European Journal of Operational Research 42 (1989) 22–38
North-Holland

# Theory and Methodology

# Performance evaluation for systems of pooled machines of unequal sizes: Unbalancing versus balancing

Kathryn E. STECKE and Ilyong KIM
*Graduate School of Business Administration, The University of Michigan, Ann Arbor, MI, USA*

**Abstract:** This research explores the appropriateness of unbalancing the workload per machine in certain types of flexible manufacturing systems (FMSs) configured as pooled machines of unequal sizes. Studies are conducted to examine the applicability of following an objective to unbalance workloads when solving the FMS planning problems of selecting part types to be machined together and determining their production ratios. Simulation is used to compare unbalancing and balancing on realistic, detailed models of flexible flow system (FFSs). The experiments are constructed to evaluate the impact of operational factors such as blocking, transportation, buffer utilizations, fixture requirements of various types, and different workload distributions among the machine types. The research results indicate that the aggregate and theoretically optimal unbalanced workloads provided by Stecke and Solberg (1981, 1985) using a closed queueing network model can be appropriate in a realistic FMS. Production rate and system and machine utilizations can all be higher when unbalancing workloads in systems of pooled machines of unequal sizes. It is also observed that: (1) system performance in terms of system utilization or production rate is sensitive to the appropriate number of pallets in the system, when either unbalancing or balancing; and (2) unbalanced part mix ratios conversely can lead to balanced machine utilizations among unequally sized pooled machine types. Overall system utilization seems to be more sensitive to the number of pallets in the system when unbalancing than when balancing. Further research needs are also noted.

**Keywords:** Flexible manufacturing systems, workload distribution

## 1. Introduction

Flexible manufacturing systems (FMSs)—the high-technology solution to factory production—are automated systems that contain several integrated CNC machine tools served by automated material handling equipment and supervised by one or more computers. They are capable of producing a variety of different part types simultaneously in an order dictated by a computer.

Received July 1987; revised August 1988



Stecke (1983) identifies five interrelated FMS planning problems as operational decisions that have to be made prior to system operation. These are:

(1) part type selection,
(2) machine grouping,
(3) part mix determination,
(4) resource allocation, and
(5) cutting tool loading.

The solutions to these planning problems for system set-up provide that all cutting tools required for each operation of the selected part types are loaded into the appropriate machines' limited capacity tool magazines. Once the FMS is set up, FMS scheduling is the next function. These problems can be solved either off-line or in real-time or both.

A research issue in FMS production planning and scheduling that is addressed here is to determine the appropriateness or not of unbalancing workloads for a system of pooled machines of unequal sizes. By using a stochastic multiserver closed queueing network (CQN) model, prior research studies indicate that the optimal assignment that maximizes production provides more workload per machine to larger groups of pooled machines than to each machine of a smaller group of pooled machines or to unpooled machines (see Stecke and Solberg, 1981, 1985). These theoretical, aggregate, and optimal unbalanced workloads per machine for various systems of pooled machines are provided, in this study, under the assumptions that the transporter and load/unload stations are infinitely fast and the size of the buffer at each machine group is adequate to hold all parts. The usual product form queueing network assumptions apply.

However, these theoretical and aggregate results have not been adequately examined on realistic systems, having limited resources, travel delays, and finite buffers, for example. A key element of our research study is the investigation of if, and under what conditions, unbalancing workloads can be appropriate in FMSs. There are two questions concerning the implementation of unbalancing in an FMS of unequally sized groups of pooled machines:

(1) For a particular FMS configuration, how well do the aggregate and theoretically optimal unbalanced workloads perform?

(2) Might these unbalanced workloads be useful to help solve other FMS planning problems?

In particular, we are concerned here with the effect of purposefully unbalancing workloads on the other production planning problems of selecting part types and determining part mix ratios.

Some previous research has investigated the accuracy of queueing network models in analyzing steady-state behavior of manufacturing systems. In one of the first such studies, Solberg (1977) developed a program modeling closed network of single class, multiserver queues, called CAN-Q. The results of this model have been compared to a detailed simulation of an existing 9-machine FMS. He notes that the system utilization, the nine machine utilizations, and production rates from the two models all differ by less than 3 percent. Dubois (1982) proposes a semi-open queueing network model having limited WIP. The aim is to extend Solberg's model to the case where an upper limit on the number of pallets allowed concurrently in an FMS exists. Deterministic simulations of an 8-machine flow system are compared with the queueing network model. He notes that the queueing model behaves worse for deterministic situations (e.g., periodic input) than for nondeterministic situations (e.g., poisson input). Suri (1983) indicates that the queueing model gives good results even for a system which does not follow the assumptions of queueing theory. For example, system performance measures are insensitive to the assumptions that processing times are not exponentially distributed. Buzacott and Shanthikumar (1985) use queueing models to analyze a dynamic job shop, where parts of a single type arrive continuously. They show the usefulness of queueing networks to model and analyze a job shop in which parts are received in a dispatch area, where their release to the shop may be controlled. The numerical results, using an open queueing model with a single class (one part type) and single, unpooled machines, are compared via simulation for the job shop. The results are very close. Suri and Diehl (1985) mention that queueing network models are not appropriate in certain situations, such as transient studies and if significant blocking might occur. They note that a queueing network model is especially useful in providing good first-cut information about the production capacity of an FMS configuration prior to the development of a scheduling rule. There are many studies that demonstrate properties of queueing network models, but do not compare results with simulation.

Many algorithms have been developed to balance workloads (i.e., see Wee and Magazine, 1981; and, Talbot, Patterson and Gehrlein, 1986), involving both optimal-seeking and heuristic approaches. This is because under ideal conditions in assembly lines, balanced workloads lead to maximum production rate, minimum idle time, and minimum WIP. Some research on FMS loading and related part input sequence problems use the objective of balancing workloads in an optimization procedure (see, i.e., Lin and Lu, 1984; Shanker and Tzen, 1985; and, Berrada and Stecke, 1986).

Moreover, several studies of the problem of selecting part types also aim to balance workloads. This problem is to select a subset of part types, often with production requirements and/or due dates, for immediate and simultaneous processing over some upcoming period of time. Whitney and Gaul (1985) partition part types into separate batches and distinct machining horizons. The goal is to minimize the number of batches and then balance workloads within each batch sequentially. This is an iterative approach that uses estimated performance indices. Hwang (1986) also attempts to balance workloads implicitly while minimizing the number of batches as the objective. His heuristic approach to minimize the frequency of tool changeovers seems to select the part types with the most number of required cutting tools as late as possible. The suggested two batching approaches do not guarantee the optimal or even near optimal solution to the problem of minimizing the number of batches. Production requirements of the part types are not considered explicitly, in order to make the problem tractable. In a related study, Rajagopalan (1986) partitions the part types, which have production requirements, into batches. A formulation to minimize the total makespan is developed under the assumption of a constant tool changeover time. Such an optimization problem is NP-complete. He also suggests balancing workloads as a good heuristic objective to select part types on a rolling horizon basis. However, the suggested heuristic rules do not consider the production requirements.

An alternative to batching is to both select the part types to be machined together and determine their production ratios using a flexible approach. A flexible approach to operate an FMS can be implemented as follows. When the production requirements of some part type(s) are finished, space in the tool magazine is freed up and some new part type(s) can perhaps be introduced into the system if this input can help make the system more highly utilized (see Stecke and Kim, 1986a). Use of a flexible approach results in the need for more frequent tool changes but in a decrease in tool changeover time. This is because when a part type finishes production only those few tools affected need to be changed. This is called a partial changeover. The reduction in tool changeover time should enable the use of such a flexible approach to lead to higher FMS productivity.

In a comparison of the flexible approach and various batching approaches from the literature, using the flexible approach enables the system to be more highly utilized over time (see Stecke and Kim, 1988). It is also noted in this study that the batching approaches require more fixtures of each type than the flexible. The system utilizations for the batching approaches seem to be sensitive to restrictions on the number of fixtures of each type.

The purpose of this paper is to investigate the appropriateness of using the aggregate CQN results to unbalance workloads for an FMS consisting of unequally sized pooled machine groups. The use of existing procedures that select aggregate part types and mix ratios on a dynamic basis is investigated for the operating objective of unbalancing workloads. A simulation model is introduced to show the advantages of unbalancing machine workloads over time in a realistic flexible flow system (FFS). An FFS is chosen so as to not confound the effects (advantages) of allowing alternative routes for a part type through pooling, with the additional advantages of allowing alternative routes in a general FMS. Statistics on blocking, transportation, buffer, cart, and machine utilizations, makespan, fixture requirements of various types, and different workload distributions among the machine types are compared for unbalancing and balancing.

This paper is organized as follows. In Section 2.1, the mathematical programming formulation that both selects part types and determines part mix ratios at various points in time is reviewed. Section 2.2 reviews solution procedures that determine aggregate part mix ratios for the objectives of balancing or unbalancing workloads using the suggested flexible approach to selecting part types. In Section 3, unbalancing and balancing are compared for groups of pooled machines of unequal sizes. The various FMS scenarios that are investigated are described in Section 3.1. Section 3.2 provides the computational results on the IP problem of selecting part mix ratios, which are subsequently input into the simulations. In Section 4, the theoretical, aggregate results on the optimality of unbalancing workloads (see Stecke and Solberg, 1981, 1985) are investigated using realistic, detailed simulation models of FFSs. The model is discussed in Section 4.1. Section 4.2 analyzes the computational results of the simulation studies. We offer guidelines for implementing an operating objective of unbalancing workloads to maximize production rate. Conclusions and future research needs are given in Section 5.

## 2. A flexible approach to FMS operation

In this section, we review how to both select part types and determine their mix ratios for the objectives of balancing or unbalancing using the flexible approach suggested by Stecke and Kim (1986a). We illustrate the suggested solution procedures in Section 3 using various scenarios of different workload distributions among the machine types.

The types of systems that are considered here are those that machine independent part types with varying production requirements. There is more freedom, and hence benefits, in determining the relative production ratios at which a particular part mix could be machined together.

### 2.1. Integer programming formulation

This section reviews the integer formulation to select part mix ratios for the objectives of unbalancing and balancing (see Stecke, 1985; and, Stecke and Kim, 1986a). Constraints such as due dates or tool magazine capacity are not considered here, for the purposes of this unbalancing study. Tool magazine constraints are considered in Stecke and Kim (1988).

Table 1
Notation

| | |
|---|---|
| $i$ | part types, $i=1,\dots,N$ |
| $j$ | machines, $j=1,\dots,M$ |
| $k$ | machine types, $k=1,\dots,K$ |
| $a_i$ | production ratio of part type $i$ |
| $r_i$ | production requirements for part type $i$ |
| $p_{ijk}$ | processing time of a part of type $i$ on a machine $j$ of machine type $k$ |
| $m_k$ | number of machines of type $k$ |
| $\mathrm{pw}_{jik}$ | average workload required by a part of type $i$ on a machine of type $k = p_{ijk}/m_k$ |
| $W_k$ | constant value indicating an aggregate, (un)balanced workload per machine on machine type $k$ over time |
| $x_{k1}$ | load over (un)balanced, $W_k$, on machine type $k$ |
| $x_{k2}$ | load under (un)balanced, $W_k$, on machine type $k$ |
| $C_{k1}$ | weight assigned to the potential overload ($x_{k1}$) |
| $C_{k2}$ | weight assigned to the potential underload ($x_{k2}$) |
| $f_i$ | maximum number of fixtures dedicated to part type $i$ |
| $n$ | total number of pallets in the system |

The notation is provided in Table 1. Given the aggregate production and processing time requirements of each part type on each machine type, the model for selecting aggregate part mix ratios is reviewed as the following integer formulation, Problem (P1).

(P1): Minimize

$$\sum_{k=1}^{K} C_{k1}x_{k1} + \sum_{k=1}^{K} C_{k2}x_{k2}$$

subject to

$$\sum_{i=1}^{N} \mathrm{pw}_{ik}a_i - x_{k1} + x_{k2} = W_k, \quad k=1,\dots,K, \tag{1}$$

$$a_i \leqslant f_i, \quad i=1,\dots,N, \tag{2}$$

$$a_i \geqslant 0 \quad \text{and integer}, \quad i=1,\dots,N, \tag{3}$$

$$x_{k1},\ x_{k2} \geqslant 0, \quad k=1,\dots,K. \tag{4}$$

The objective function can be changed by weighting the coefficients ($C_{k1}$ and $C_{k2}$) of the overload and underload on each machine type differently. This provides alternative sets of optimal mix ratios. Constraint (1) describes the average workload on each machine type. This enables the workload per machine to be specified as unbalanced for those systems configured as pooled machines of unequal sizes. The relative, aggregate, and unbalanced workloads per machine on each machine type are calculated using the closed queueing network model, CAN-Q (Solberg, 1977). The workloads, $W_k$, can be scaled arbitrarily. Since the processing time information is input to (P1), different scalings for the $W_k$ result in a proportional scaling for the $a_i$. Constraint (2) restricts the maximum ratio values (i.e., maximum number of parts of each type to be allowed in the system simultaneously). This could be caused by a limitation on the number of fixtures of each type as well as on the remaining production requirements. The coefficients, $C_{k1}$ and $C_{k2}$, can be selected arbitrarily. They only serve to guide the integer program to an alternative set of optimal part mix ratios.

### 2.2. Solution procedures to select part types and mix ratios using the flexible approach

In this section, we review the implementation of the flexible approach using the integer formulation (P1). The following procedure selects the subset of part types to be machined together and determines their part mix ratios over the upcoming flexible time period.

*Part type / Mix ratio Algorithm*

*Step* 1. Formulate and solve Problem (P1) for a particular set of parameters $W_k$, $C_{k1}$, $C_{k2}$.

*Step* 2. For those part types with positive ratio values in the optimal solution (i.e., $a_i \geqslant 1$), produce at those ratios until some event, such as the completion of the requirements of some part type(s) occurs.

*Step* 3. Update the part mix ratios by introducing the following constraints:

– $a_{i_1} \geqslant 1$, where $i_1 = \{$part types that have not yet completed their requirements$\}$,

– $a_{i_2} = 0$, where $i_2 = \{$part types that have completed their requirements$\}$.

*Step* 4. If all requirements for all part types are completed, STOP. Otherwise go to Step 1.

In a static problem, the algorithm is iterated over time until the requirements of all part types are completed. In usual implementation, orders would arrive over time and the same procedure is followed. At Step 2, the part types with near zero ratio values are not selected to be produced simultaneously over the upcoming time horizon. Step 3 updates the part mix as well as their ratios, if the input of one or more new part types makes the machine tools' aggregate workloads more balanced. Otherwise, only the mix ratios of the same set of part types are updated. The part types that do not complete their requirements continue production over the next horizon without cutting tool changeovers.

## 3. An experimental study of unbalancing and balancing

In this section, the part mix ratios selected for the objectives of unbalancing and balancing workloads are compared using a realistic FFS configured as pooled machines of unequal sizes. Section 3.1 describes the problem sets. In Section 3.2, computational results on finding part mix ratios are provided.

### 3.1. Scenarios investigated

The three problem sets of Table 2 are used to investigate the performances of unbalancing and balancing workloads. There are ten part types and their production requirements ordered to be produced on an FMS having pooled machines of unequal sizes. In particular, there are pooled drills and VTLs, each group having two identical machines. There is only one mill. The processing times and three different sets of production requirements for each of the ten part types are

Table 2
Processing times and production requirements for ten part types on three machine types with five machines

| Part type | Mill(1) | Drill(2) | VTL(2) | Production requirements | | |
|---|---|---|---|---|---|---|
| | | | | Problem 1 | Problem 2 | Problem 3 |
| PT1 | 10 [a] | 60 | 50 | 65 [b] | 40 | 60 |
| PT2 | 15 | 20 | 40 | 55 | 60 | 50 |
| PT3 | 40 | 10 | 30 | 20 | 30 | 20 |
| PT4 | 30 | 20 | 20 | 20 | 30 | 30 |
| PT5 | 10 | 50 | 20 | 40 | 45 | 35 |
| PT6 | 10 | 30 | 20 | 50 | 55 | 45 |
| PT7 | 20 | 10 | 10 | 20 | 15 | 15 |
| PT8 | 15 | 20 | 30 | 10 | 15 | 25 |
| PT9 | 25 | 10 | 20 | 20 | 35 | 30 |
| PT10 | 05 | 40 | 40 | 70 | 60 | 50 |

[a] Processing times are in minutes.
[b] Production requirements are in number of parts.

provided for this system of three machine types and five machines. Processing times are in minutes.

The problem sets were chosen to cover a variety of realistic scenarios. In particular, in Problem 1 of Table 2, the total average processing times ($\Sigma_i \text{pw}_{ik} \cdot r_i$, $k =$ Mill, Drill, VTL) are distributed more to the pooled drills and VTLs than to the mill. In Problem 2, the mill is more heavily loaded. Finally, the total average workloads per machine are about equally distributed in the third Problem.

One difficulty in trying to compare the results of balancing and unbalancing machine workloads for a dynamic situation over time is the following. The same numbers of the same part types with given production requirements are not produced for these objectives over the same time horizon. There is no regeneration point. In order to try to provide common bases for comparison purposes, two different methods of *selecting part types* are considered here.

For *Method* 1, the integer formulation for balancing attempts to select the *same* part types as those selected by the unbalanced problem during each run if possible. However, the sets of selected parts are usually identical only for the first run. Even then, the production ratios are not the same.

*Method* 2 to select part types is introduced to reduce the dependence of the objective function value for the *last run* in each series of runs upon the distribution of the total workloads per machine. This second method attempts to select the part types and their mix ratios with the best objective function values for both unbalancing and balancing workloads. With this method, the sets of selected parts are *not* the same. For both methods, the determined best part mix ratios for both unbalancing and balancing machine workloads are compared using simulation in Section 4.

### 3.2. *Part type / mix ratio selection for unbalancing / balancing workloads*

Initially, the number of part (pallets) in the system is fixed as seven for these three problems of Table 2. The integer programs (P1) are run using LINDO on an AMDAHL 5860. The values of parameters $W_k$, $C_{k1}$, and $C_{k2}$ are specified as 100, 1, and 1, respectively, when workloads are balanced.

For the problems of Table 2 and the system of three groups of 1, 2 and 2 machines each, the unbalanced average workloads per machine, $W_k$, that provide the maximum expected production (i.e., $[W_1 : W_2 : W_3] = [80 : 105 : 105]$ for $n = 7$—see Stecke and Solberg, 1981) are used for unbalancing in the integer Problem (P1) that provides the part mix ratios. These ratios will then unbalance aggregate workloads, as the theoretical unbalanced optimum suggests.

A sequence of Problems (P1) is re-solved over time, *as production requirements for some part types are completed* and new part types are to begin production. The following rule to prevent too frequent and unnecessary tool changeovers is used. If, after the completion of requirements of some part type(s), the total processing time required to complete the remaining requirements of any one part type is less than four hours, only the ratios of the remaining part types are updated. A new part type is *not* introduced.

#### 3.2.1. *Analysis of the IP* (P1) *results for Method* 1

First, Method 1 of selecting part types is applied to the three problems of Table 2. This method attempts to select the same part types for both the unbalancing and balancing objectives as much as possible, to try to make comparisons more straightforward.

Table 3 provides both the unbalanced and balanced part mix ratios and also demonstrates the use of the flexible approach to select part types. The unbalanced workloads, $W_k$, come from using CAN-Q. These workloads maximize expected production.

Table 3 is obtained by first running (P1) to select part types and their mix ratios, and then running a simulation to check which part type finishes its production requirements first. The finished part type is deleted, those not finished remain and (P1) is run again to see if new part types should enter the system. Then another simulation is run. The series of Problems (P1) and simulations are run until all production requirements for all part types are finished. The details of the simulation runs are described in Section 4.

The rules labeled (a) in Table 3 imply that new part types are scheduled to enter the system. These new part types are noted in boldface. The rules labeled as (b), (c), and (d) indicate that the current ratios are updated with no new part type entering,

Table 3
Integer optimum solutions using Method 1 to select part types for the objectives of unbalancing/balancing workloads when seven pallets are in the system

| | Run | Rule | Selected part types | Production ratios | Objective function value | CPU time (seconds) |
|---|---|---|---|---|---|---|
| *Problem* 1 | 1 | UB(a) [1] | **2, 5, 6, 8, 10** | 2:1:2:1:1 | 0 | 3.583 |
| | | B(a) [2] | **2, 5, 6, 8, 10** | 1:1:1:3:1 | 20 | 1.501 |
| | 2 | UB(a) | 2, 5, 6, **7**, 10 | 2:1:1:1:2 | 0 | 0.942 |
| | | (b) | 5, 6, 7, 10 | 1:2:2:2 | 25 | 0.992 |
| | | B(a) | 2, 5, 6, **7**, 10 | 2:1:2:2:1 | 10 | 2.080 |
| | 3 | UB(a) | **3**, 5, 6, 10 | 1:1:1:3 | 15 | 0.960 |
| | | (b) | 3, 5, 6 | 1:3:2 | 55 | 1.042 |
| | | B(a) | 2, **3**, 5, 6, 10 | 2:1:1:2:1 | 10 | 1.504 |
| | 4 | UB(a) | **1**, 3, **4** | 3:1:1 | 25 | 0.738 |
| | | B(a) | **1**, 2, 3, **4**, 5, 10 | 1:1:1:1:1:1 | 10 | 1.121 |
| | | (b) | 1, 3, 4, 5, 10 | 1:1:1:2:1 | 30 | 1.337 |
| | 5 | UB(a) | 1, 4, **9** | 3:1:1 | 15 | 0.636 |
| | | (b) | 1, 9 | 3:2 | 15 | 0.699 |
| | | B(a) | 1, 4, 5, **9**, 10 | 1:1:1:2:1 | 20 | 1.853 |
| | | (b) | 1, 4, 9, 10 | 2:1:2:1 | 5 | 1.244 |
| | | (c) | 1, 4, 10 | 1:3:2 | 15 | 1.722 |
| | | (d) | 1, 10 | 3:1 | 75 | 1.253 |
| *Problem* 2 | 1 | UB(a) | **2, 5, 6, 8, 10** | 2:1:2:1:1 | 0 | 3.583 |
| | | B(a) | **2, 5, 6, 8, 10** | 1:1:1:3:1 | 20 | 1.501 |
| | 2 | UB(a) | 2, 5, 6, **7**, 10 | 2:1:1:1:2 | 0 | 0.942 |
| | | B(a) | 2, 5, 6, **7**, 10 | 2:1:2:2:1 | 10 | 2.080 |
| | 3 | UB(a) | **3**, 5, 6, 10 | 1:1:1:3 | 15 | 0.960 |
| | | B(a) | 2, **3**, 5, 6, 10 | 2:1:1:2:1 | 10 | 1.504 |
| | 4 | UB(a) | **1**, 3, 5, 6 | 2:1:1:1 | 20 | 1.056 |
| | | (b) | 1, 3, 5 | 3:1:1 | 20 | 1.102 |
| | | B(a) | **1**, 2, 3, 5, 10 | 1:2:1:1:1 | 15 | 1.086 |
| | 5 | UB(a) | 1, 3, **9** | 3:1:1 | 25 | 1.337 |
| | | B(a) | 1, 3, 5, **9**, 10 | 1:1:1:1:2 | 10 | 1.153 |
| | | (b) | 1, 5, 9, 10 | 1:1:3:2 | 20 | 1.254 |
| | 6 | UB(a) | 3, **4**, 9 | 1:1:1 | 170 | 1.588 |
| | | (b) | 4, 9 | 2:1 | 160 | 1.023 |
| | | B(a) | 1, **4**, 9, 10 | 2:1:2:1 | 5 | 1.122 |
| | | (b) | 1, 4, 9 | 3:1:1 | 25 | 1.050 |
| | | (c) | 4, 9 | 2:2 | 140 | 1.536 |
| *Problem* 3 | 1 | UB(a) | **2, 5, 6, 8, 10** | 2:1:2:1:1 | 0 | 3.583 |
| | | (b) | 2, 5, 8, 10 | 1:2:2:1 | 10 | 2.090 |
| | | (c) | 2, 5, 10 | 3:2:1 | 20 | 1.600 |
| | | B(a) | **2, 5, 6, 8, 10** | 1:1:1:3:1 | 20 | 1.501 |
| | 2 | UB(a) | **1**, 5, **9**, 10 | 1:1:2:2 | 10 | 2.574 |
| | | B(a) | **1**, 2, 5, 6, **9**, 10 | 1:1:1:1:2:1 | 15 | 1.105 |
| | 3 | UB(a) | 1, **4**, 9, 10 | 1:1:1:3 | 0 | 1.781 |
| | | (b) | 1, 4, 9 | 3:1:1 | 15 | 1.029 |
| | | B(a) | 1, 2, **4**, 5, 6, 10 | 1:1:1:1:1:1 | 35 | 1.317 |
| | 4 | UB(a) | 1, 4, **7** | 3:1:1 | 15 | 1.343 |
| | | (b) | 4, 7 | 2:1 | 160 | 0.964 |
| | | B(a) | 1, 2, 4, 6, **7**, 10 | 1:1:1:2:1:1 | 5 | 1.443 |

Table 3 (continued)

| Run | Rule | Selected part types | Production ratios | Objective function value | CPU time (seconds) |
|---|---|---|---|---|---|
| 5 | UB(a) | **3**, 4, 7 | 1:1:1 | 170 | 1.037 |
| | (b) | 3, 4 | 1:1 | 180 | 1.341 |
| | B(a) | 1, 2, **3**, 4, 7, 10 | 1:1:1:1:1:2 | 40 | 1.317 |
| | (b) | 1, 2, 3, 4, 7 | 2:1:1:1:1 | 35 | 1.104 |
| | (c) | 1, 3, 4 | 3:1:1 | 5 | 1.106 |
| | (d) | 1, 3 | 2:2 | 60 | 0.524 |

Bold face indicates the new part types to be introduced for the upcoming run.
(1) UB refers to the unbalanced integer Problem (P1) specifying that $W_{mill} = 80$, $W_{drill} = 105$, and $W_{vtl} = 105$.
(2) B refers to the balanced integer Problem (P1) specifying that $W_{mill} = 100$, $W_{drill} = 100$, and $W_{vtl} = 100$.

even though the requirements of some part type(s) have been completed.

The problems considered here are static (i.e., orders are not arriving). A series of problems is solved, until all requirements of all part types are completed. The objective function value of Problem (P1) for the last of each series of runs depends on the fixed distribution of the total workloads per machine among the three machine types. For the last run, there is no choice as to which part types to select. The last of each set of runs is, hence, not representative of the performance of the approaches. These ending conditions bias the results. The more usual situation where the part type selection approaches are applicable is dynamic, as production orders arrive and the finished orders leave. In typical production, new orders would continuously arrive and be considered for input into the system.

For example, in Problem 2 of Table 3, the sixth (last) objective function values are very large. This is because the remaining workload to finish all requirements of all ten part types is much higher on the unpooled mill. This results in large overload values on the mill. For the third Problem, the fifth objective function values are large for the unbalancing objective. This is because the total workloads per machine in Problem 3 were selected to be *balanced* about equally on the three machine types. This bias in the last run would tend to not occur in the more typical situation, in which random orders arrive.

The following additional observations about selecting part types and mix ratios can be made from Table 3, using Method 1 to select part types followed by a simulation run.

(1) The completion of each simulation run can lead to a partial tool changeover (changing only a few cutting tools). There are four partial tool loadings in both Problems 1 and 3, and five partial changeovers in Problem 2.

(2) For both the balancing and unbalancing runs, most solutions to Problem (P1) suggest various combinations of 3–5 part types that are compatible for immediate and simultaneous machining.

(3) Setting $W = 100$ (a low number) in Problem (P1) keeps the production ratio values small enough to be directly useful in realistic situations where there are limited numbers of pallets and dedicated fixtures. These low ratios values are useful to help solve subsequent scheduling problems, such as determining a good part input sequence (see Stecke, 1985). The summation of the ratios for each run of Table 3 is always less than nine. If $W = 1000$ were used, for example, the sums of the ratio values would all be a bit less than 90, which is too large (and unnecessary) to work with.

(4) The objective function values often tend to get larger with the number of runs. This is because the problems considered here are static, having fixed orders with no new arrivals. In the more typical dynamic situation of orders arriving to the FMS continuously, a better objective function value can be anticipated. With fresh orders arriving, there would usually be a better opportunity to balance or unbalance workloads.

(5) All CPU times are less than four seconds. The balanced problems have shorter CPU times than the unbalanced. This is because for the balanced integer Problem (P1), the ratio values of those part types not selected by the previous unbalancing Problem (P1) are now set equal to zero. This reduces the size of the balanced Problem (P1), which reduces the CPU time. The part

Table 4
Integer optimum solutions using Method 2 to select part types for the objectives of unbalancing/balancing workloads for Problems 1, 2 and 3

| $n$ | rule | selected part types | production ratios | Objective function value | CPU time (seconds) |
|---|---|---|---|---|---|
| 6 | UB [a] | 1, 7, 8, 10 | 2:1:2:1 | 3 | 5.216 |
| | B [b] | 1, 3, 4, 6, 10 | 1:1:1:1:2 | 0 | 4.555 |
| 7 | UB [c] | 2, 5, 6, 7, 10 | 2:1:1:1:2 | 0 | 4.493 |
| 8, 9 | UB [d] | 2, 5, 7, 9, 10 | 1:1:1:1:3 | 3 | 6.516 |
| 10, 11 | UB [e] | 5, 8, 9, 10 | 1:1:2:3 | 6 | 10.688 |
| 12, 13 | UB [f] | 1, 4, 6, 8 | 2:1:1:2 | 5 | 7.096 |

[a] Specifies $W_{mill} = 76$, $W_{drill} = 106$, $W_{vtl} = 106$.
[b] Balanced mix ratios are the same for $n = 6, \ldots, 13$.
[c] Specifies $W_{mill} = 80$, $W_{drill} = 105$, $W_{vtl} = 105$.
[d] Specifies $W_{mill} = 84$, $W_{drill} = 104$, $W_{vtl} = 104$.
[e] Specifies $W_{mill} = 88$, $W_{drill} = 103$, $W_{vtl} = 103$.
[f] Specifies $W_{mill} = 90$, $W_{drill} = 102.5$, $W_{vtl} = 102.5$.

mix ratios for the balancing problem have zero objective function values.

The theoretical, aggregate unbalanced workloads have been used as input into models used to solve the problems of selecting part types and determining their production ratios. The performance of these unbalanced ratios are compared to the balanced ratios in Section 4.2.

### 3.2.2. *Analysis of the IP* (P1) *results for Method* 2

We now use Method 2 to select part types. Problems (P1) are run again using the processing time data of Table 2. Here, we are only *demonstrating part type selection* (for both balancing and unbalancing) *for the first run only*. Hence the production requirements are not considered and no simulations are performed.

Table 4 presents part mix ratios using Method 2, which selects part types with the best objective function values for both the unbalanced and balanced Problems (P1) for a given number of pallets in the system [$n = 6$, 7, 8, 9, 10, 11, 12 and 13]. The theoretical unbalanced optimal workloads provided in Stecke and Solberg (1981) are used to select part types and determine their mix ratios. The unbalanced part mix ratios are different for each value of $n$, the number of pallets in the system.

For the balancing objective, the *same* part types are always selected in the *same* ratios, for all values of $n$. This is because the workload parameter, $W$, is never changed. $W$ is always 100, for each machine type. The following observations can be made from Table 4.

(1) The unbalanced problems (P1) usually have longer CPU times. This is because processing times are not scaled similar to the theoretical unbalanced optimal average workloads. All CPU times to solve Problem (P1) are under 11 seconds, for these three problems.

(2) All solutions of (P1) for both unbalancing and balancing workloads suggest various combinations of 3–5 part types that are compatible for subsequent simultaneous machining.

(3) Although the unbalanced workloads change only slightly as $n$ increases, the selected part types and their production ratios are quite different. However, for each run, these are just one of the many optimal sets of ratios.

There is no discernible advantage to using Method 2 instead of Method 1, since different part types and ratios are selected by both methods. Method 1 had been perceived to favor unbal-

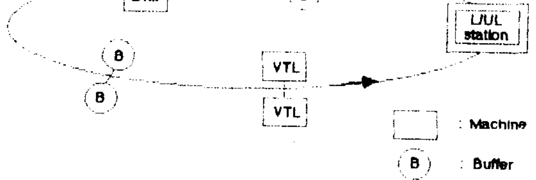


Figure 1. System configuration

ancing when selecting part types. However, no differences were observed.

## 4. Simulated studies

In this section, we compare unbalancing to balancing workloads using a realistic simulation model of an FFS of groups of pooled machines of unequal sizes. Section 4.1 outlines the various parameters of the simulation model. In Section 4.2, the simulation results are analyzed.

### *4.1. The model*

The simulation model of the FFS is developed in GPSS/H. The FMS configuration is provided in Figure 1. It is an FFS having uni-directional transportation. There are five machines: one mill, two pooled drills, and two pooled VTLs. There are three buffer spaces: one after the mill and before the drills, and two in between the drills and lathes. There are five identical load/unload stations and five transporters. Travel times are one minute between all links, i.e., between L/UL–mill; mill–drill; buffer–drill; drill–buffer; buffer–VTL; VTL–L/UL.

In the simulation model, only seven parts in total can be in the system. There are two cases of fixture limitations: First, the number of fixtures of each type is limited for each part type to be four ($f_i = 4$, $i = 1, \ldots, 10$). The second case requires no restriction on this value ($f_i < \infty$, $i = 1, \ldots, 10$).

A fixed number of parts (representing the pallet limitations) of mixed types having nonzero production ratio values is always in the system. Whenever the production requirements of some part type(s) are finished, a current simulation run is usually terminated. When one or more new part types are selected to be input into the system, new ratios are found to begin the next simulation run. Otherwise, if no new part type is to enter, the current simulation run continues. However, new *optimal* production ratios are found for the reduced set of part types.

There was no precise algorithm to be found in the literature that finds a good part input sequence into a flow shop having pooled machines. The part input sequence into the FFS here is determined by using a combination of a modified Johnson's algorithm and the calculated part mix ratios (see Stecke and Kim, 1986b). This algorithm appears reasonable. This sequencing method is followed because, on average and over time, it will provide the desired unbalanced or balanced workloads. However, further research is required to determine a good part input sequence.

For the simulations, the priorities of the part types of Table 2 and Problem 1 are calculated using a modified Johnson's algorithm: 10, 2, 6, 8, 5, 1, 4, 3, 9 and 7. Then, when the part mix ratios of part types 5, 6, 7 and 10 are 1 : 2 : 2 : 2 (Run 2 UB(b) of Table 3), for example, the part input sequence is 10, 10, 6, 6, 5, 7, 7. The input sequence is followed, regardless of which type of part just left the system.

In one related study, simulation experiments by Schriber and Stecke (1988) are conducted to investigate the performance of aggregate part mix ratios after an FFS has reached steady state. The production requirements of part types are not considered. Their results show that the system utilization and production rate achieved using the unbalanced mix ratios are about 6 to 10 percent higher than those achieved by simply selecting all part types for concurrent production, in ratios proportional to the requirements. This is the usual approach that many studies take.

### *4.2. The simulation results*

In this section, simulation results are presented to investigate unbalancing and balancing using both Methods 1 and 2. *Processing* (*transportation, blocking*) *utilizations* are found for each machine type. These indicate the proportions of total processing (transportation, blocking) times to total makespan. *Processing utilization* is calculated as the ratio of total actual machining time to total makespan, for each machine type: Mill, Drill and VTL. *Machine utilization* is expressed as the sum of processing, transportation, and blocking utilizations, for each machine type. *System utilization* is a weighted average of the *processing* utilizations of the three machine types and is a measure of overall system usage. System utilization is equal to the sum of the Mill processing utilization, twice the Drill processing utilization, and twice the VTL processing utilization, and divided by five.

The machine and system utilizations in all of the subsequent Figures are average values. These are cumulative utilizations and calculated as re-

Table 5
Simulation results using Method 1 after the completion of all production requirements of all ten part types for Problem 1

| Comparison | Four fixtures | | No limitations | |
|---|---|---|---|---|
| | UB | B | UB | B |
| Makespan (minutes) | **7054** | 7436 | **7044** | 7419 |
| Mill utilization | **0.927** | 0.883 | 0.948 | **0.953** |
| **Processing utilization** | **0.734** | 0.695 | **0.734** | 0.697 |
| Transportation utilization | 0.072 | 0.069 | 0.072 | 0.070 |
| Blocking utilization | 0.121 | 0.119 | 0.142 | 0.186 |
| Drill utilization | **0.916** | 0.871 | **0.918** | 0.873 |
| **Processing utilization** | **0.886** | 0.840 | **0.887** | 0.842 |
| Transportation utilization | 0.030 | 0.031 | 0.030 | 0.031 |
| Blocking utilization | 0.000 | 0.000 | 0.001 | 0.000 |
| VTL utilization | **0.887** | 0.849 | **0.888** | 0.853 |
| **Processing utilization** | **0.847** | 0.803 | **0.848** | 0.805 |
| Transportation utilization | 0.040 | 0.046 | 0.040 | 0.048 |
| Blocking utilization | 0.000 | 0.000 | 0.000 | 0.000 |
| **System utilization** | **0.840** | 0.796 | **0.841** | 0.798 |
| Average buffer utilization | 0.340 | 0.205 | 0.359 | 0.218 |
| Cart utilization | 0.060 | 0.057 | 0.059 | 0.058 |
| Number of dedicated fixtures | 30 | 31 | 35 | 37 |
| CPU time (seconds) | 3.182 | 2.442 | 2.325 | 2.265 |

Table 6
Simulation results using Method 1 after the completion of all production requirements of all ten part types for Problem 2

| Comparison | Four Fixtures [a] | | No limitations [a] | |
|---|---|---|---|---|
| | UB | B | UB | B |
| Makespan (minutes) | 8090 | **7533** | 8090 | **7524** |
| Mill utilization | 0.921 (0.898) | **0.939 (0.938)** | 0.932 (0.913) | **0.946 (0.938)** |
| **Processing utilization** | 0.754 (0.682) | **0.809 (0.791)** | 0.754 (0.682) | **0.810 (0.791)** |
| Transportation utilization | 0.059 (0.069) | 0.071 (0.079) | 0.060 (0.070) | 0.073 (0.079) |
| Blocking utilization | 0.108 (0.147) | 0.059 (0.068) | 0.118 (0.161) | 0.063 (0.068) |
| Drill utilization | 0.743 **(0.922)** | **0.801** (0.850) | 0.743 **(0.923)** | **0.803** (0.850) |
| **Processing utilization** | 0.716 **(0.892)** | **0.769** (0.815) | 0.716 **(0.892)** | **0.770** (0.815) |
| Transportation utilization | 0.027 (0.030) | 0.032 (0.035) | 0.027 (0.031) | 0.032 (0.035) |
| Blocking utilization | 0.000 (0.000) | 0.000 (0.000) | 0.000 (0.000) | 0.001 (0.000) |
| VTL utilization | 0.800 **(0.889)** | **0.822** (0.870) | 0.839 **(0.890)** | **0.841** (0.870) |
| **Processing utilization** | 0.716 **(0.850)** | **0.769** (0.821) | 0.716 **(0.850)** | **0.770** (0.821) |
| Transportation utilization | 0.038 (0.039) | 0.047 (0.049) | 0.038 (0.040) | 0.047 (0.049) |
| Blocking utilization | 0.046 (0.000) | 0.006 (0.000) | 0.085 (0.000) | 0.024 (0.000 |
| **System utilization** | 0.724 **(0.833)** | **0.777** (0.813) | 0.724 **(0.833)** | **0.778** (0.813) |
| Average buffer utilization | 0.271 (0.368) | 0.190 (0.195) | 0.288 (0.393) | 0.200 (0.195) |
| Cart utilization | 0.052 (0.059) | 0.059 (0.065) | 0.052 (0.060) | 0.060 (0.065) |
| Number of dedicated fixtures | 31 | 33 | 40 | 40 |
| CPU time (seconds) | 2.914 | 2.840 | 2.956 | 2.949 |

[a] ( ) indicates cumulative utilizations minus the last run.

Table 7
Simulation results using Method 1 after the completion of all production requirements of all ten part types for Problem 3

| Comparison | Four fixtures [a] | | No limitations [a] | |
|---|---|---|---|---|
| | UB | B | UB | B |
| Makespan (minutes) | 7338 | **6921** | 7350 | **6921** |
| Mill utilization | 0.925 (0.915) | **0.935 (0.920)** | **0.941 (0.936)** | 0.935 (0.920) |
| **Processing utilization** | 0.752 (0.710) | **0.798 (0.753)** | 0.751 (0.709) | **0.798 (0.753)** |
| Transportation utilization | 0.057 (0.0621) | 0.065 (0.070) | 0.057 (0.063) | 0.065 (0.070) |
| Blocking utilization | 0.116 (0.143) | 0.072 (0.097) | 0.133 (0.164) | 0.072 (0.097) |
| Drill utilization | 0.807 (**0.947**) | **0.857** (0.920) | 0.806 (**0.946**) | **0.857** (0.920) |
| **Processing utilization** | 0.780 (**0.917**) | **0.827** (0.888) | 0.778 (**0.915**) | **0.827** (0.888) |
| Transportation utilization | 0.026 (0.029) | 0.030 (0.032) | 0.027 (0.029) | 0.030 (0.032) |
| Blocking utilization | 0.001 (0.001) | 0.000 (0.000) | 0.001 (0.002) | 0.000 (0.000) |
| VTL utilization | 0.842 (**0.909**) | **0.860** (0.887) | 0.857 (**0.907**) | **0.860** (0.887) |
| **Processing utilization** | 0.769 (**0.871**) | **0.816** (0.840) | 0.768 (**0.869**) | **0.816** (0.840) |
| Transportation utilization | 0.036 (0.038) | 0.044 (0.047) | 0.036 (0.038) | 0.044 (0.047) |
| Blocking utilization | 0.037 (0.000) | 0.000 (0.000) | 0.052 (0.000) | 0.000 (0.000) |
| **System utilization** | 0.770 (**0.857**) | **0.817** (0.842) | 0.769 (**0.855**) | **0.817** (0.842) |
| Average buffer utilization | 0.325 (0.399) | 0.295 (0.338) | 0.338 (0.414) | 0.295 (0.338) |
| Cart utilization | 0.052 (0.058) | 0.059 (0.063) | 0.053 (0.058) | 0.059 (0.063) |
| Number of dedicated fixtures | 35 | 29 | 42 | 30 |
| CPU time (seconds) | 2.542 | 2.670 | 2.946 | 2.458 |

[a] ( ) indicates cumulative utilizations minus the last run.

quirements are completed after each simulation run. The difference between machine and system utilizations provides the average amount of time spent in transportation and blocked. The *all machines utilization* is calculated as the sum of the Mill machine utilization, twice the Drill machine utilization and twice the VTL machine utilization, and divided by five.

#### 4.2.1. *Analysis of the simulation results for Method 1*

The simulation studies are performed using Method 1 for two cases. One case allows only four fixtures of each type. The second has no fixture limitations. This second case is considered in order to examine the number of fixtures of each type that would be required to finish all requirements of all part types if enough were available. The number of pallets in the system is fixed as seven for Problems 1, 2, and 3 of Table 2. The ratios provided in Table 3 are used in the series of simulations.

Tables 5, 6 and 7 provide simulation results for the three Problems on the machine, processing,

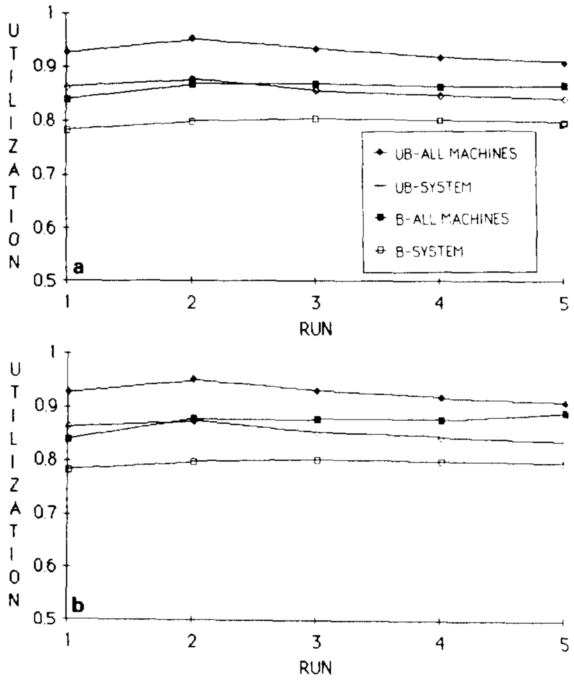


Figure 2. Cumulative utilizations of the unbalancing and balancing objectives for Problem 1. (a) The number of fixtures of each type is limited to be four. (b) No fixture limitations

and system utilizations as well as makespan. The higher utilizations and lower makespans are noted in boldface. Figures 2–4 also show the *cumulative* machine and system utilizations for each of the distinct simulation runs required to finish the requirements of all part types for the two cases, with and without fixture limitations. Tables 6 and 7 provide the average utilizations both for all runs *and* for all runs except for the last run.

The following observations can be made from the results from Tables 5–7 and Figures 2–4. Both of the utilization measures (system and machine) are better when unbalancing than when balancing, for Problem 1 (see Table 5). For example, the system utilization is 5.5% better than when balancing.

For Problems 2 and 3, the cumulative system utilization for the last run of each unbalancing problem is lower than balancing because of the ending conditions of finishing all requirements for all part types. See Tables 6 and 7 and Figures 3 and 4. This would not happen in dynamic situations. A particular reason for the lower utilizations for unbalancing for the last run in these four problems is because the total workloads per machine are distributed equally or more to the mill. This results in worse optimal objective values for the last run (of objectives) when solving Problem (P1) to select part types and determine their mix ratios. The remaining requirements have to be finished. The results of the last run are not representative of the typical FMS operating mode, where the approach to FMS operation used here would be appropriate.

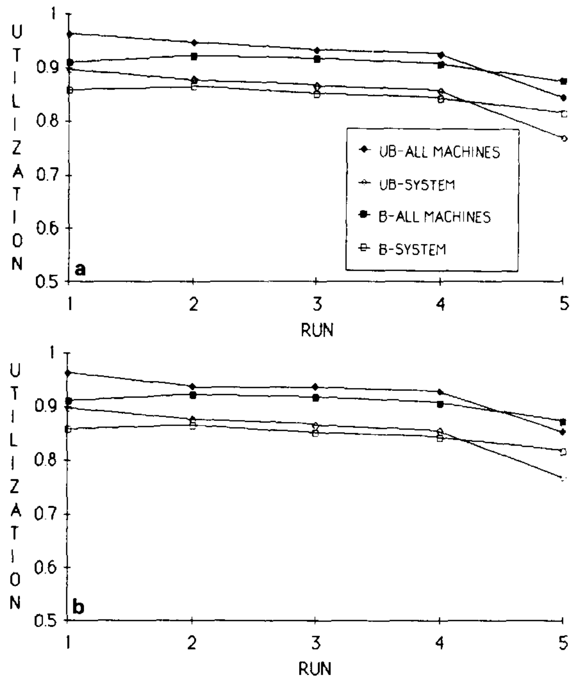


Figure 4. Cumulative utilizations of the unbalancing and balancing objectives for Problem 3. (a) The number of fixtures of each type is limited to be four. (b) No fixture limitations

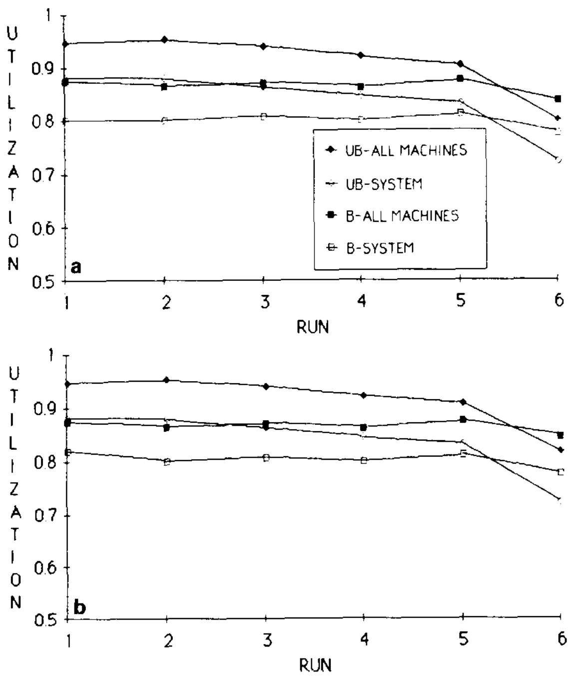


Figure 3. Cumulative utilizations of the unbalancing and balancing objectives for Problem 2. (a) The number of fixtures of each type is limited to be four. (b) No fixture limitations

However, for Problems 2 and 3, Tables 6 and 7 also provide the cumulative utilizations while *excluding* the last run. These utilizations are more representative of the actual operating situation, as the ending conditions are now excluded. These results, in conjunction with the associated Figures, all show unbalancing to be consistently better, *until* the last run forces completion of all requirements.

The amount of blocking for the mill is usually larger when unbalancing than when balancing (except when there are no required fixture limitations for Problem 1). For example, see Table 5.

The amount of blocking for the drills and VTLs as well as the number of dedicated fixtures required in the unbalanced situations are similar to

Table 8
Simulation (50-hours) results using Method 2 for unbalancing/balancing objectives for Problems 1, 2 and 3

| Comparison | $n=6$ | | $n=7$ | | $n=8$ | | $n=9$ | |
|---|---|---|---|---|---|---|---|---|
| | UB | B | UB | B | UB | B | UB | B |
| Mill utilization | 0.840 | **1.00** | 0.987 | **1.00** | 0.999 | **1.00** | **1.00** | **1.00** |
| **Processing utilization** | 0.668 | **0.833** | 0.742 | **0.833** | 0.760 | **0.834** | 0.755 | **0.835** |
| Transportation utilization | 0.071 | 0.066 | 0.067 | 0.066 | 0.082 | 0.067 | 0.107 | 0.053 |
| Blocking utilization | 0.101 | 0.101 | 0.178 | 0.101 | 0.157 | 0.099 | 0.138 | 0.112 |
| Drill utilization | **0.959** | 0.863 | **0.989** | 0.863 | **0.968** | 0.864 | **0.972** | 0.858 |
| **Processing utilization** | **0.924** | 0.831 | **0.957** | 0.831 | **0.929** | 0.831 | **0.923** | 0.832 |
| Transportation utilization | 0.035 | 0.032 | 0.032 | 0.032 | 0.035 | 0.033 | 0.044 | 0.026 |
| Blocking utilization | 0.000 | 0.000 | 0.000 | 0.000 | 0.004 | 0.000 | 0.005 | 0.000 |
| VTL utilization | **0.944** | 0.874 | **0.980** | 0.874 | **0.963** | 0.874 | **0.962** | 0.912 |
| **Processing utilization** | **0.905** | 0.817 | **0.946** | 0.817 | **0.915** | 0.818 | **0.908** | 0.818 |
| Transportation | 0.039 | 0.057 | 0.034 | 0.057 | 0.048 | 0.056 | 0.054 | 0.041 |
| Blocking utilization | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.053 |
| **System utilization** | **0.865** | 0.826 | **0.910** | 0.826 | **0.890** | 0.826 | **0.883** | 0.827 |
| Average buffer utilization | 0.302 | 0.113 | 0.446 | 0.214 | 0.404 | 0.117 | 0.424 | 0.136 |
| Cart utilization | 0.061 | 0.057 | 0.064 | 0.057 | 0.071 | 0.057 | 0.080 | 0.050 |
| Number of dedicated fixtures | 8 | 7 | 9 | 9 | 11 | 12 | 13 | 12 |
| CPU time (seconds) | 1.465 | 1.490 | 1.571 | 1.393 | 1.592 | 1.500 | 1.565 | 1.544 |
| | $n=10$ | | $n=11$ | | $n=12$ | | $n=13$ | |
| | UB | B | UB | B | UB | B | UB | B |
| Mill utilization | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** |
| **Processing utilization** | 0.767 | **0.823** | 0.762 | **0.835** | 0.705 | **0.822** | **0.047** | 0.045 |
| Transportation utilization | 0.093 | 0.065 | 0.102 | 0.056 | 0.099 | 0.088 | 0.005 | 0.005 |
| Blocking utilization | 0.140 | 0.112 | 0.136 | 0.109 | 0.196 | 0.090 | 0.948 | 0.950 |
| Drill utilization | **0.934** | 0.858 | **0.938** | 0.892 | 0.990 | **0.991** | 0.990 | **0.991** |
| **Processing utilization** | **0.896** | 0.827 | **0.892** | 0.832 | 0.806 | **0.823** | **0.043** | 0.041 |
| Transportation utilization | 0.038 | 0.031 | 0.041 | 0.040 | 0.023 | 0.024 | 0.001 | 0.001 |
| Blocking utilization | 0.000 | 0.000 | 0.005 | 0.020 | 0.161 | 0.144 | 0.946 | 0.949 |
| VTL utilization | 0.941 | **0.972** | **0.979** | **0.979** | **0.981** | 0.979 | **0.981** | 0.979 |
| **Processing utilization** | **0.878** | 0.808 | **0.873** | 0.809 | 0.749 | **0.794** | 0.010 | **0.016** |
| Transportation | 0.055 | 0.033 | 0.045 | 0.030 | 0.035 | 0.025 | 0.001 | 0.001 |
| Blocking utilization | 0.008 | 0.131 | 0.061 | 0.140 | 0.197 | 0.160 | 0.970 | 0.962 |
| **System utilization** | **0.863** | 0.819 | **0.858** | 0.823 | 0.763 | **0.811** | 0.031 | **0.032** |
| Average buffer utilization | 0.360 | 0.282 | 0.522 | 0.490 | 0.758 | 0.725 | 0.961 | 0.960 |
| Cart utilization | 0.075 | 0.056 | 0.077 | 0.058 | 0.060 | 0.056 | 0.002 | 0.002 |
| Number of dedicated fixtures | 14 | 12 | 11 | 11 | 16 | 12 | 14 | 13 |
| CPU time (seconds) | 1.576 | 1.744 | 1.643 | 1.864 | 0.961 | 1.556 | 0.966 | 1.626 |

those required by the balanced. For example, see Table 5.

The utilizations for unbalancing workloads decrease quicker with the number of runs in Problems 2 and 3, than in Problem 1. For example, see Figures 2 and 3. This is because the workloads are distributed more to the pooled drills and VTLs in Problem 1. This allows the optimal objective function value for unbalancing to be maintained better until the last run.

We can conclude that for the problems presented here, unbalancing workloads results in higher overall utilizations than balancing. All of the Figures showed unbalancing to be better than balancing until the last run. These *last run* ending conditions would not occur in reality, as orders would continuously arrive to the system. Other simulated examples show similar results (see Stecke and Kim, 1986b).

### 4.2.2. *Analysis of the simulation results for Method 2*

Method 2 attempts to select part types and their mix ratios with the best objective function values of unbalancing and balancing workloads for a given number of pallets, $n$. Simulation runs are performed again using Method 2 for a variety of $n = 6$, 7, 8, 9, 10, 11, 12 and 13 and for 50 simulated hours. The ratios found in Table 6 are used in the simulations. The production requirements are not considered here because part mix ratios are determined for the first run only. These simulations are run to steady state, similar to the Schriber and Stecke (1988) study.

Table 8 provides simulation results for the unbalancing and balancing objectives when there are no fixture limitations. For both the unbalancing and balancing rules, Figure 5 also shows the machine and processing utilizations as well as their standard deviations. The following observations can be made from the results in Table 8 and Figure 5.

The processing utilizations for the drills and VTLs are always better when unbalancing than when balancing, until the system becomes saturated with 11 pallets.

For $n = 10$ of Table 8, the unbalanced problem results in less VTL machine utilization than the balanced, but has more processing utilization. This indicates that the higher machine utilization from balancing results from more blocking.

The *machine utilizations* for the balancing objective are unbalanced among the three machine types. But unbalancing workloads leads to balanced machine utilizations among the three machine types pooled unequally. This is mainly because the pooled identical machines with more workloads share the total transportation time required by finishing all requirements for all part types.

The *processing utilizations* are in general more balanced for the balancing objective. The *system*

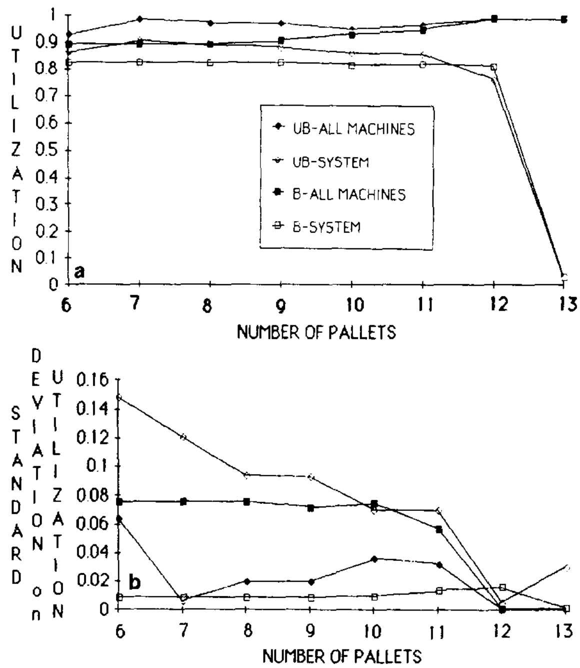


Figure 5. No fixture limitations for Problems 1, 2 and 3 using Method 2. (a) Utilizations of the unbalancing and balancing objectives. (b) Cumulative comparison of degrees of balance on utilizations

*utilization* is *better* when *unbalancing*, for six to eleven pallets in the system.

When unbalancing, the processing utilizations of the pooled drills and VTLs with more workloads tend to increase, then decrease with the number of pallets in the system after a particular saturation point is reached.

The overall *best system utilization* occurs when there are seven pallets in the system for Problems 1–3 (see Table 8). For the unbalancing objective, performance deteriorates as more pallets are added. For balancing, the results are almost the same for 6, 7, 8, 9, 10, and 11 pallets.

It can be seen in Figure 5 and Table 8 that the average *machine* and *system utilizations* are *less sensitive* to the number of pallets when balancing than when unbalancing. This implies that the appropriate number of pallets in the system should be determined in advance for a given system, in order to maximize system utilization.

The amount of blocking as well as the number of dedicated fixtures required in the unbalanced situations are similar to those required by the balanced.

Thirteen pallets are the maximum number that can be in the system. This includes the five machine tools, three buffers, and five load/unload stations. For this maximum number physically possible, the processing utilizations are almost zero. This is because the system has *deadlocked* and nothing can move. Most of the machine utilization consists of blocking. This does happen in practice, and policies to prevent deadlock need to be determined.

Therefore, it can be concluded that for these examples, the overall *system utilization* has always been *better* (except when the ending conditions are considered) when *unbalancing* the assigned machine workloads. Some of the observations here are particular to these problem sets while most are general. The tendencies demonstrated here are validated in other simulation studies. These other, similar observations can be found in Stecke and Kim (1986b).

## 5. Conclusions

This paper investigates the appropriateness of unbalancing the workloads per machine for a system having groups of pooled machines of unequal sizes. Also, this paper shows how well the aggregate and theoretically optimal unbalanced workloads perform in a realistic FMS configuration. This paper also demonstrates how these unbalanced workloads can be applied in part type selection and production ratio problems.

The simulation studies of the FFS show that the overall system utilization is better when unbalancing. It is also observed that unbalancing part mix ratios conversely leads to balanced machine utilizations among machine types pooled unequally. This is mainly because the total transportation times are shared by the identical machines of each group.

In order to maximize system utilization or production rate, the appropriate number of pallets should be examined for a given system in advance of either unbalancing or balancing. This is in part because deadlock can occur if too many pallets are in the system. More importantly, system utilization seems to be sensitive to the number of pallets in the system, especially when unbalancing (for example, see Figure 5). Finally, it can be concluded that for the variety of situations examined (here and in Stecke and Kim, 1986b), unbalancing workloads performs better than balancing for systems of pooled machines of unequal sizes, at least until the ending conditions are considered.

There are further research needs along these lines. The studies reported here are for a flexible flow line type of system. However, unbalancing workloads should be even more appropriate for more general FMSs. There are more advantages to pooling in a job shop type of FMS, where alternative routes are available. System flexibility is increased by producing a part mix, when the parts have alternative routes. Pooling increases this system flexibility and reduces the frequency of blocking. It also helps cope well with machine breakdowns. Unbalancing needs to be further investigated in these situations.

Similar studies should be done in a more dynamic situation, for example, when there are often changes in production orders or random machine failures. Implementation of the results here in the more general situations is being developed. Also, the use of unbalancing workloads for the subsequent FMS planning and operating problems should be examined. For example, further research is required to determine a good part input sequence.

## Acknowledgement

Kathy Stecke would like to acknowledge partial support from a Summer Research Grant from the Graduate School of Business Administration of The University of Michigan.

## References

Berrada, M., and Stecke, K.E. (1986), "A branch and bound approach for machine load balancing in flexible manufacturing systems", *Management Science* 32 (10), 1316–1335.

Buzacott, J.A., and Shanthikumar, J.G. (1985), "On approximate queueing models of dynamic job shops", *Management Science* 31 (7), 870–887.

Dubois, D. (1983), "A mathematical model of a flexible manufacturing system with limited in-process inventory", *European Journal of Operational Research* 14 (1), 66–78.

Hwang, S. (1986), "Part selection problems in flexible manufacturing systems planning stage", in: *Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*, K.E.

Stecke and R. Suri, (Eds.), Elsevier Science Publishers, Amsterdam, 297–309.

Lin, L.-E., S., and Lu, C.Y.J. (1984), "The scheduling problem in random flexible manufacturing systems", *Proceedings of the First ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*, in: K.E. Stecke and R. Suri, (Eds.), 278–283.

Rajagopalan, S. (1986), "Formulation and heuristic solutions for parts grouping and tool loading in flexible manufacturing systems", *Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*, K.E. Stecke and R. Suri (Eds.), Ann Arbor, MI, Elsevier Science Publishers, Amsterdam, 311–320.

Shanker, K., and Tzen, Y.-J.J. (1985), "A loading and dispatching problem in a random flexible manufacturing system", *International Journal of Production Research* 23 (3) 579–595.

Schrage, L.E. (1981), *Linear Programming Models with LINDO*, The Scientific Press, Palo Alto, CA.

Schriber, T.J., and Stecke, K.E. (1988), "Machine utilizations achieved using balanced FMS production ratios in a simulated setting", *Annals of Operations Research* 15.

Solberg, J.J. (1977), "A mathematical model of a computerized manufacturing system", *4th Internal Conference on Production Research*, Tokyo, Japan, 22–30.

Stecke, K.E. (1983), "Formulation and solution of nonlinear integer production planning problems for flexible manufacturing systems", *Management Science* 29 (3) 273–288.

Stecke, K.E. (1985), "Procedures to determine both appropriate production ratios and minimum inventory requirements to maintain these ratios in flexible manufacturing systems", Working paper no. 448, Graduate School of Business Administration, The University of Michigan, Ann Arbor, MI.

Stecke, K.E., and Kim, I. (1986a), "A flexible approach to implementing the short-term FMS planning function", *Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*, K.E. Stecke and R. Suri (Eds.), Ann Arbor, MI, Elsevier Science Publishers, Amsterdam, 283–295.

Stecke, K.E., and Kim, I. (1986b). "Decision aids for part type selection using aggregate production ratios to study pooled machines of unequal sizes", Working paper no. 478, Graduate School of Business Administration, The University of Michigan, Ann Arbor, MI.

Stecke, K.E., and Kim, I. (1988), "A study of FMS part type selection approaches for short-term production planning", *International Journal of Flexible Manufacturing Systems* 1 (1) 7–29.

Stecke, K.E., and Solberg, J.J. (1981), "The CMS loading problem", Report no. 20, School of Industrial Engineering, Purdue University, W. Lafayette, IN.

Stecke, K.E., and Solberg, J.J. (1985), "The optimality of unbalancing both workloads and machine group sizes in closed queueing networks of multiserver queues", *Operations Research* 33 (4) 882–910.

Suri, R. (1983), "Robustness of queueing network formulas", *Journal of the Association for Computing Machinery* 30 (3) 564–594.

Suri, R., and Diehl, G.W. (1985), "MANUPLAN: A precursor to simulation for complex manufacturing systems", *Proceedings of the Winter Simulation Conference*, San Francisco CA, 411–420.

Talbot, F.B., Patterson, J.H., and Gehrlein, W.V. (1986), "Comparative evaluation of heuristic line balancing techniques", *Management Science* 32 (4) 430–454.

Wee, T.S., and Magazine, M.J. (1981), "An efficient branch and bound algorithm for assembly line balancing—Part 2: Maximize the production rate", Working paper no. 151, University of Waterloo, Waterloo, Ontario.

Whitney, C.K., and Gaul, T.S. (1985), "Sequential decision procedures for batching and balancing in FMSs", *Annals of Operations Research* 3, 301–316.