective function value is possible by way of a recursive function. In multi-cell systems, however, a queue is formed in front of each cell and the order in which empty containers are released doesn't necessarily follow their loading order. Thus, in spite of the deterministic nature of these systems, one cannot use a trivial function to find the entrance order into subsequent cells. To overcome this difficulty, a simulation program was written to estimate the objective value in these cases. Incidentally, pairwise exchange configurations are not guaranteed to be feasible in multi-loop CONWIP systems. When an infeasible solution was detected, the program assigned to it a very large objective value making the chances for its acceptance as the new current solution slim.

4. Experimental design

A series of experiments, consisting of two major stages, was designed. The purpose of the first stage was to evaluate the performance of the SA algorithm and to test the validity of the parameters chosen in its construction. The second stage was planned for assessing the two SFC alternatives that are here considered for controlling the multi-family, multi-cell manufacturing environment: multi-loop CONWIP and single-loop CONWIP.

Two sets of experiments were run in this paper. The first one represents a system with sequence independent setups (allowing for adding the setup time to the processing time for each job). This set of experiments serves to validate the SA procedure as well as to provide an in-depth investigation of the SFC mechanism and a comparison of the two CONWIP techniques modeled in this paper. The second set represents a much larger system where setup times are considered separately from processing times. This set serves mainly as an illustration of the CONWIP-based SFC under more realistic settings.

(1) The first set models a layout that contains three manufacturing cells with five machines in each cell. Initially, jobs may either pass through cell 1 or 2. Upon completion, these jobs may either pass through cell 3 or get dispatched outside the system. Cells 1 and 2 are essentially machining or fabrication lines, while cell 3 represents a multi-model assembly line.

The shop processes four product families with as many as 30 products in them. The experiments were designed as a sequence of runs with an increasing number of products, starting with four products (one in each family). Each new run added a product to the next family according to a pre-determined order so that the number of products within each of the four families in the largest run were eight, eight, seven and seven (as shown in the first two columns in Tables 2 and 3). The layout along with the processing routes for the product families is shown in Fig. 2.

(2) The second set models a layout that contains eight cells with two to five machines in each cell. Here, jobs must start in cells 1, 2, or 3. Then, some jobs continue to either cell 4 or 5 while others by-pass this stage and are routed straight on to either cell 6 or 7. Finally, some jobs leave the system through cell 8 while others leave after being processed in cells 6 or 7. These routings define eight product families that are processed in this environment.

The shop processes eight product families with up to four products in each family. The simulation hierarchy consisted of three levels. At the upper level, there were 10 runs (each starting with a new set of randomly generated processing times). For each one of these runs, the second level runs through over 30 WIP levels (from WIP = 2 to 32) and the third level implements four different models:

(i) Single-loop CONWIP (using SA).
(ii) Multi-loop CONWIP (again, using SA).
(iii) Best of random solutions (the number of random solutions attempted was equal to the number of permutations used by the single-loop technique.
(iv) No-split heuristic (forcing the production of all the products belonging to the same family in a consecutive manner to avoid setups). All 8! = 40 320 possible permutations were tried and the best solution was used for comparison purposes.

The second layout is given in Fig. 3.

The design of the shop floor just described was done in a way that will maintain close contact with real-world shop floor situations. Each cell may processes more than a single product family. The entry cells handle both end-items and intermediate products. The products belong to two categories: those completing their entire required processing in a single cell and those who require pro-

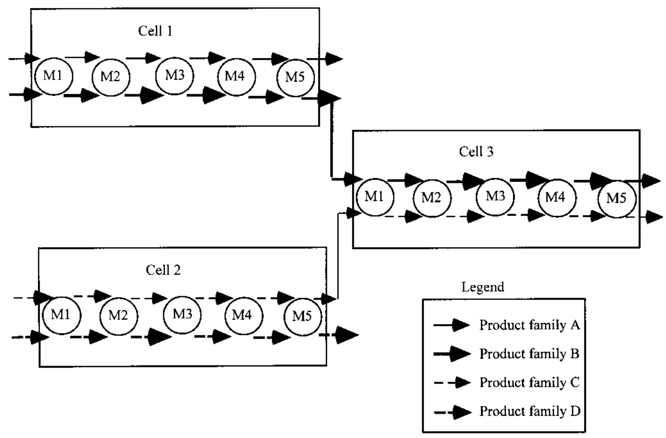

Fig. 2. Material flows in the 3-cell system.

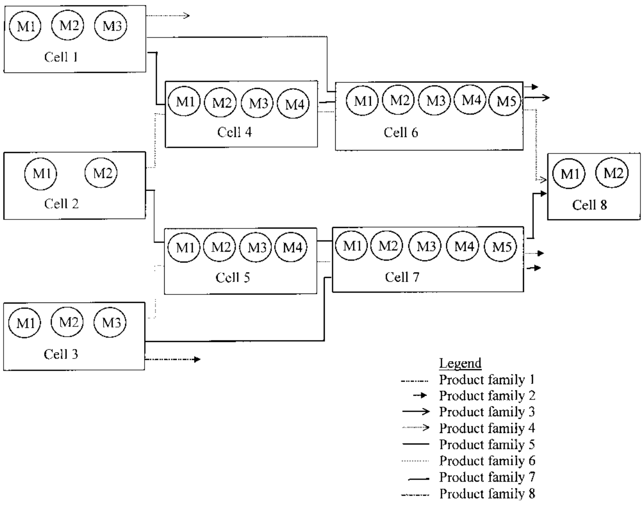

Fig. 3. Material flows in the 8-cell system.

cessing in two or more cells. The latter group requires processing using at most 12 machine centers. Our experience shows that as a result of recent trends towards outsourcing and flattening of the Bill-of-Materials it is very rare to find products who go through more than 10 processing centers in a given shop.

5. Simulation results

5.1. *Layout no. 1*

This layout was used first to validate the models and then to analyze the performance of the two competing CONWIP SFC systems (single versus multi-loop).

5.1.1. *Validation stage*

The validation and testing of the SA procedure and its parameters were done through a number of preliminary simulation runs in the first set of experiments. These runs were aimed at evaluating the SA application with respect to two criteria: the required computation effort (in CPU time) and the quality of its solutions (in comparison to optimal and other heuristic solutions).

5.1.2. *Evaluating the computational effort*

The following experiment was designed to evaluate the CPU time requirements of the SA algorithm:

- Up to 30 products, divided into four families (seven or eight products in each family).
- 15 machines, evenly divided into three cells.
- Processing times for the four products families were generated from a Uniform [0,100] distribution and repeated for three WIP levels (four, eight and 12 containers).
- The required CPU was computed as the average of 20 independent simulation runs for each system (single-loop CONWIP and multi-loop CONWIP).

The time required to obtain a SA solution for each of the SFC systems was observed as a function of the number of items. A power function of the form: CPU (in seconds) $= a \times n^b$ was fitted to the outcomes for each WIP level. Table 1 presents the fitted parameters and the correlation coefficient that describe the quality of each fit.

As expected, when the number of items increases, the required CPU time also increases. However, in all cases the power parameter that was fitted was smaller than two, i.e., less than a quadratic function. This conclusion concurs with the finding of Wilhelm and Ward [16] who used similar evaluation of the SA algorithm in the context of solving quadratic assignment problems. As for identifying unique values the parameter a, b in the function describing the relations between the number of items (n) and the CPU time, this evaluation showed that these depend on the WIP levels. Further, the results show that the performance of the algorithm may be dependent on specific problem structures in a way that makes it difficult to predict which WIP levels will require more computation time.

5.1.3. *Solution quality*

The processing time for each product on each machine in this evaluation was determined as follows: zero with probability 0.2 and a realization taken from a Uniform [60,100] distribution with probability 0.8. The SA solutions were compared with optimal solutions (for small size problems) and with random solutions for larger problems. We were forced to compare the SA to random solutions since no heuristic that was specifically built to solve the multi-cell problem is mentioned in the current literature. Specifically, the SA procedure was compared with the best random solution out of thousands that were generated using the same CPU time as the one consumed by the SA procedure. Each comparison was based on the average of 10 independent simulation runs.

Tables 2 and 3 compare the performance of the SA algorithm with exact solutions for small problems with various configurations containing several product types and different WIP levels. The SA algorithm was able to obtain in most cases the optimal solution (the maximal deviation here was 2%). It is worth while noting that the few cases where deviations from optimal solutions were observed are associated with "boundary" WIP levels, i.e.,

Table 1. Fitting parameters to the CPU function

System	*WIP level*	a	b	r^2
Single-loop CONWIP	4	1.47	1.027	0.95
	8	1.69	1.161	0.97
	12	1.42	1.218	0.97
Multi-loop CONWIP	4	0.43	1.755	0.97
	8	0.23	1.900	0.99
	12	0.30	1.672	0.99

Table 2. Comparison of SA and exact solutions for single-loop CONWIP system

Family	*Number of products*	*Total number of products*	*WIP levels*	*SA solution*	*Optimal solution*	*Deviation (%)*
1	2		3	921	921	0.00
2	1	5	4	841	841	0.00
3	1		5	774	774	0.00
4	1		6	774	774	0.00
1	2		3	1313	1313	0.00
2	2		4	1158	1158	0.00
3	1	6	5	921	921	0.00
4	1		6	862	862	0.00
			7	862	862	0.00
1	2		3	1525	1525	0.00
2	2		4	1225	1223	0.16
3	2	7	5	1090	1090	0.00
4	1		6	927	927	0.00
			7	927	927	0.00

Table 3. Comparison of SA and exact solutions for multi-loop CONWIP system

Family	*Number of products*	*Total number of products*	*WIP levels*	*SA solution*	*Optimal solution*	*Deviation (%)*
1	2		1	2078	2078	0.00
2	1		2	1053	1053	0.00
3	1	5	3	791	791	0.00
4	1		4	774	774	0.00
			5	774	774	0.00
1	2		1	2787	2787	0.00
2	2		2	1418	1418	0.00
3	1		3	1036	1036	0.00
4	1	6	4	863	863	0.00
			5	862	862	0.00
			6	862	862	0.00
1	2		1	3316	3316	0.00
2	2		2	1680	1680	0.00
3	2		3	1233	1233	0.00
4	1	7	4	1009	1009	0.00
			5	927	927	0.00
			6	927	927	0.00
			7	927	927	0.00
1	2		1	3632	3632	0.00
2	2		2	1816	1816	0.00
3	2		3	1276	1251	2.00
4	2	8	4	1036	1036	0.00
			5	927	927	0.00
			6	927	927	0.00
			7	927	927	0.00
			8	927	927	0.00
1	3		1	1947	1947	0.00
2	2		2	1354	1354	0.00
3	2		3	1072	1058	1.32
4	2	9	4	971	971	0.00
			5	927	927	0.00
			6	927	927	0.00
			7	927	927	0.00

levels where throughput approaches its maximal value. When the WIP was increased beyond that point the SA algorithm always found the optimal solution.

Figures 4 and 5 compare SA with heuristic procedures. The figures show C_{max} values that were obtained for various WIP levels as well as the differences between random and SA solutions in the single and multi-loop CONWIP systems, respectively.

Again, the SA algorithm outperformed the random solutions in all cases and its superiority was particularly evident near the boundary WIP levels.

5.1.4. *Analysis stage*

In the previous stage we have established that the SA technique works effectively for the simple form of the production control systems we are interested in. Now, we turn to investigate some of the properties of these systems using SA as the tool to obtain ordered backlog list needed to control the systems. Our objective is to explore the properties of the two competing SFC systems and to identify which one of them is superior to the other.

The comparison was based on 10 independent simulation runs for each system. The runs simulated a scenario with three cells, five machines in each cells with four families and five products in each family (i.e., 20 products in total). The processing times for each family on each machine were taken from a Uniform (0,100) distribution.

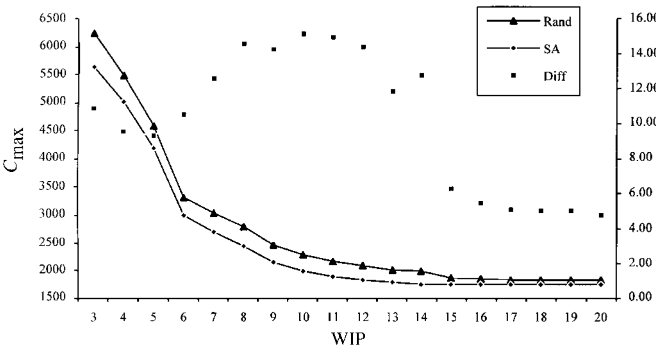

Fig. 4. Difference between SA and random solutions in the single-loop system.

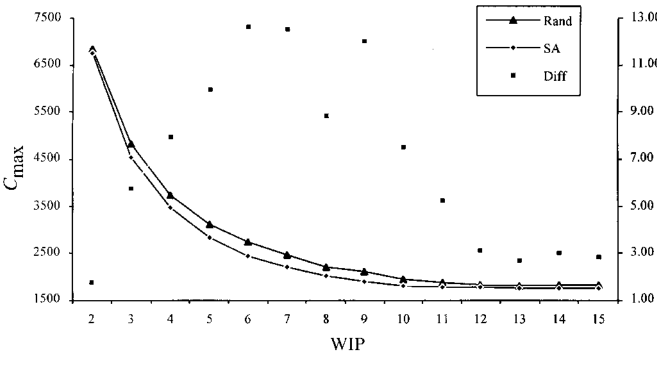

Fig. 5. Difference between SA and random solutions in the multi-loop system.

The average C_{max} and WIP outcomes are given in Fig. 6.

Observations:

- The results came out in favor of the single-loop CONWIP SFC that dominates multi-loop CONWIP throughout all WIP levels.
- Both systems have reached their maximal throughput with 10 containers (although the number of machines in the entire system is 15). Namely, on average one finds at any given point of time about five idle machines in the system. This is a result of the imbalance in the processing times of the different products.
- A qualification that must be added to these results is that the space of possible configurations for single-loop CONWIP is much smaller than that of multi-loop CONWIP. Therefore, one might attribute the better results of single-loop CONWIP to SA's ability to search more effectively the reduced feasible space of this SFC.

5.2. *Layout no. 2*

The purpose of this experiment was to simulate a production environment that is more realistic than the first layout and use it to compare the performance of the two competing SFC systems to each other and to two other heuristics (best of random solutions and best of no-split solutions).

As shown in Fig. 3, the second layout consists of eight cells with two to five machines in each cell. There are eight product families with four products in each family. Again, 10 independent runs were used to evaluate the SFC systems in this layout. In each run, the processing times for each family on each machine were drawn again from a uniform distribution. Three of the four systems were evaluated with respect to 31 WIP levels (1–32) while the fourth, the multi-loop system, was evaluated only with respect to 21 WIP levels (10–31). This was done since

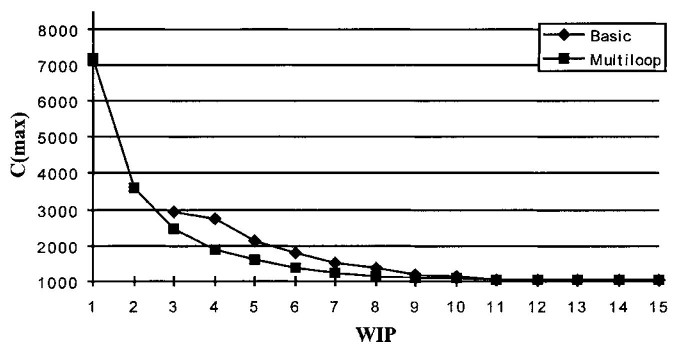

Fig. 6. Average C_{max} versus WIP in single-loop and multi-loop system.

the latter system must maintain eight separate loops for the eight cells with at least one CONWIP container in each loop.

The outcomes of this experiment are summarized in Figs. 7–11. Starting with the C_{max} evaluation (Fig. 7), we find that the single-loop CONWIP system outperformed all the other systems across all WIP levels. Its C_{max} was on average 4.4% smaller than that of the best random solution and about 2.8% better than the no-split solution (although the latter picks up the best of 8! solutions). The multi-loop system starts with very poor C_{max} values at low WIP levels and slowly converges towards the results of the other systems (though it always stays above them). The single-loop system reaches its maximal throughput at WIP levels 19–20. Since the second layout consists of 28 machines (see Fig. 3), this means that at any given time one will find about one-third of the machines in an idle position. This result reinforces the same finding (convergence at WIP level 10 in a 15-machine layout) reported earlier for the first layout.

Figure 8 presents the average number of setups (changes between families) required by the different systems. The no-split solution which forces the production of all the requirements for a particular family in a consecutive order is naturally better than all the other systems in this respect. It is followed by the multi-loop, single-loop and random systems in that order.

Figure 9 offers a similar view as Fig. 7 only instead of looking at the longest completion time it focuses on the time spent by all items in the system (ΣF_i). Again, the single loop outperforms its rivals by about the same proportions as reported for the C_{max}.

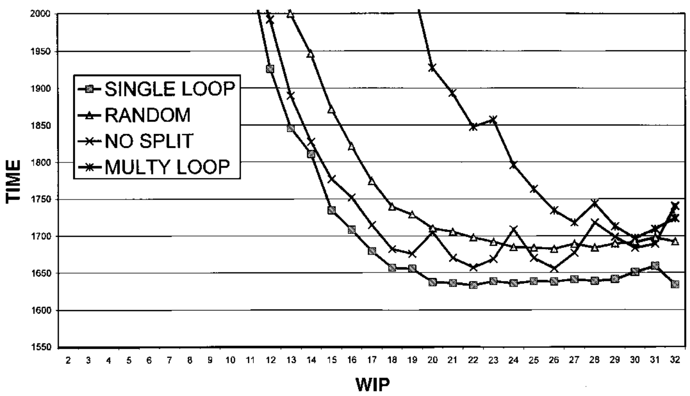

Fig. 7. C_{max} versus WIP for layout number 2.

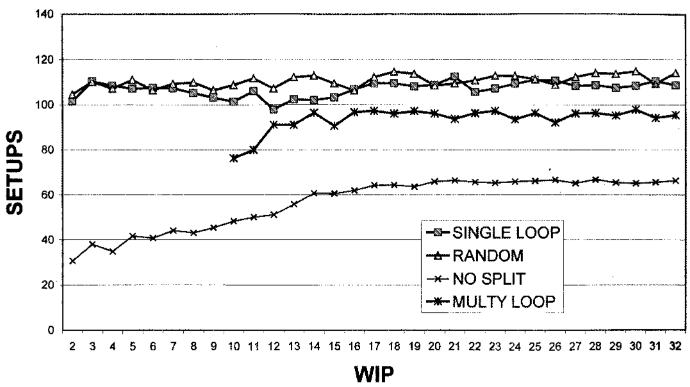

Fig. 8. Number of setups (layout number 2).

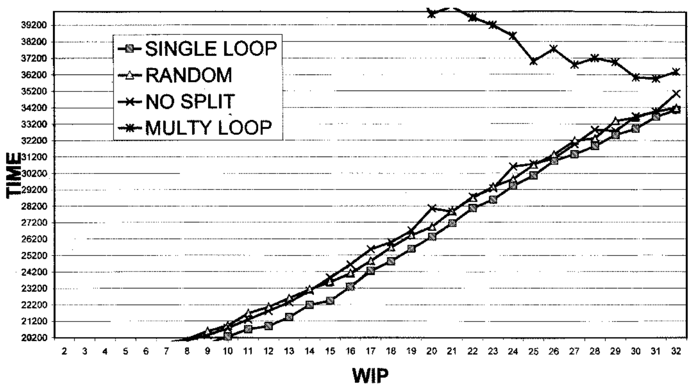

Fig. 9. Flow time (ΣF_i).

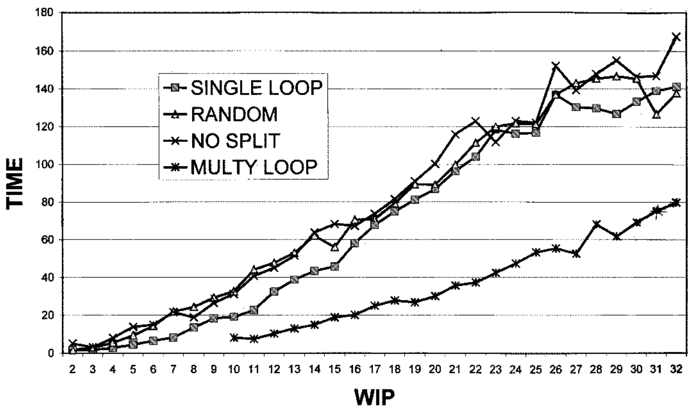

Fig. 10. Waiting time at non-bottleneck machine (layout number 2).

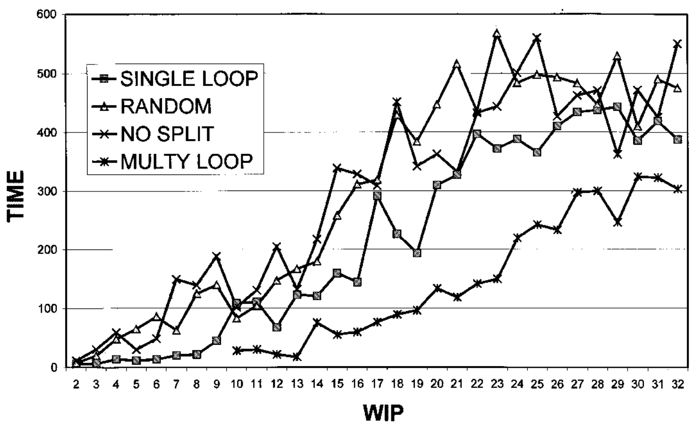

Fig. 11. Waiting time at bottleneck machine (layout number 2).

Finally, Figs. 10 and 11 look at the queuing patterns before bottleneck and non-bottleneck machines created by the four competing systems in the different WIP levels. The multi-loop system is clearly the winner in that dimension. This result apparently stems from keeping separate loops for the eight cells, a fact that makes it able to move items within each cell quicker than the other systems. However, as the previous figures show, it apparently suffers from long delays in moving from one cell to another. These delays eventually determine its longer $C_{\max}$ and ΣF_i.

6. Summary

This paper extends previous research on CONWIP controlled systems into multi-family, multi-cell settings. The paper presents a mathematical programming model that accurately describes the problem and then solves it using a simulated annealing heuristic. The generic SA algorithm was first customized for the specific needs of the systems under investigation, and then compared with exact solutions in ordering the backlog list of the jobs entering the system for small problems. The comparisons indicated that the SA procedure achieves solutions that are within 1–2% of the optimal solution while checking a small proportion of the possible configurations. The SA was further compared with two other heuristics for larger problems and was found as superior to them in all cases.

The model was used to investigate the properties of systems where material is allowed to flow between cells. Two kinds of systems were compared. In one, the multi-loop CONWIP system, containers are restricted to stay in given cells while in the other, a single-loop CONWIP system, containers can circulate everywhere in the system. The latter system was found to be superior to the former, that is, under any restriction on the overall number of containers in the system, the single-loop CONWIP system enables better throughput than the multi-loop CONWIP system. We note, however, that in comparing the two systems we have not dwelt on the qualitative aspects of their operations (e.g., convenience of control, computational efficiency, etc.) but rather focused only on the quantitative issues of WIP size and throughput.

This research focused on a single period analysis where the demand to all the items was known in advance. A natural extension is to explore multi period scenarios where the demand in each period is known in certainty at the beginning of the period. These dynamic planning situations will have to account for the WIP that is already present at the system at the beginning of each period. Also, adding stochastic elements that characterize real-world systems (e.g., machine breakdowns and uncertain demand) will make the analysis more realistic. The objective of such future research will be to analyze the trade-offs between WIP and throughput in the CONWIP context and to establish robust rules for efficient determination of WIP sizes and sequencing of jobs in the backlog list.

References

[1] Ruben, R.A., Mosier, C.T. and Mahmoodi, F. (1993) A comprehensive analysis of group scheduling heuristics in a jobshop cell. *International Journal of Production Research*, **31**(6), 1343–1369.

[2] Kannan, V.R. and Ghosh, S. (1996) A virtual cellular manufacturing approach to batch production. *Decision Sciences*, **27**(3), 519–538.

[3] Suresh, N.C. and Meridith, J.R. (1994) Coping with the loss of pooling synergy in cellular manufacturing systems. *Management Science*, **40**, 466–483.

[4] Spearman, M.L., Woodruff, D.L. and Hopp, W.J. (1990) CONWIP: a pull alternative to kanban. *International Journal of Production Research*, **28**(5), 879–889.

[5] Duenyas, I., Hopp, W.J. and Spearman, M.L. (1993) Characterizing the output process of a CONWIP line with deterministic processing and random outages. *Management Science*, **39**(8), 975–988.

[6] Duenyas, I. (1994) Estimating the throughput of a cyclic assembly system. *International Journal of Production Research*, **32**(7), 1403–1419.

[7] Duenyas, I. and Keblis, M.F. (1995) Release policies for assembly systems. *IIE Transactions*, **27**, 507–518.

[8] Dar-El, E.M., Herer, Y. and Masin, M. (1999) CONWIP based production lines with multiple bottlenecks, performance and design implications. *IIE Transactions*, **31**, 99–111.

[9] Herer, Y. and Masin, M. (1997) Mathematical programming formulation of CONWIP based production lines; and relationship to MRP. *International Journal of Production Research*, **35**(4), 1067–1076.

[10] Bard, J.F. and Golany, B. (1991) Determining the number of kanbans in a multi-product, multistage production system. *International Journal of Production Research*, **29**(5), 881–895.

[11] Chang, T.M. and Yih, Y. (1994) Generic kanban systems for dynamic environments. *International Journal of Production Research*, **32**(5), 889–902.

[12] Spearman, M.L. and Hopp, W.J. (1996) *Factory Physics: Foundations of Manufacturing Management*, Irwin, Chicago, IL, p. 439.

[13] Greco, M.P. (1996) Sequential policy for a CONWIP production system. M.Sc. thesis, Department of Industrial & Systems Engineering, Virginia Tech, Blacksburg, VA.

[14] Lenstra, J.K., Rinnoy, K.A. and Brucker, P. (1977) Complexity of machine scheduling problems. *Annals of Discrete Mathematics*, **1**, 343–362.

[15] William, H. (1986) *Numerical Recipes – The Art of Scientific Computing*, Cambridge University Press.

[16] Wilhelm, M.R. and Ward, T.L. (1987) Solving quadratic assignment problems by simulated annealing. *IIE Transactions*, **19**, 107–119.

[17] Osman, I.H. and Potts, C.N. (1989) Simulated annealing for the permutation flowshop scheduling. *OMEGA, The International Journal of Management Science*, **17**, 551–557.

[18] Ogbu, F.A. and Smith, D.K. (1990) The application of the simulated annealing algorithm to the solution of the $n/m/C_{\max}$ flowshop problem. *Computers and Operations Research*, **17**(3), 243–253.

[19] Laarhoven, V.P.J.M., Aarts, E.H.L. and Lenstra, J.K. (1992) Job shop scheduling by simulated annealing. *Operations Research*, **40**(1), 113–125.

[20] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N. and Tellet, A.H. (1953) Equation of state calculation by fast computing machines. *Journal of Chemical Physics*, **21**, 1087–1092.

Appendix

Specification of the SA Algorithm

System Initialization: the algorithm is started with arbitrary (random) solutions [19].

Generation of neighboring solutions: Previous SA applications employed two mechanisms for generating neighboring solutions: pairwise exchange and insertion but neither of the two was generally proven as superior to the other. Thus we chose to adapt a pairwise exchange technique in the following manner as:

Pairwise Exchange Algorithm:
Step 1. Randomly select an integer $i \leqslant n$
Step 2. Randomly select an integer $j \leqslant n$
Step 3. If $BL(i) \neq BL(j)$ goto (4); otherwise, return to *Step (2)*.
Step 4. $BL'(i) = BL(j)$; $BL'(j) = BL(i)$

where

$BL(i)$ = identity of the item in the ith position in the current backlog list,
$BL'(i)$ = identity of the item in the ith position in the new (neighboring solution) backlog list.

Accepting a new current solution: a new candidate solution is accepted according to a certain probability function. The $C_{\max}$ criterion replaces the energy expression in the original work [2]. The probability function decreases with the decrease in the control parameter T_r. Each value of T_r defines a 'stage' in the procedure. Following Wilhelm and Ward [17], we let the procedure move into a new stage if one of the two conditions below is met:

- The procedure has reached an equilibrium status in its current stage where equilibrium is achieved when the average objective function value during a certain interval of iterations is not better than the average value in the three preceding intervals (of equal length)
- The procedure has attempted more than a predetermined maximal number of iterations $N_{\max}$ in its current stage.

The values for the control parameter were set, as in previous SA studies, by:

Algorithm for temperature setting
Step 1. Set T_1 so that the probability of accepting an average change in the objective value is 0.5.
Step 2. $T_{r+1} = T_r CR^{r-1}$, where CR (Cooling Rate) was set to 0.9.

Termination of the SA procedure: the SA procedure is stopped if over the course of three consecutive stages (T_r) no improvement was recorded in the objective function value.

To summarize, we write below the entire procedure.

Simulated Annealing Algorithm:
Step 1. Determine T_r for $r = 1, 2, \ldots$; Set $r = 1$.
Step 2. Select an initial solution and compute its objective value.
Step 3. Select a neighbor solution and compute its objective value.
Step 4. Compute the objective value for the current solution.
Step 5. Determine if the candidate solution replaces the current one. If it does, goto *Step 6*, otherwise, return to *Step 3*.
Step 6. Test whether the equilibrium conditions are met. If they do, goto *Step 7*. Otherwise, check the number of iterations executed in the current stage. If it is larger than $N_{\max}$, goto *Step 7*, otherwise, return to *Step 3*.
Step 7. $r = r + 1$. Test if the termination condition is satisfied. If it does, stop, otherwise return to *Step 3*.

Biographies

Ezey Dar-EI holds the Harry Lebensfelf Chair in Industrial Engineering at the Davidson Faculty of Industrial Engineering & Management at the Technion. He has published extensively in areas such as: the design and analysis of production/assembly systems; project and FMS scheduling and productivity development. Dr. Dar-El received his B.Mech.E, M.Eng.Sc. and Ph.D. from the University of Melbourne (Australia). He is a Fellow of IIE and WCPS and Senior Member of SME, HFS, IFPR, ISPQR and ORSIS. He is on the editorial Board of the *IJP*R, *IJPP&C* and the *Quality Observer*. Dr. Dar-El is an internationally known consultant having worked with over 60 companies located in Australia, UK, USA, The Netherlands Antilles, Italy, Peru and Israel.

Boaz Golany is an Associate Professor and Associate Dean at the Davidson Faculty of Industrial Engineering and Management at the Technion. He has a B.Sc (Cum laude) from Technion (1982), and a Ph.D. from the University of Texas at Austin (1985). He was awarded the Naor Prize of the Israeli Operations Research Society in 1982, the Alon Fellowship from the Israeli Education Ministry in 1986 and the Technion Academic Excellence Award in 1991. Dr. Golany has published over 50 papers in academic and professional journals and books and since 1996 he is an editor of *JPA*. His publications are in the areas of Industrial Engineering, Operations Research and Management Science. Dr. Golany had served as a consultant to numerous companies and agencies in Israel and the US in various areas including military milieu, marketing studies, efficiency improvement projects, inventory control and project management. He is an active member of INFORMS and ORSIS (the Israeli O.R. Society) where he served as the Treasurer in 1989–1992 and the vice-President 1996–1999.

Noam Zeev earned his B.Sc. in Physics (Cum laude, 1992) and a M.Sc. in Industrial Engineering from the Technion (1995). Currently, Mr. Zeev is a business consultant at Electronic Data Systems (EDS) Israel, where he is working in diverse areas including marketing, quality assurance and project management. Previously, Mr. Zeev served as an organizational consultant at Bank Hapoalim, the largest commercial bank in Israel.