DETERMINING TRANSFER BATCH SIZES IN TRIP-BASED MATERIAL HANDLING SYSTEMS

Yavuz A. Bozer
and
Jonghwa Kim
Department of Industrial and Operations Engineering
The University of Michigan
Ann Arbor, MI 48109-2117

Technical Report 94-6

March 1994
Revised November 1994

DETERMINING TRANSFER BATCH SIZES IN TRIP-BASED MATERIAL HANDLING SYSTEMS

Yavuz A. Bozer

Jonghwa Kim

Department of
Industrial and Operations Engineering
The University of Michigan
Ann Arbor, MI 48109-2117, USA

ABSTRACT

Trip-based material handling systems such as AGV systems, lift trucks, etc. are often designed with a given flow matrix (or FROM-TO chart) which is usually treated as the number of loaded trips that the devices must perform per unit time between the workstations. In reality, the number of trips that would result from parts flow in a facility is dictated by the transfer batch size, i.e., the number of parts that are transferred from one workstation to the next in one trip. In this paper, we present analytical and simulation results aimed at determining optimal or near-optimal transfer batch sizes in manufacturing systems.

1 INTRODUCTION

Trip-based material handling systems consist of one or more handling devices that are self-powered and operate independently from each other [18]. Examples of trip-based handling systems used in manufacturing include Unit Load Automated Guided Vehicle (AGV) systems, lift trucks, microload Automated Storage/Retrieval (AS/R) systems, bridge cranes, and so on. In a trip-based handling system, a device performs a trip to move one unit load. We assume that the device may handle only one unit load on each trip. (This assumption applies to many handling systems except those such as tractor-trailer systems where a device may pull multiple unit loads at the same time.) The *transfer batch size* (TBS) refers to the size of the unit load handled on each trip. We assume a discrete parts flow system where the unit load size can be expressed as an integer number of parts handled by the device.

In this study, we show that the TBS has a significant impact on the performance of the handling devices as well as work-in-process (WIP) level in the system. However, the problem of determining optimal or near-optimal TBSs has received little attention and, to our knowledge, there are no analytical models presented in the literature.

2 PROBLEM DESCRIPTION

In the manufacturing system we study, there are two types of workstations: input/output(I/O) stations and processing stations. Each workstation has an *input queue* and an *output queue*. Parts arriving from outside the system are directly placed in the output queue of an I/O station, while parts that require no further processing enter the input queue of an I/O station and leave the system instantly. Each external arrival occurs according to a Poisson process and consists of one or more parts depending on the TBS for that part type. Flow is not necessarily conserved at an I/O station, since parts may enter the system from one I/O station and exit from another.

At a *processing station*, parts arrive at the input queue and wait until the processor is available. When the processor is available, a part is removed from the input queue, one part at a time, by the First-Come-First-Served (FCFS) rule and processed for a certain period. After a part is processed, it is staged by the processor until the desired TBS is reached, at which point the parts are placed in the output queue as a unit load. (Material handling concerns *within* the workstations are beyond the scope of our study.)

Once a unit load is placed in an output queue, it is defined as a "move request" and waits for a handling device to pick it up. The handling device first travels empty to the output queue where the move request is located and then it delivers the load to the input queue of the next station. Thus, each trip consists of empty travel followed by loaded travel. The next load to be moved by an empty device is determined by the empty device dispatching rule. For this study, we assume that the move requests are served on a FCFS basis; i.e., an empty device is assigned to the oldest unit load waiting in an output queue. Although as a dispatching rule the FCFS discipline leads to increased empty device travel, it is easier to treat analytically than other empty device dispatching rules such as shortest-travel-time-first (STTF). When a device becomes empty, if there are no unassigned move requests in the system, the device becomes idle at its last delivery point. Also, at the time a move request occurs, if more than one idle device is available, the oldest idle device is dispatched.

In the following study, we are concerned with the handling devices and the total work-in-process (WIP) in the system. The total WIP in the system may be divided into four categories: 1. loads waiting in the input queues, 2. loads being processed and staged by the processors, 3. loads waiting in the output queues, and 4. loads being

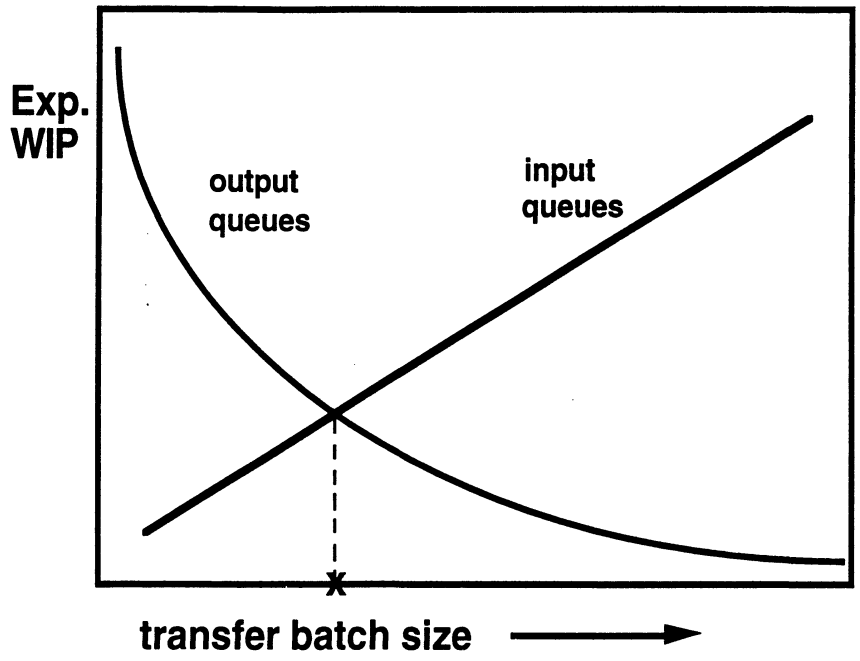


Figure 1: *Transfer batch size .vs WIP*

moved. Generally speaking, for a fixed throughput level, the expected WIP in the input queues increases with the TBS (see Figure 1). This is due to the fact that the arrival of a unit load is analogous to "bulk arrivals" in queueing. On the other hand, the expected WIP in the output queues generally decreases with the TBS since the number of trips that the devices must perform per time unit decreases as the TBS is increased. Given the tradeoff shown in Figure 1, we are concerned with determining the "optimal" TBS so that the total expected WIP (or the total cost associated with WIP) in the system is minimized. For that purpose, we develop separate analytical models to estimate the expected WIP levels in the input and output queues. For the former, we used an $M^{(b)}/G/1$ type of approximation. For the latter, we used an $M/G/c$ type of approximation, where we explicitly capture the empty device travel time as a function of the device dispatching rule (in this case, FCFS).

Remark : Note that determining "optimal" TBS is different from determining the optimal production lot size (PLS) in several ways. First, the PLS is determined by examining the tradeoffs between inventory holding costs and set up costs, while the TBS is determined by examining the tradeoffs between WIP costs associated with the processors and the handling devices. Second, the PLS is determined at a higher level (in the master schedule) than the TBS which is determined at the shop level. Thus, expressed in number of parts, for most manufacturing systems, the TBS is generally considerably smaller than the PLS for a particular part type (unless, of course, PLSs of "one" become a reality). Third, the stochastic behavior of WIP is not explicitly taken into account in determining the PLS.

3 LITERATURE REVIEW

The expected waiting time of a unit load in an output queue is an important measure to estimate WIP associated with the handling system. However, there are few analytical models that take into account a specific device dispatching rule. Chow [4, 5] presents an approximate analytical model to estimate the utilization of a single device (namely, a storage/retrieval, or S/R machine) and the expected waiting times. Assuming that the S/R machine is never blocked, the first and the second moment of the service time distribution are obtained from the flow matrix by a probabilistic argument. Then, He models the system as a single M/G/1 queue with FCFS service. In approximating the expected waiting times, empty travel of the device is not included as part of the waiting time, since "service" is assumed to start as soon as the S/R machine is dispatched to pick up the unit load. (The service time itself, however, includes empty travel.) Cho [3] also uses the M/G/1 queueing model with FCFS service to model a material handling system with a single device. He estimates the expected waiting times in the output queues by "tagging" a move request. The expected service time and device utilization is the same as in Chow's model [4]. However, Cho includes empty travel time as part of the expected waiting times in the output queues. Also, he shows that his model works well for single-device systems and extends it to multiple-device systems by increasing the speed of a single device by a factor equal to the number of devices. However, Cho observes that his model underestimates the expected waiting times in the output queues for handling systems with multiple devices.

Yao and Buzacott [20, 21, 22] model the material handling system as a central server station. In the first study, the service time distribution of the handling system is assumed to be general, while in the other two studies, the service time distribution of the handling system is assumed to be exponential. By modeling the handling system as a central server station, the expected waiting times in the output queues are estimated as a single estimate (i.e., the expected waiting time at the central server queue), regardless of individual workstations. Furthermore, the expected travel times between all the stations are the same regardless of the origin and destination of the move requests. Also, the probability that a job will be routed to a particular station does not depend on where the job is picked up. Solberg [16] and Solot [17] use a similar approach by modeling the material handling system as another workstation.

Bertrand [1] extends the classical economic production batch size model to include WIP carrying cost in a production shop consisting of multiple workstations. He uses the closed queueing network analysis developed by Solberg [16] to estimate the expected time in the system for each job. He shows that the production batch size influences the batch waiting time and WIP in the manufacturing system, and that ignoring WIP carrying cost may result in substantial errors in both determining the optimal production batch size and the optimal cost. Also, he develops an

iterative algorithm to determine optimal production batch size. However, he models the material handling system as another workstation as Solberg [16] and Yao and Buzacott [20, 21, 22].

Bozer, Cho and Srinivasan [2] develop an iterative algorithm to estimate the expected waiting times in the output queues with a single device operating under the MOD FCFS rule. The algorithm is based on their earlier analytic model [18]. They estimate the expected queue lengths first and use Little's formula to estimate the expected waiting times. Starting with an initial estimate of the queue lengths, they observe that the algorithm fails to converge only when the utilization of the device is around 0.99 or higher.

Egbelu [7] proposed an optimization model to determine the container size (i.e., the TBS) and the number of handling devices required. He assumed that only one container is selected for all part types. Hence, the weight capacity of the selected container determines the TBS for each part type; that is,

$$TBS\,of\ part\ type\ j = max\ integer \leq \frac{the\ weight\ capacity\ of\ selected\ container}{the\ weight\ of\ part\ type\ j}.$$

For each container size, his method requires a number of simulation runs to estimate expected WIP in the system and the number of devices required. Subsequently, he uses the above estimates in an optimization model to determine the optimal container size. Although his model includes several types of costs that occur in a manufacturing system, it relies heavily on simulation. Also, some feasible TBSs are not evaluated since the container is always filled up to its capacity. For example, suppose the weight capacity of a selected container is 500 pounds and a part weighs 100 pounds. Then, TBS of that part type is set equal to five and TBSs of one through four are not evaluated, although such TBSs are feasible with the selected container size. In his later model [8] Egbelu determines the number of processors required in addition to the container size and the number of handling devices required by similar approach.

Grasso and Tanchoco [10] investigate the effect of the production batch size from a different point of view. They include the material handling cost and the storage space cost in the cost function of a multi-product MRP problem. The material handling cost is defined as unit cost per move multiplied by the total number of movements in a planning horizon. They show that different batch sizes require different material handling efforts and that including handling and storage costs reduces the optimal order quantities and results in less total cost.

4 ANALYTICAL MODEL TO ESTIMATE WIP

4.1 Notation

The following notation is used throughout the paper. In the analytic model, subscript i and j refer to a station, while subscript k refers to a part type. Let

M = number of workstations in the system
JT = number of part types in the system
D_k = demand for part type k (parts/time unit)
Q_k = transfer batch size of part type k
R_i = set of part types which require processing at workstation i
$=\{k \mid$ part type k visits workstation $i\}$
Ω_k = the set of workstations at which part type k requires processing
$=\{i \mid$ workstation i is visited by part type $k\}$
Λ_i = arrival rate at the input queue of workstation i (unit loads/time unit)
Λ_T = sum of the arrival rates at the input queues of workstations ($\sum_{i=1}^{M} \Lambda_i$)
S_i = processing time at workstation i (time units/part)
$E(N_i)$ = expected number of parts in a unit load arriving at the input queue of workstation i
$E(J_i)$ = expected number of parts in a unit load arriving at the output queue of workstation i ($E(N_i) = E(J_i)$ for processing stations)
λ_i = arrival rate at the output queue of workstation i (unit loads/time unit)
λ_T = sum of the arrival rates at the output queues of workstations ($\sum_{i=1}^{M} \lambda_i = \Lambda_T$)
ND = number of handling devices in the system
p_{ij} = fraction of jobs routed from workstation i to j
P = time required for a device to pick-up or deposit a unit load (constant)
σ_{ij} = expected empty travel time from workstation i to j
$\sigma_{ij}^{(2)}$ = second moment of expected empty travel time from workstation i to j
τ_{ij} = expected loaded travel time from workstation i to j ($\sigma_{ij} + 2P$)
$\tau_{ij}^{(2)}$ = second moment of the expected loaded travel time from workstation i to j

In the above list, Q_k is the primary decision variable, while Λ_i, Λ_T, $E(N_i)$, $E(J_i)$, λ_i, λ_T, and p_{ij} are functions of Q_k. All others are given as parameters and their values are specified by the user. Note that we implicitly assume that the loaded and empty travel time parameters (σ_{ij}, $\sigma_{ij}^{(2)}$, τ_{ij}, and $\tau_{ij}^{(2)}$) as well as the load pick-up or deposit times (P) are independent of the TBS. This assumption is justified for many trip-based handling systems since the weight of the unit load typically has little or no impact on the travel speed of the device.

4.2 Assumptions

We make the following assumptions for the analytical model we develop to estimate the expected WIP in the system:

1. We assume that multiple part types are processed in the system and that the TBS of each part is determined independently.

2. No set up times are taken into account for different part types processed at a particular workstation. However, if the set up times are known in advance, they can be incorporated into the service time for the processor.

3. A workstation always has sufficient processing capacity; it is utilized less than 100%.

4. We assume that unit loads are delivered at the input queue of a station according to a Poisson process; the instant at which a unit load is delivered *is a random point in time* [18].

5. We assume that the move requests occur according to a Poisson process. This assumption is perhaps not entirely valid. However, a general result of renewal theory shows that the superposition of increasingly many component processes yields a Poisson process [12].

6. The layout of the system, part routes, throughput requirements for each part type, and the speed of handling device are assumed to be given.

7. We assume that the travel times between the workstations are exponentially distributed with known mean and are independent of the TBS.

8. For the devices, no blocking effects due to possible congestion are taken into account both in developing the analytical model and in simulation.

9. The devices are homogeneous and they each move one unit load at a time; a unit load includes only one part type.

4.3 Expected Waiting Times in the Input Queues

Parts arrive in bulk at the input queue of each workstation and the number of parts in each arriving unit load varies depending on the TBS of each part type. In this case, based on Poisson arrival of unit loads, the expected waiting time of a *part* in the input queue of workstation i, WI_i^p, is given as follows [14]:

$$WI_i^p = \frac{E(S_i)[E(N_i^{(2)}) - E(N_i)]/E(N_i) + \Lambda_i E(N_i) E(S_i^{(2)})}{2[1 - \Lambda_i E(N_i) E(S_i)]}. \tag{1}$$

In equation (1), the arrival rate (unit load/time unit) at each input queue, Λ_i, is given by

$$\Lambda_i = \sum_{k \in R_i} \frac{D_k}{Q_k}. \tag{2}$$

The first and the second moment of the number of parts in a unit load arriving at the input queue of workstation i are given as follows:

$$E(N_i) = \sum_{k \in R_i} \frac{\frac{D_k}{Q_k} Q_k}{\sum_{l \in R_i} \frac{D_l}{Q_l}} = \frac{\sum_{k \in R_i} D_k}{\Lambda_i} \tag{3}$$

$$E(N_i^{(2)}) = \sum_{k \in R_i} \frac{\frac{D_k}{Q_k} Q_k^2}{\sum_{l \in R_i} \frac{D_l}{Q_l}} = \frac{\sum_{k \in R_i} D_k Q_k}{\Lambda_i}. \tag{4}$$

Note that $E(N_i)$ and $E(N_i^{(2)})$ are assumed to be zero for an I/O station, since all the unit loads arriving at the input queue of an I/O stations either directly enter the output queue or leave the system immediately. By substituting equations (2), (3), and (4) into equation (1), we obtain

$$WI_i^p = \frac{E(S_i)\left[\frac{\sum_{k \in R_i} D_k(Q_k - 1)}{\sum_{k \in R_i} D_k}\right] + E(S_i^{(2)}) \sum_{k \in R_i} D_k}{2[1 - \sum_{k \in R_i} D_k E(S_i)]}. \tag{5}$$

On the other hand, the expected waiting time of *a unit load* in the input queue of station i, WI_i^u, is expressed as follows [14]:

$$\begin{aligned} WI_i^u &= WI_i^p - \text{waiting time due to the parts in the same unit load} \\ &= \frac{E(S_i)[E(N_i^{(2)}) - E(N_i)]/E(N_i) + \Lambda_i E(N_i) E(S_i^{(2)})}{2[1 - \Lambda_i E(N_i) E(S_i)]} \\ &\quad - \frac{E(S_i)[E(N_i^{(2)}) - E(N_i)]}{2E(N_i)}. \end{aligned} \tag{6}$$

By substituting equations (2), (3) and (4) into equation (6), and simplifying, we obtain

$$WI_i^u = \frac{E(S_i)^2[\sum_{k \in R_i} D_k(Q_k - 1)] + E(S_i^{(2)})(\sum_{k \in R_i} D_k)}{2(\sum_{k \in R_i} D_k)[1 - \sum_{k \in R_i} D_k E(S_i)]}. \tag{7}$$

4.4 Expected Waiting Times in the Output Queues

As stated previously, a trip-based material handling device first performs an empty trip and then a loaded trip to satisfy a move request. Under the FCFS dispatching rule, the expected empty travel time to workstation i, E_i, is estimated as follows [3]:

$$E_i = \sum_{h=1}^{M}\sum_{j=1}^{M} p_{hj}\sigma_{ji}\frac{\lambda_h}{\lambda_T} = \sum_{j=1}^{M}\frac{\Lambda_j}{\lambda_T}\sigma_{ji}, \tag{8}$$

since $\sum_{h=1}^{M} p_{hj}\lambda_h = \Lambda_j$. Likewise, the second moment of the expected empty travel time to workstation i, $E_i^{(2)}$, is given by

$$E_i^{(2)} = \sum_{j=1}^{M}\frac{\Lambda_j}{\lambda_T}\sigma_{ji}^{(2)}. \tag{9}$$

The expected loaded travel time from workstation i, L_i, and its second moment, $L_i^{(2)}$, are easily obtained by

$$L_i = \sum_{j=1}^{M} p_{ij}\tau_{ij} \tag{10}$$

and

$$L_i^{(2)} = \sum_{j=1}^{M} p_{ij}\tau_{ij}^{(2)}, \tag{11}$$

where $\tau_{ij} = \sigma_{ij} + 2P$. Note that we include pickup/deposit time, P, as part of the expected loaded travel time.

Based on equations (8) and (10), the expected service time for a device to serve a move request at the output queue of workstation i, T_i, is given by

$$T_i = E_i + L_i, \tag{12}$$

and the second moment of the above service time is given by

$$T_i^{(2)} = E_i^{(2)} + L_i^{(2)} + 2E_iL_i. \tag{13}$$

The expected device utilization is estimated as the total workload divided by the number of devices. The total workload for the handling system, WL, is given by

$$WL = \sum_{i=1}^{M}\lambda_i T_i. \tag{14}$$

Thus, the expected device utilization, ρ, is given by

$$\rho = \sum_{i=1}^{M}\frac{\lambda_i T_i}{ND}, \tag{15}$$

since the handling system consists of ND homogeneous devices. The device utilization is broken down into two components: the expected fraction of loaded travel time and the expected fraction of empty travel time [18]. The expected fraction of loaded travel time, α_f, is easily obtained by

$$\alpha_f = \frac{1}{ND} \sum_{i=1}^{M} \lambda_i \sum_{j=1}^{M} p_{ij} \tau_{ij}, \tag{16}$$

or

$$\alpha_f = \frac{1}{ND} \sum_{k=1}^{JT} \frac{D_k}{Q_k} \sum_{i,j \in \Omega_k} \tau_{ij} \tag{17}$$

where τ_{ij} is the expected loaded travel time from workstation i to workstation j, where i precedes j in the route. Hence, the expected fraction of empty travel time, α_e, is given by

$$\alpha_e = \rho - \alpha_f. \tag{18}$$

We now estimate the expected waiting time of a unit load in each output queue. Under the FCFS rule, the location of each move request (relative to the location of a device) does not play a role in determining the next move request to be served. (This is not true for handling systems in which a location-dependent dispatching rule-such as the STTF rule is used.) However, the location of each move request affects the expected travel time of the devices. Thus, we can assume that each move request forms a single "conceptual queue" and served by the FCFS rule as long as we keep track of the origin and destination of the move requests. Furthermore, we have assumed that all the move requests occur according to a Poisson process and are independent. Hence, we can model a trip-based material handling system with the FCFS dispatching rule as an (M/G/c) queue with FCFS service.

In an (M/G/c) queue with FCFS service, the expected waiting time in the queue is given by the following formula [13]:

$$W_q = \frac{\lambda E(S^2)[\lambda E(S)]^{c-1}}{2(c-1)![c-\lambda E(S)]^2 \left[\sum_{n=0}^{c-1} \frac{[\lambda E(S)]^n}{n!} + \frac{[\lambda E(S)]^c}{(c-1)![c-\lambda E(S)]}\right]}, \tag{19}$$

where c is the number of servers, S is the mean service time and λ is the arrival rate. In equation (19), $\lambda E(S)$ and $\lambda E(S^2)$ represent the first and second moment of the total workload, respectively. The total workload for the handling system was derived earlier in equation (14), and the second moment of the total workload is equal to $\sum_{i=1}^{M} \lambda_i T_i^{(2)}$. Thus, the expected waiting time of a *unit load* in an output queue is

given by

$$W_q = \frac{\sum_{i=1}^{M} \lambda_i T_i^{(2)} [\sum_{i=1}^{M} \lambda_i T_i]^{ND-1}}{2(ND-1)![ND - \sum_{i=1}^{M} \lambda_i T_i]^2 \left[\sum_{n=0}^{ND-1} \frac{[\sum_{i=1}^{M} \lambda_i T_i]^n}{n!} + \frac{[\sum_{i=1}^{M} \lambda_i T_i]^{ND}}{(ND-1)![ND - \sum_{i=1}^{M} \lambda_i T_i]} \right]}. \quad (20)$$

While deriving equation (20), the waiting time in the output queue is defined as the time spent by a move request from the instance of arrival at the output queue to the instance of receiving service. However, in a trip-based handling system, the device starts empty travel while the move request physically remains in the output queue until the device arrives and picks it up. Therefore, the actual expected waiting time of a move request in the output queue of workstation i, WO_i^u, is defined as the time spent by a move request from the instance of arrival at the output queue to the instance of being removed by a handling device and is given by

$$WO_i^u = W_q + E_i. \quad (21)$$

4.5 Expected Time in the System

The expected time in the system of a unit load for part type k, TW_k^u, is given by

$$TW_k^u = \sum_{i,j \in \Omega_k} (WI_i^u + PR_{ik}^u + WO_i^u + \tau_{ij}), \quad (22)$$

where PR_{ik}^u is the expected "time at the processor" at workstation i. Time at the processor is defined as the time spent by the first part of a unit load from the instance of its removal from the input queue to the instance of its placement in the output queue of workstation i. Since parts are "staged" by the server until all the parts in the unit load are processed, PR_{ik}^u is given by

$$PR_{ik}^u = Q_k E(S_i). \quad (23)$$

In equation (22), the right-hand side represents the expected time spent by a unit load at workstation i plus the expected travel time from workstation i to the workstation j. Recall that parts arrive at the input queue of a workstation as a unit load and are picked up from the output queue as a unit load. Thus, the expected time spent by a unit load at a workstation is equal to the expected time spent by a part at the workstation. Also, the expected travel time of a unit load is equal to the expected travel time of a part, since parts are moved as a unit load. Hence, the

expected time spent by a unit load in the system is equal to the expected time spent by a part, by definition.

4.6 Expected WIP in the System

In this section, the expected total WIP in the input queues, output queues, and in the system are estimated by Little's formula. First, the expected total WIP in the input queues, WIP_T^{in}, is given by

$$\begin{aligned} WIP_T^{in} &= \sum_{i=1}^{M} \Lambda_i E(N_i) WI_i^p \\ &= \sum_{i=1}^{M} \sum_{k \in R_i} D_k WI_i^p. \end{aligned} \tag{24}$$

Second, to express the expected total WIP in the output queues in terms of the number of parts, the expected number of parts in a move request should be calculated. The expected number of parts in a move request which arrives at workstation i, $E(J_i)$, is obtained as

$$E(J_i) = \frac{\sum_{k \in R_i} \frac{D_k}{Q_k} Q_k}{\lambda_i} = \frac{\sum_{k \in R_i} D_k}{\lambda_i}. \tag{25}$$

Therefore, the total expected WIP in the output queues, WIP_T^{out}, is obtained as

$$\begin{aligned} WIP_T^{out} &= \sum_{i=1}^{M} \lambda_i E(J_i) WI_i^p \\ &= \sum_{i=1}^{M} \sum_{k \in R_i} D_k WO_i^u. \end{aligned} \tag{26}$$

Recall that $E(J_i)$ is equal to $E(N_i)$ for the processing stations since flow is conserved (i.e., $\Lambda_i = \lambda_i$). However, at I/O stations, $E(N_i)$ is assumed to be zero, while $E(J_i)$ may have a positive value depending on the job routing.

Finally, the total expected WIP in the system, WIP_T^{sys}, is obtained as

$$WIP_T^{sys} = \sum_{k=1}^{JT} Q_k \frac{D_k}{Q_k} TW_k^u = \sum_{k=1}^{JT} D_k TW_k^u. \tag{27}$$

In the next chapter, the performance of the above model, which we refer to as the "WIP model" is evaluated via simulation.

5 EVALUATION OF THE WIP MODEL

To test the WIP model, we used three layouts: layout 1 from Srinivasan, Bozer and Cho [18], layout 2 we generated, and layout 3 from Egbelu [6]. Layout 1 and corresponding input parameters are presented in this section, while layout 2 and input parameters used with layout 2 and 3 are presented in Appendix. Layout 1 is shown in Figure 2 with the distance matrix shown in Table 1. Input parameters, including the number of part types, the throughput requirements for each part type, job routings, and the speed of the devices are shown in table 2. The processing time at each station is assumed to be exponentially distributed with the same known mean regardless of part type. The utilization of each processing station is assumed to be 0.7. Also, the travel times between the workstations are assumed to be exponentially distributed. To evaluate the WIP model, we compare the results obtained by the WIP model with those obtained from simulation. All the simulation results are obtained from 10 replications, in which at least 1000 unit loads of each part type are processed through the system. (The simulation model simulates the "actual" system. That is, load arrivals to input and output queues of *processing* stations are dictated by the simulation model; they are not generated as a Poisson process.) For simplicity, we varied the TBS of part type 6 only. Also, as we increase the number of devices, we reduce the device speed and increase the pickup/deposit time to maintain comparable device utilization.

In evaluation of the WIP model, we have observed similar trends between different layouts. Thus, in this section, we present the results obtained with layout 1 and present the results obtained with other layouts in Appendix.

5.1 Expected Waiting Times in the Input Queues

The results obtained with layout 1 are presented in Figure 3 for each processing station. In Figure 3, the lines labeled as WIP represent the results obtained by the WIP model, while the lines labeled as sim 1, 2, and 4 represent the results obtained via simulation with 1, 2, and 4 devices. As the TBS increases, the WIP model shows two types of trends. For the "affected" workstations (i.e., those workstations on the route of part type 6 such as workstations 5, 8, 9 and 11), the WIP model overestimates the expected waiting times in the input queues. On the other hand, for "unaffected" workstations, (i.e., those that are not on the route of part type 6 such as workstations 6, 7 and 10), the expected waiting times in the input queues are independent of the TBS in the WIP model, while the simulation results suggest that it slightly increases with the TBS. The difference between the two results are small if the TBS is small but becomes large as the TBS increases. We also observe from the simulation that the number of devices in the system has little or no effect on the expected waiting times in the input queues. The same is true for the WIP model since in the WIP model, the expected waiting time in the input queue of a workstation is estimated independent of the number of devices. (Recall that the handling system always meets

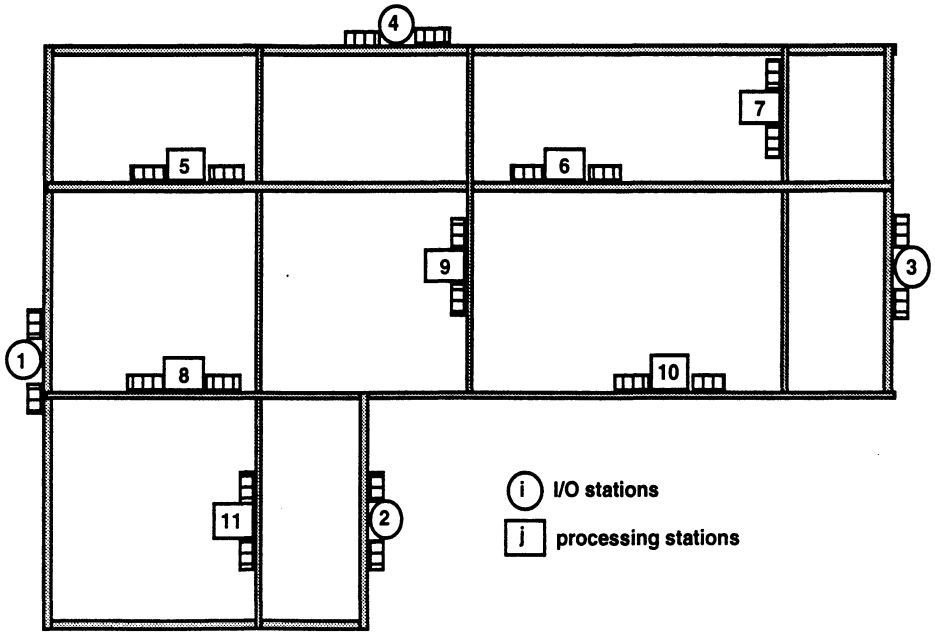


Figure 2: *Layout 1*

Table 1: *Distance matrix of layout 1*

station number	1	2	3	4	5	6	7	8	9	10	11
1	0	22	47	30	14	32	46	8	27	31	17
2	22	0	36	29	23	24	38	14	16	20	14
3	47	36	0	33	37	19	12	39	27	16	41
4	30	29	33	0	16	14	21	25	13	28	27
5	14	23	37	16	0	18	32	16	17	32	18
6	32	24	19	14	18	0	14	27	8	23	29
7	46	38	12	21	32	14	0	41	22	18	43
8	8	14	39	25	16	27	41	0	19	23	9
9	27	16	27	13	17	8	22	19	0	15	21
10	31	20	16	28	32	23	18	23	15	0	25
11	17	14	41	27	18	29	43	9	21	25	0

Table 2: *Job routes and throughput requirements #1 : layout 1*

part type	parts/min	Job Route 1.L1
1	0.1	1 - 5 - 6 - 7 - 9 - 10 - 3
2	0.2	1 - 8 - 11 - 10 - 6 - 9 - 3
3	0.1	1 - 11 - 5 - 8 - 7 - 4
4	0.1	2 - 10 - 9 - 7 - 6 - 4
5	0.1	2 - 8 - 5 - 10 - 7 - 6 - 3
6	0.2	2 - 11 - 8 - 9 - 5 - 4

Device speed:from 200 (with 1 device) to 50 (with 4 devices) distance units/min

Pickup/deposit time is neglegible.

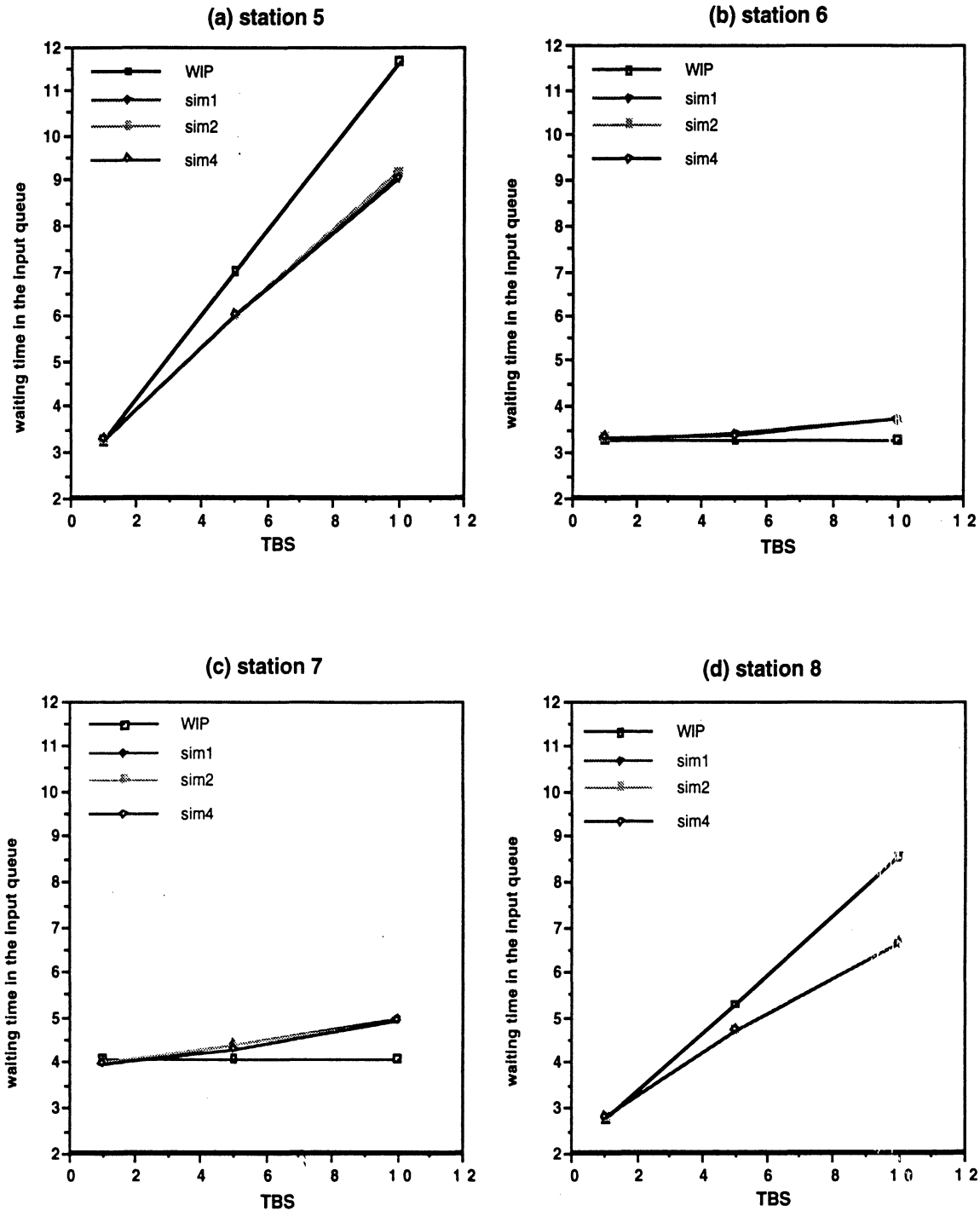


Figure 3: *Expected waiting times in the input queues, layout 1*

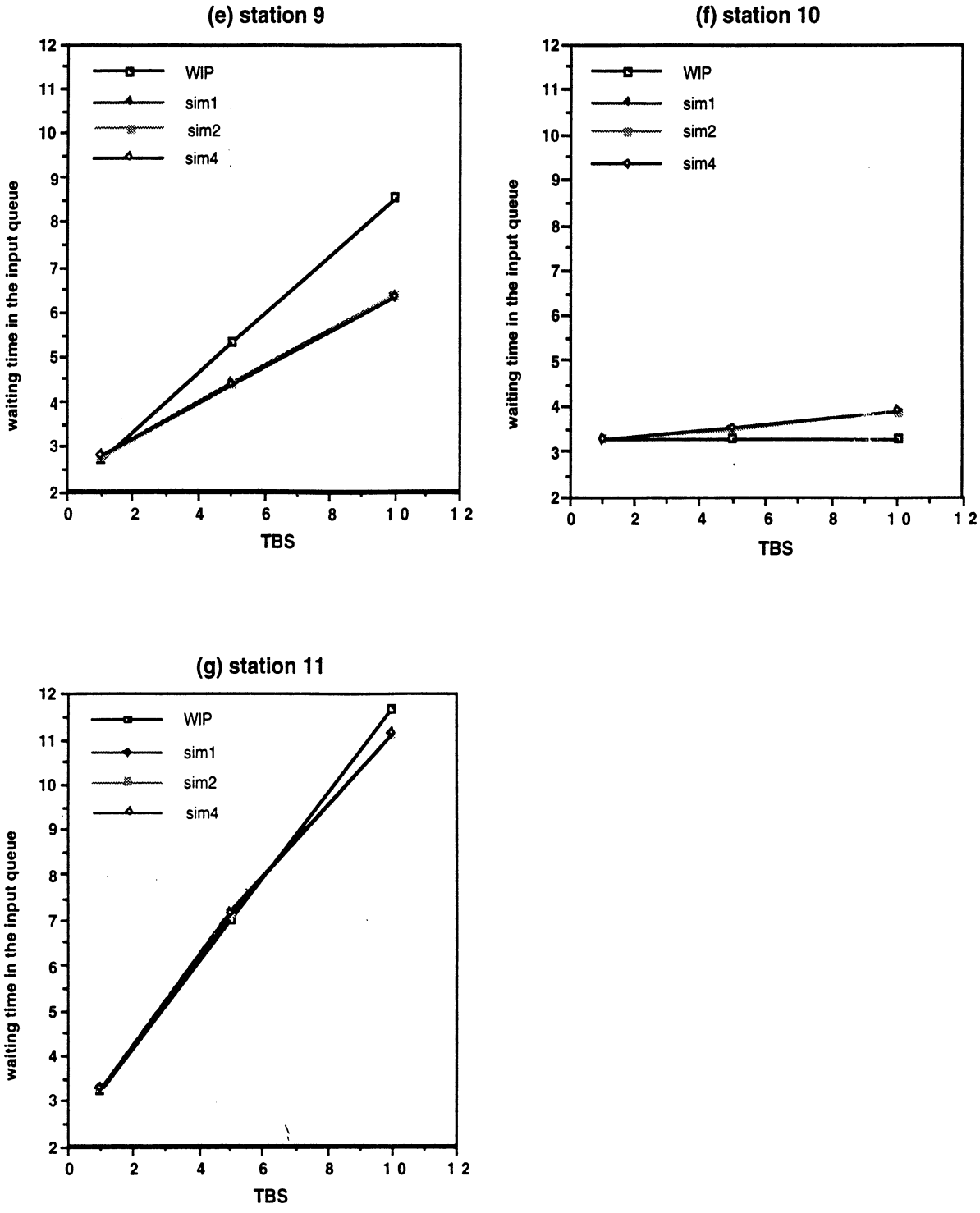


Figure 3: *(contd.) Expected waiting times in the input queues, layout 1*

throughput requirements.)

In spite of the absolute differences observed, the results of the WIP model generally agree with the simulation results in terms of the overall trends. That is, as the TBS increases, the waiting times in the input queues increase or remain approximately the same depending on the route of job type 6.

5.2 Expected Waiting Times in the Output Queues

For the output queues, we compare the results of the WIP model not only with the simulation results but the single faster device approximation (SFDA) using M/G/1. (Although Srinivasan, Bozer and Cho [18] state that the SFDA *is not likely to yield satisfactory results*, we will evaluate it for comparison purpose.) Figures 4, 5, and 6 show the results obtained with 1, 2, and 4 devices, respectively. Each graph presents the expected waiting times in the output queues obtained by SFDA (mg1), by the WIP model (WIP), and by simulation (sim).

In general, the WIP model estimates the expected waiting times in the output queues reasonably well, regardless of the TBS or the number of devices. The maximum relative error is less than 15% and in most cases, it is less than 10%. Also, the absolute error of the WIP model is very small, less than 6 seconds in all cases.

The results agree with our earlier claim; that is, the expected waiting times in the output queues decrease as the TBS increases. As the TBS increases, the total workload (the number of unit loads moved by the handling system per time unit) decreases since the throughput requirement of each part type and the number of devices is fixed. Thus, the expected device utilization decreases, which decreases the expected waiting times in the output queues.

In comparison to SFDA, the results obtained by the WIP model remain reasonably close to the simulation results. However, SFDA underestimates the expected waiting times in the output queues and the differences become larger as the number of devices increases (and the speed of the devices is reduced proportionally at the same time). This confirms the observation made earlier by Srinivasan and Bozer [19], who stated that using a smaller number of faster devices results in less expected waiting times in the output queues than using a larger number of proportionally slower devices.

5.3 Expected Device Utilization

The expected device utilization serves as an important measure to determine the number of devices. (Recall that the expected device utilization should be less than 1.0. to meet the throughput requirement.) In Table 3, we compare device utilizations estimated by the WIP model and by simulation for the TBSs of 1, 5, and 10. Note that

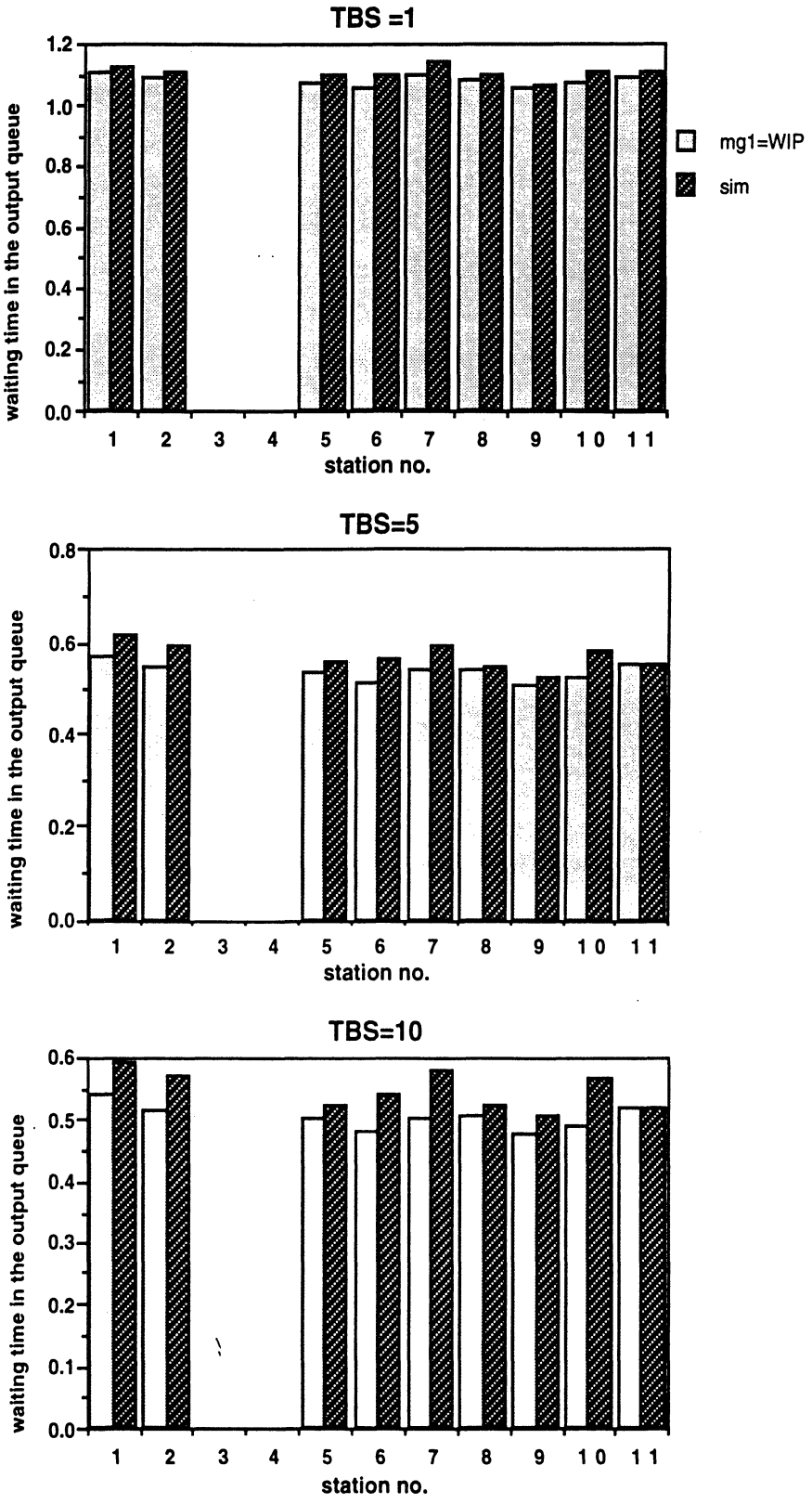


Figure 4: *Expected waiting times in the output queues, layout 1 with one device*

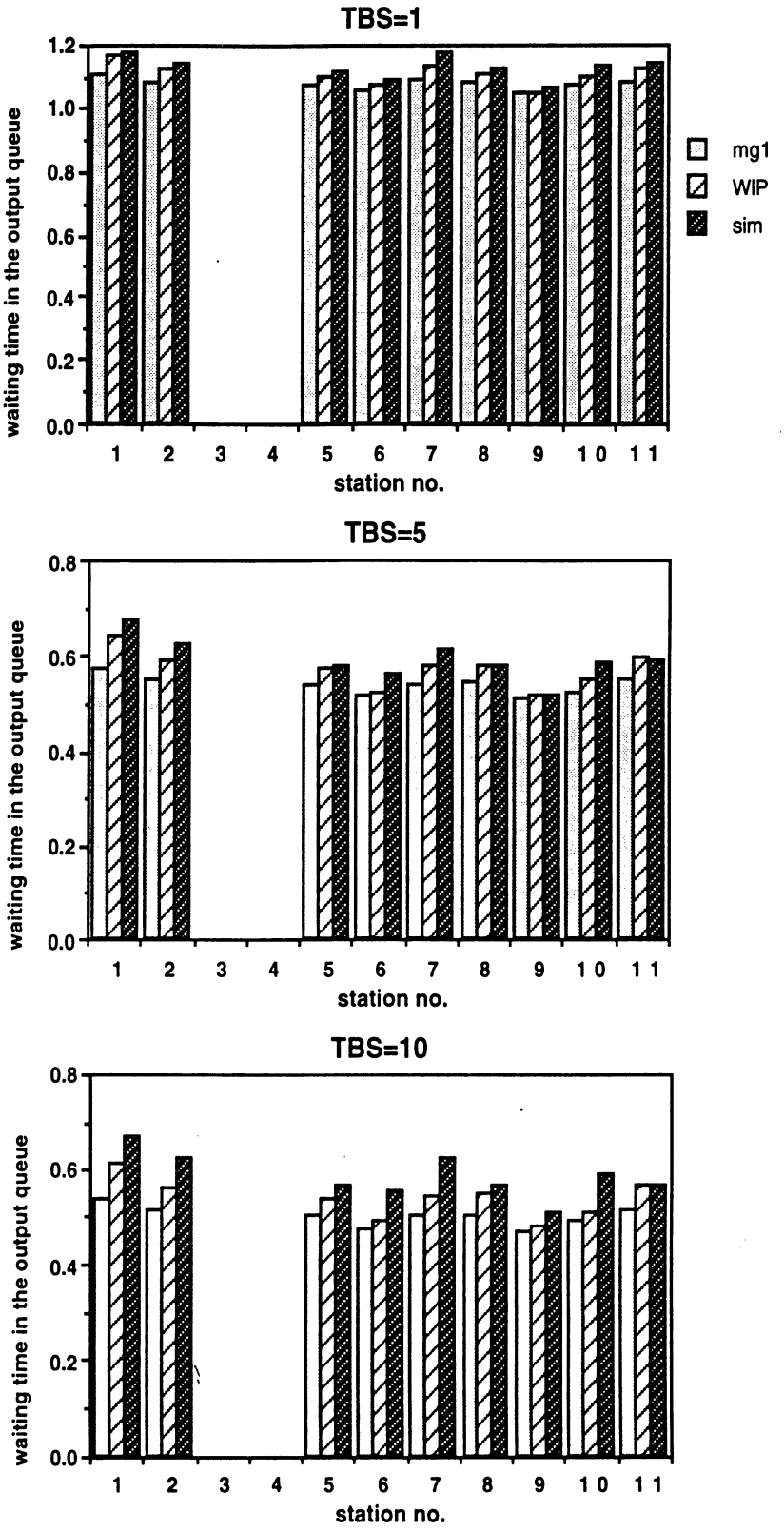


Figure 5: *Expected waiting times in the output queues, layout 1 with two devices*

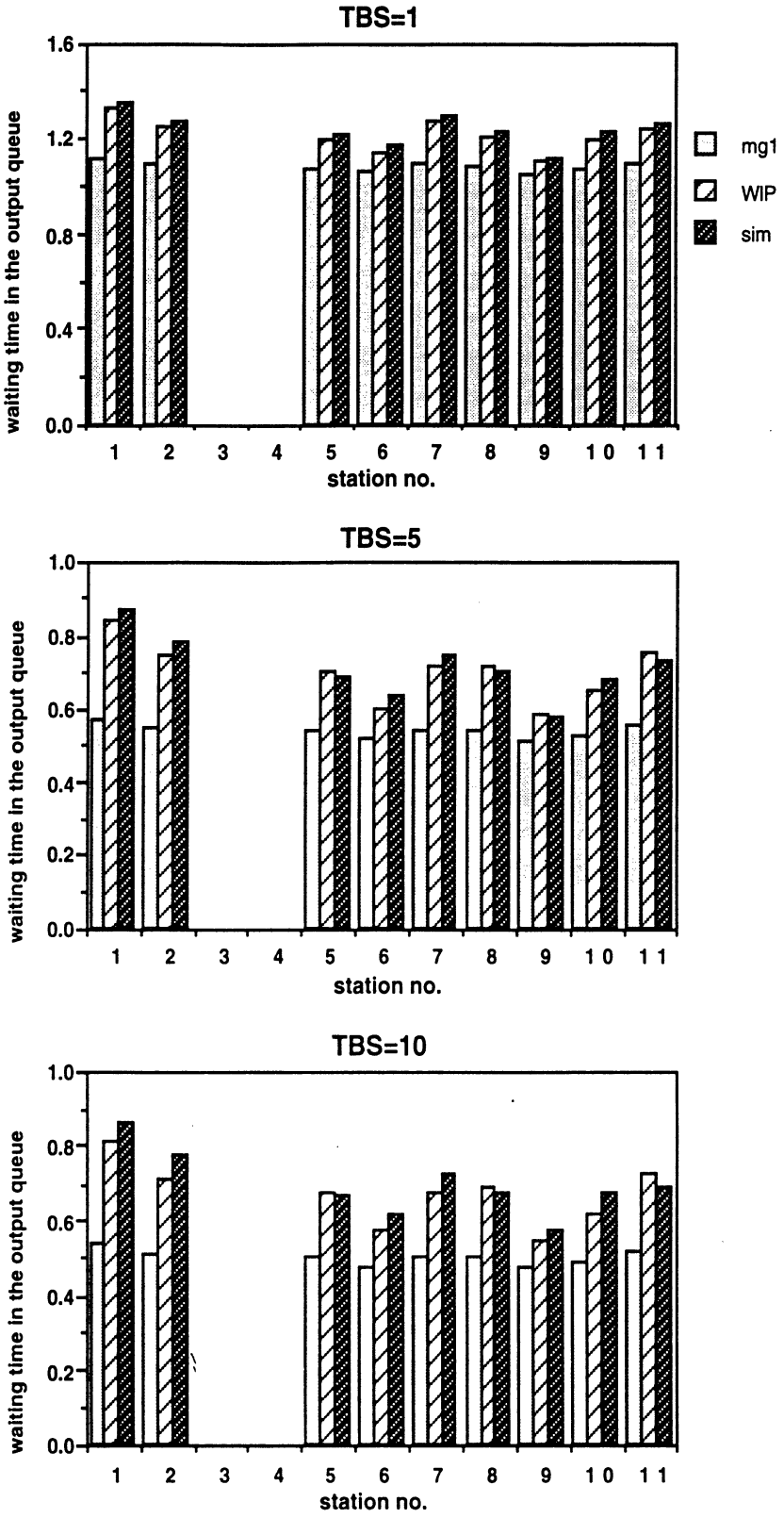


Figure 6: *Expected waiting times in the output queues, layout 1 with four devices*

Table 3: *Expected device utilization, layout 1*

transfer batch size of job 6	1 device		2 devices		4 devices	
	WIP	simulation	WIP	simulation	WIP	simulation
1	0.8536	0.8507	0.8536	0.8494	0.8536	0.8494
5	0.7150	0.7130	0.7150	0.7112	0.7150	0.7101
10	0.6972	0.6901	0.6972	0.6891	0.6972	0.6885

the expected device utilization estimated by the WIP model is the same regardless of the number of devices since we adjusted the speed of the device and the pickup/deposit times to maintain the same expected device utilization as the number of devices varies. The results in Table 3 indicate that the WIP model slightly overestimates the expected device utilization. However, the errors are less than 2% in all cases.

5.4 Total WIP in the System

We use the data shown in Table 4, which is designed to test the cases where the expected device utilization is higher than 0.9. We use six devices and increase the TBS for each part one at a time. The results are shown in Figure 7 for each part type. (The results for part types 4 and 5 are not shown since they are almost identical to those obtained for part type 1.)

If the device utilization is high (> 0.9) such as in Figure 7a and 7d, the WIP model estimates total expected WIP in the input queues quite accurately, while it underestimates total WIP in the output queues. This contradicts the observations made in Section 5.1. However, it can be explained as follows: Since the throughput requirement of part type 1 (7a) represents only a small portion of the total throughput, increasing the TBS of part type 1 does not noticeably increase the expected number of parts in a unit load arriving at the input queues. In addition, we observe in Section 5.1 that the WIP model estimates the expected waiting times in the input queues more accurately when the expected number of parts in a unit load arriving at the input queues is smaller. Hence, the WIP model estimates the expected total

Table 4: *Job routes and throughput requirements #2, layout 1*

part type	parts/min	Job Route 2.L1
1	0.01	1 - 5 - 6 - 9 - 11 - 10 - 3
2	0.03	1 - 7 - 5 - 8 - 10 - 11 - 3
3	0.08	1 - 6 - 7 - 10 - 9 - 4
4	0.01	2 - 5 - 7 - 8 - 11 - 6 - 3
5	0.01	2 - 7 - 9 - 8 - 6 - 11 - 4
6	0.02	2 - 5 - 8 - 9 - 10 - 4

Device speed:from 100 (with 1 device) to 20 (with 4 devices) distance units/min

Pickup/deposit time:is neglegible.

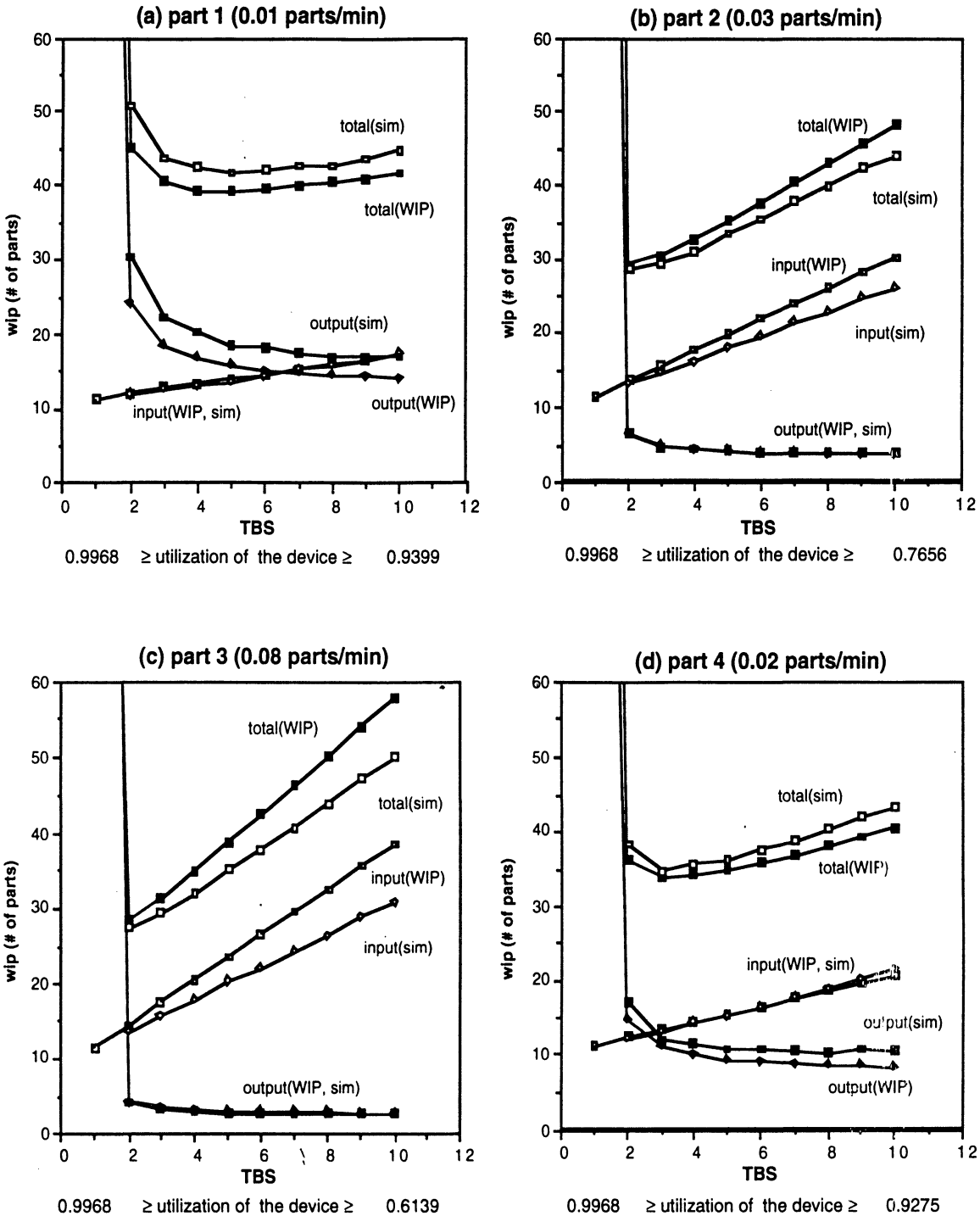


Figure 7: *Expected total WIP in the system*

WIP in the input queues quite accurately. On the other hand, increasing the TBS of part type 1 reduces the total workload for the handling system only by a small amount. Therefore, the device utilization remains above 0.9 even when the TBS is increased to 10. It is well known in heavy traffic theory that when utilization is very high, the margin of error for an approximate model becomes large. Due to the above reasons, total WIP in the output queues estimated by the WIP model results in larger absolute errors.

If the device utilization is less than about 0.9, such as in Figures 7b and 7c, the total WIP in the input queue is overestimated, while total WIP in the output queue is close to the simulation results, which is in agreement with our previous observations. In general, if the device utilization is high, the error in the total WIP in the output queues "dominates" the error in the total WIP in the input queues, resulting in an underestimation of the total WIP. Otherwise, the error in total WIP in the input queues "dominates" the error in the total WIP in the output queues and, consequently, the total WIP is overestimated. However, regardless of the device utilization, the WIP model captures the changes in total WIP quite accurately as the TBS increases.

Note that the expected total WIP in the output queues is almost "flat" as the TBS becomes large. That is, as the TBS increases, the increase in total WIP in the input queues surpasses the decrease in total WIP in the output queues. It indicates that there is an upper bound for the optimal TBS.

In summary, the WIP model performs best when the TBS is small and the device utilization is not too high (less than approximately 0.85).It also captures the behavior of the expected total WIP in the system as a function of the TBS over a wide range of parameter/TBS values.

6 DETERMINING THE "OPTIMAL" TRANSFER BATCH SIZE

In this section, alternative objectives are considered to minimize the operational cost of a manufacturing system within the context of transfer batch sizing. These objectives are used with several constraints to formulate optimization problems. Subsequently, we compare the solution obtained by the WIP model with simulation results.

6.1 Alternative Objectives

One simple cost function is the one based only on the expected WIP cost. In the constraints, the utilizations of both the workstations and the devices should be less than 1.0 to avoid infeasible systems. However, since we assume that a workstation always has sufficient processing capacity, only the device utilization is included as a constraint. Also, there is an upper bound on the TBS of each part type, since the

volume or weight capacity of a device is limited. The resulting problem is formulated as follows:

$$\begin{array}{lll} \text{(P1)} & MIN & C_W WIP_T^{sys} \\ & s.t. & \rho \leq 1 \\ & & Q_k \leq UB_k \quad k = 1, \ldots, JT \\ & & Q_k : \text{positive integer.} \end{array}$$

where C_W is the unit cost of WIP in the system per time unit and UB_k is the upper bound on the TBS of part type k.

Note that, if necessary, two different cost coefficients can be used for the expected total WIP associated with the processors and the handling system. The expected WIP associated with the handling system consists of WIP_T^{out} and the expected number of parts being handled. To obtain the expected number of parts being handled, the expected number of parts in a unit load, PU, is necessary, which is obtained as $\sum_{i=1}^{M} \lambda_i E(J_i)/\lambda_T$. Thus, the expected number of parts being handled is given by $PU \cdot \alpha_f \cdot ND$. Also, the expected WIP associated with the processors are obtained by subtracting the expected WIP associated with the handling system from the expected total WIP in the system. Hence, the objective function can be rewritten as

$$\text{(P2)} \quad MIN \quad C_{WI}(WIP_T^{sys} - WIP_T^{out} - PU \cdot \alpha_f \cdot ND) + C_{WO}(WIP_T^{out} + PU \cdot \alpha_f \cdot ND)$$

where C_{WI} is the unit cost of WIP associated with the processors per time unit and C_{WO} is the unit cost of WIP associated with the handling system per time unit.

In Problems 1 and 2, the number of devices is assumed to be given and fixed. However, the number of devices is also an important design variable since the number of devices is likely to affect the expected WIP level. In combining the fleet-sizing problem with the transfer batch sizing problem, the unit cost of a device should be converted to its annual (or monthly) equivalent cost, since the initial investment in a device is a non-recurring cost. Also, the capital investment required for the devices should be within the budget constraint of the firm. The resulting problem can be formulated as follows:

$$\begin{array}{lll} \text{(P3)} & MIN & C_W WIP_T^{sys} + C_D ND \\ & s.t. & \rho \leq 1 \\ & & Q_k \leq UB_k, \quad k = 1, \ldots, JT \\ & & C_D ND \leq B \\ & & Q_k : \text{positive integer.} \end{array}$$

where B is the available budget for the devices, and C_D is the equivalent cost per time unit per device. (Maintenance costs and other operational costs associated with a device can be included in C_D.)

The cost associated with each space in the input or output queue may also be included in the objective function (as in Grasso and Tanchoco [10]). Since we assume

infinite queue capacities in this study, we will not address the space cost. However, if necessary, the space cost can be included as part of C_{WI} and C_{WO} in Problem 2.

6.2 Preliminary Numerical Results

To demonstrate the type of solutions obtained from the WIP model, Problem 1 is used with $C_W = 1$ and $UB_k = 10$ for all k. Layout 1 is used with the data in Table 4 and the number of devices is varied from one to six devices. (This time, we do not adjust the device speed and P/D time.) The TBSs are obtained through exhaustive enumeration; i.e., we consider all possible TBSs and use the WIP model to estimate the resulting expected WIP. In Figure 8, ten best solutions determined by the WIP model are shown as vectors, where the kth component represents the TBS of part type k. Also, simulation results obtained for each vector are presented in Figure 8 along with linear regression lines based on the *method of least squares.* We cannot claim that the solutions determined by the WIP model are the actual optimal solutions since the WIP model itself is an approximate model. However, in section 6.3, we evaluate the solutions obtained from the WIP model and show empirically that (within the solution space we could search) they are optimal or near-optimal solutions.

First, the best solution obtained by the WIP model is the best or second-best solution according to the simulation results. Also, although we incur some errors in estimating the absolute WIP levels, the WIP model accurately captures the relative changes in total WIP as a function of the TBSs.

Second, the trends in objective function values show "flatness" in both results, regardless of the number of devices. For example, the difference between the objective value of the best solution and the objective value of the 10th best solution is less than 5%. Therefore, any of the 10 solutions may be acceptable for practical purposes. To further check the sensitivity of the objective function, we randomly generated five new solutions, which meet the throughput requirements and are in the interval of the best solution $\pm$ 2 parts. The simulation results with five randomly generated solutions are shown in Figure 9, where the first five TBSs are five best solutions shown in Figure 8, while the last 5 TBSs are those generated randomly and sorted by ascending order of their objective values. The results suggest that the objective function values vary significantly, especially when the number of devices is large. For example, with five devices, the total expected WIP in the system with TBS (1,1,2,1,1,1) is about 40% more than the total expected WIP with the TBS (1,2,2,1,1,1), although the two TBSs differ only in part type 2. Hence, our preliminary results indicate that although the objective function may be "flat" around the "optimal" solution, it may still be sensitive to small changes in the TBS of certain part type. On the other hand, in Figure 9, the objective function is less sensitive when smaller number of devices is used. This can be explained as follows: when the number of devices is small, the TBS

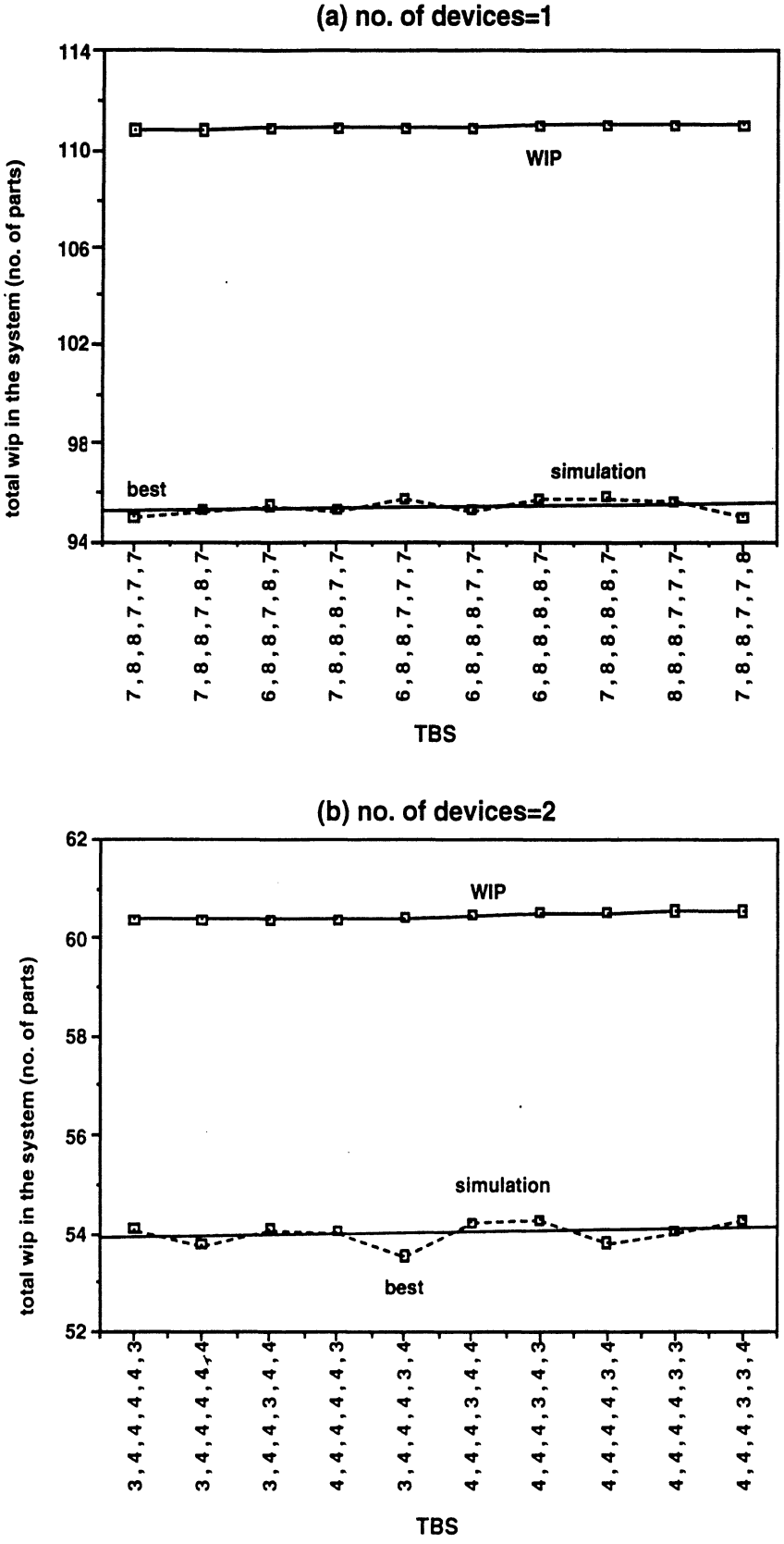


Figure 8: *Best 10 solutions for example problem*

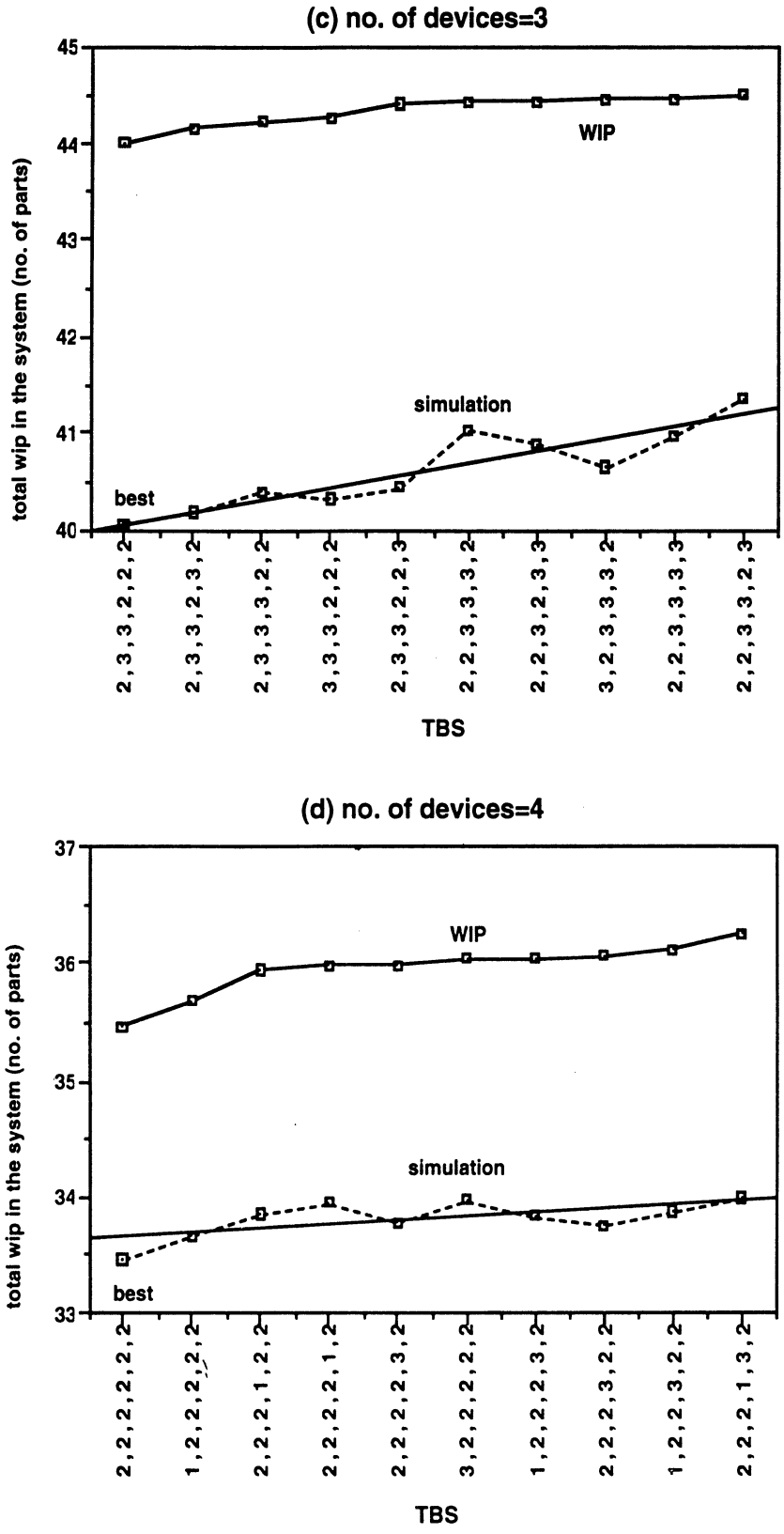


Figure 8: *(contd.) Best 10 solutions for example problem*

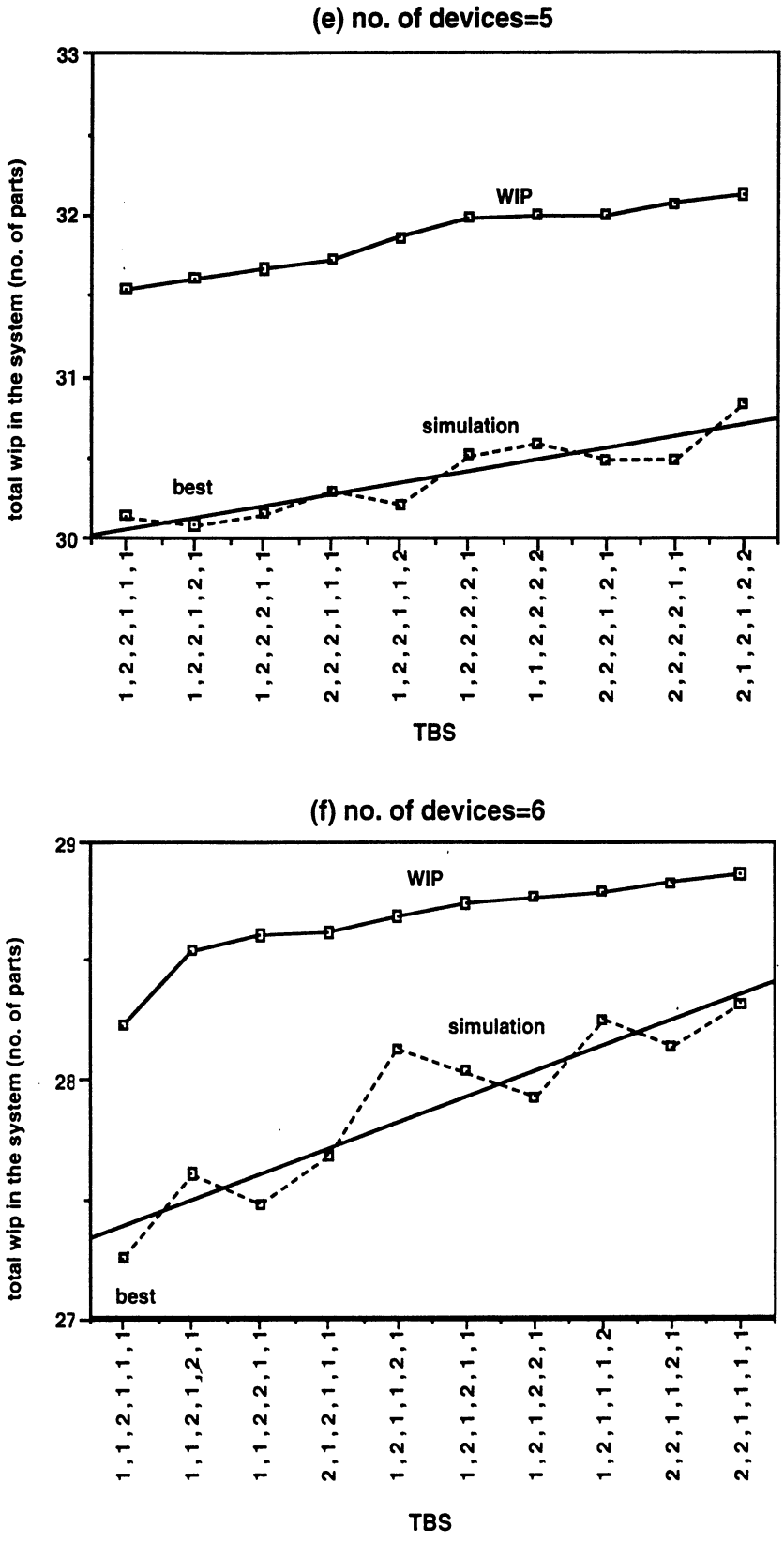


Figure 8: *(contd.) Best 10 solutions for example problem*

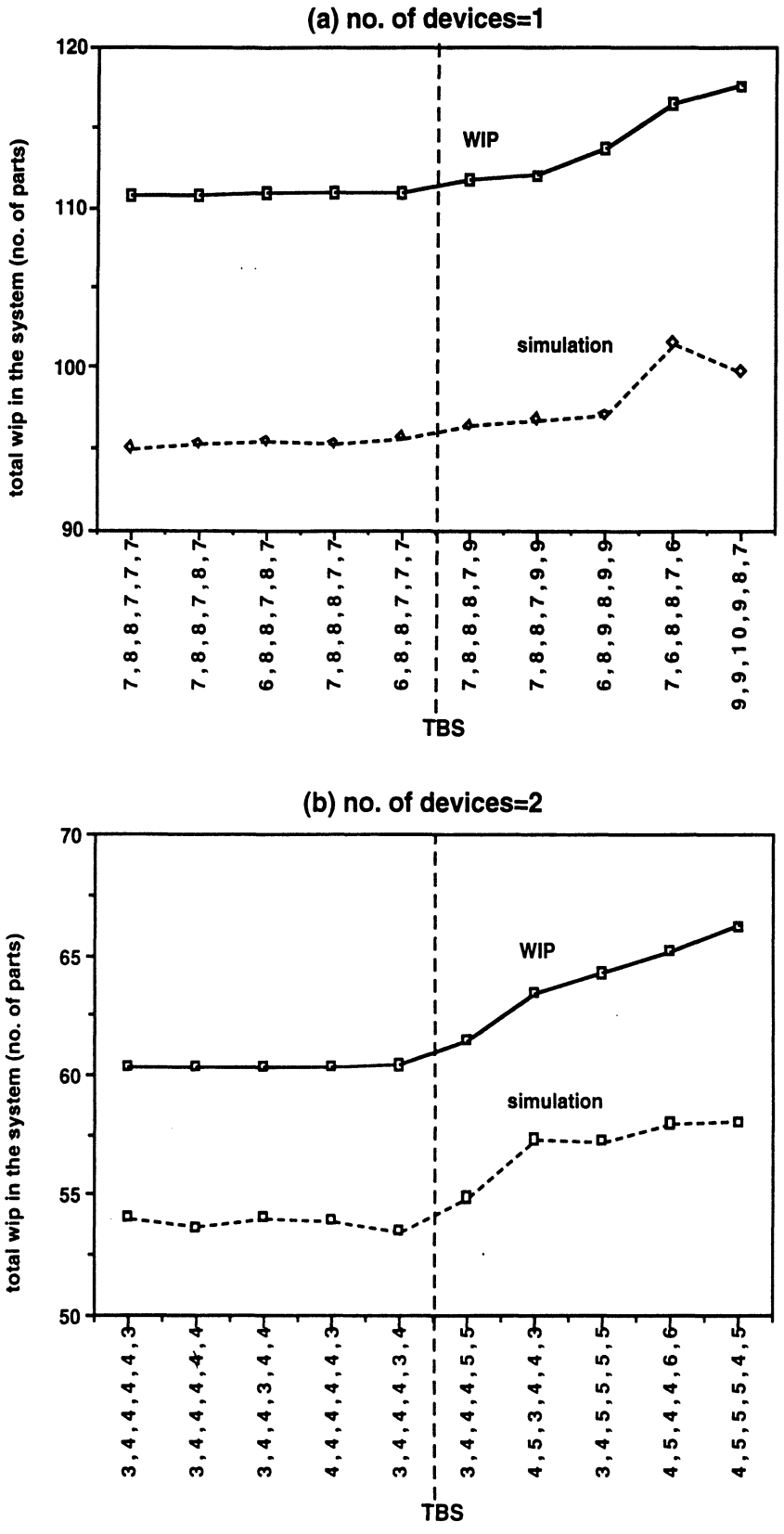


Figure 9: *Best 5 solutions and random solutions for example problem*

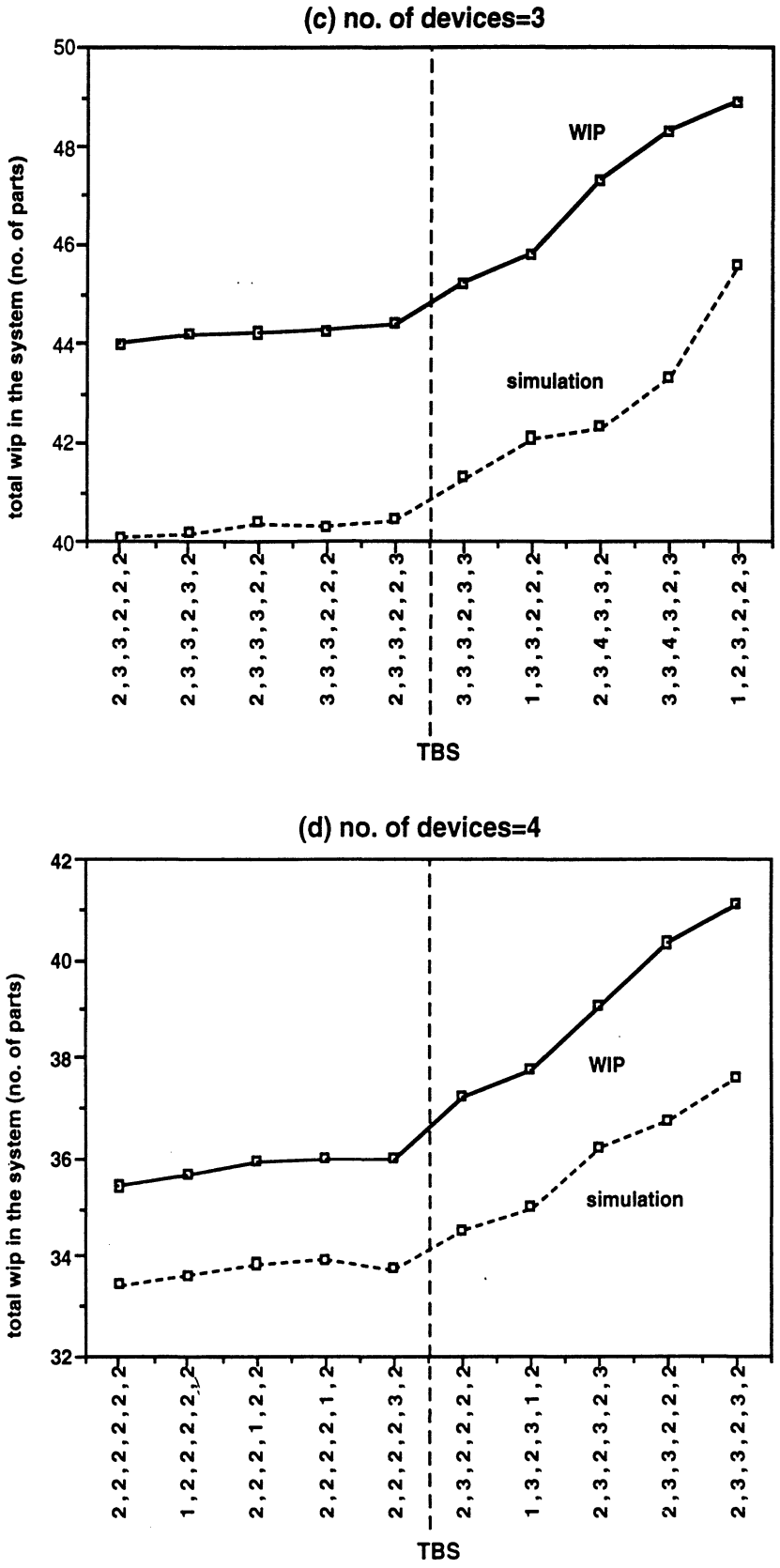


Figure 9: *(contd.) Best 5 solutions and random solutions for example problem*

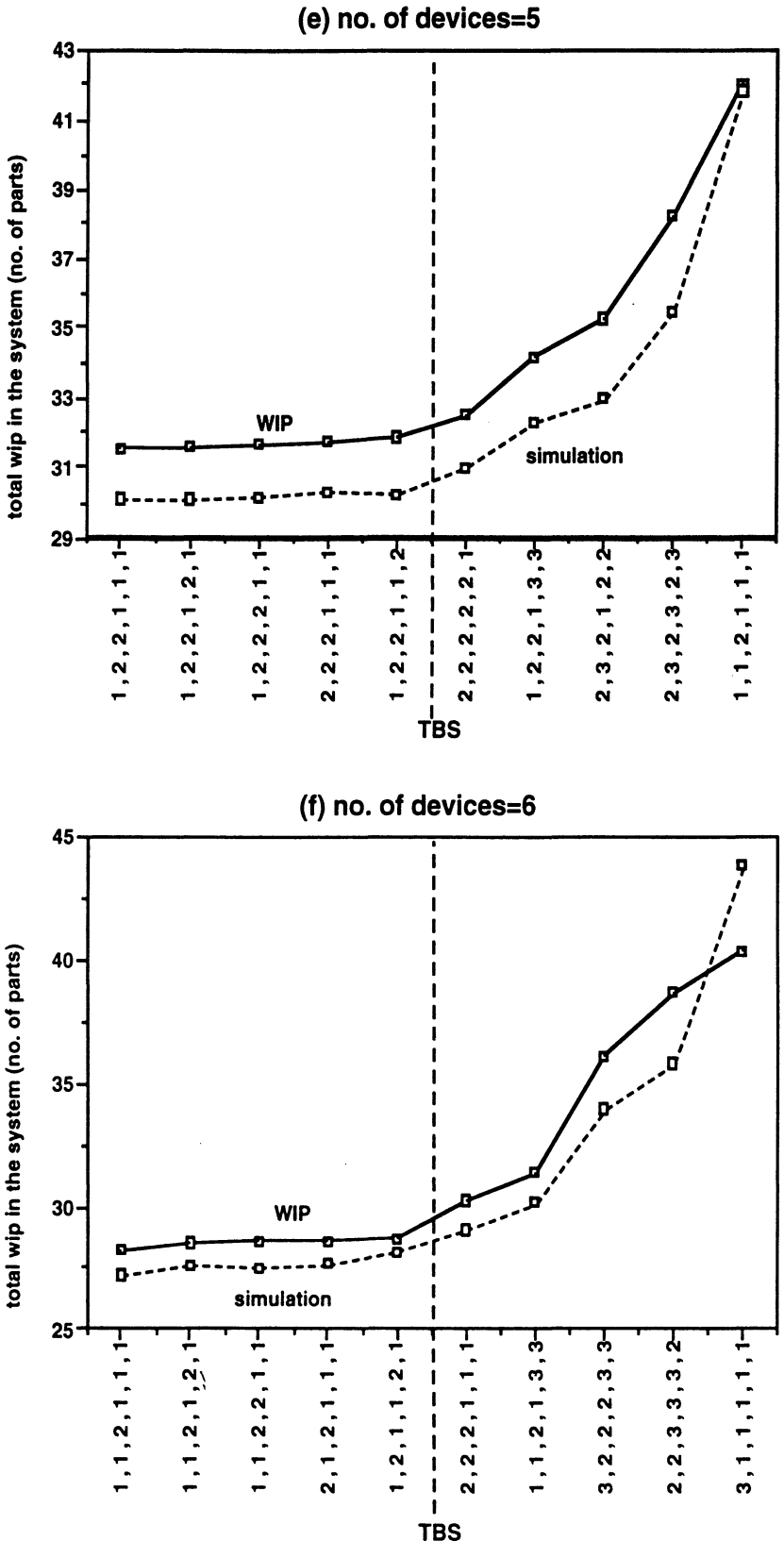


Figure 9: *(contd.) Best 5 solutions and random solutions for example problem*

of each part type is generally large to meet the throughput requirement. Therefore, the *absolute* total expected WIP in the system is already high. Therefore, varying the TBS by 1 or 2 parts produces relatively small differences in the objective function values.

Third, in Figure 8, the difference between the TBS of each part type in a given solution is only one or two parts, which may imply that there is no significant difference between the TBS of each part type in the "optimal" solution. We also observe in Figure 8 that part type 3, which has the largest throughput requirement, also has the largest TBS. To further investigate the effect of the throughput requirement on the solution, we tested the case with all the parts having the same throughput requirements as shown in Table 5, where the device speed and the pickup/deposit times are the same as in Table 4. We tested with 2, 4, and 6 devices and the results are presented in Figure 10, where the TBSs represent 1st, 5th, 10th, 15th, and 20th best solutions. The trends in the objective values are similar to those shown in Figure 8. However, the TBS of each part type shows a larger variation. That is, the same throughput requirement does not necessarily result in the same TBS. This small experiment indicates that the throughput requirement is not the only parameter which affects the "optimal" TBS. For that reason, we need to perform sensitivity analysis on how other parameters may affect the optimal TBS.

6.3 Neighborhood Search

To further test the quality of the solution determined by the WIP model, we simulated all the feasible solutions within the $\pm$ 2 range of the "optimal" solution (determined by exhaustive enumeration using the WIP model). Subsequently, we simulated all such feasible solutions and ranked the resulting objective function values in ascending order. Then we compared the rank obtained from simulation with the rank obtained from the WIP model. The results presented in Figures A12, A13, and A14 in Appendix indicate that above two rankings are highly correlated (with correlation coefficients of 0.9 or larger). Scatter diagrams also support the above observation.

Given that our main concern is to determine the good solutions, we compared the quality of the solutions determined by the WIP model and simulation. Table 6 summarizes ten best solutions obtained by the WIP model and the corresponding objective values obtained by simulation for those TBSs. Also, ten best solutions and objective values determined by simulation are presented. The last column shows percent errors relative to the WIP model. The "optimal" solution obtained from the WIP model is not always the best solution obtained from simulation results. However, taking into account that the error in the objective value is quite small (less than 5% in all test problems) and that simulation results contain random variation, we conclude that the WIP model leads to good solutions.

Table 5: *Job routes and throughput requirements #3 : Layout 1*

part type	parts/min	Job Route 3.L1
1	0.026	1 - 5 - 3
2	0.026	1 - 7 - 5 - 6 - 9 - 8 - 10 - 11 - 3
3	0.026	1 - 6 - 10 - 9 - 4
4	0.026	2 - 5 - 10 - 11 - 6 - 3
5	0.026	2 - 7 - 8 - 11 - 4
6	0.026	2 - 5 - 7 - 8 - 9 - 4

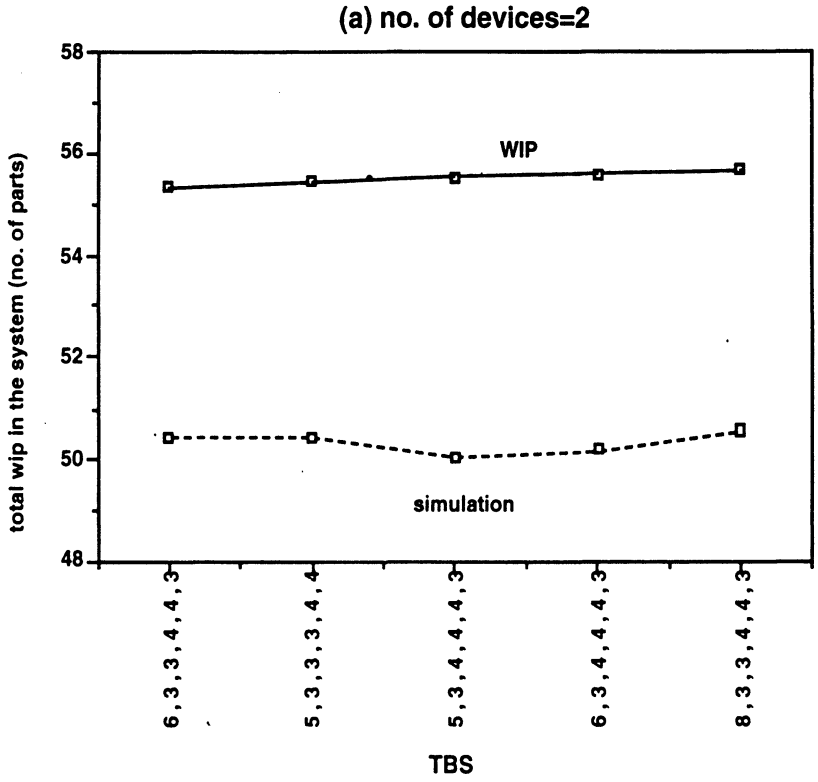


Figure 10: *Solutions obtained with parts that have equal throughput requirements*

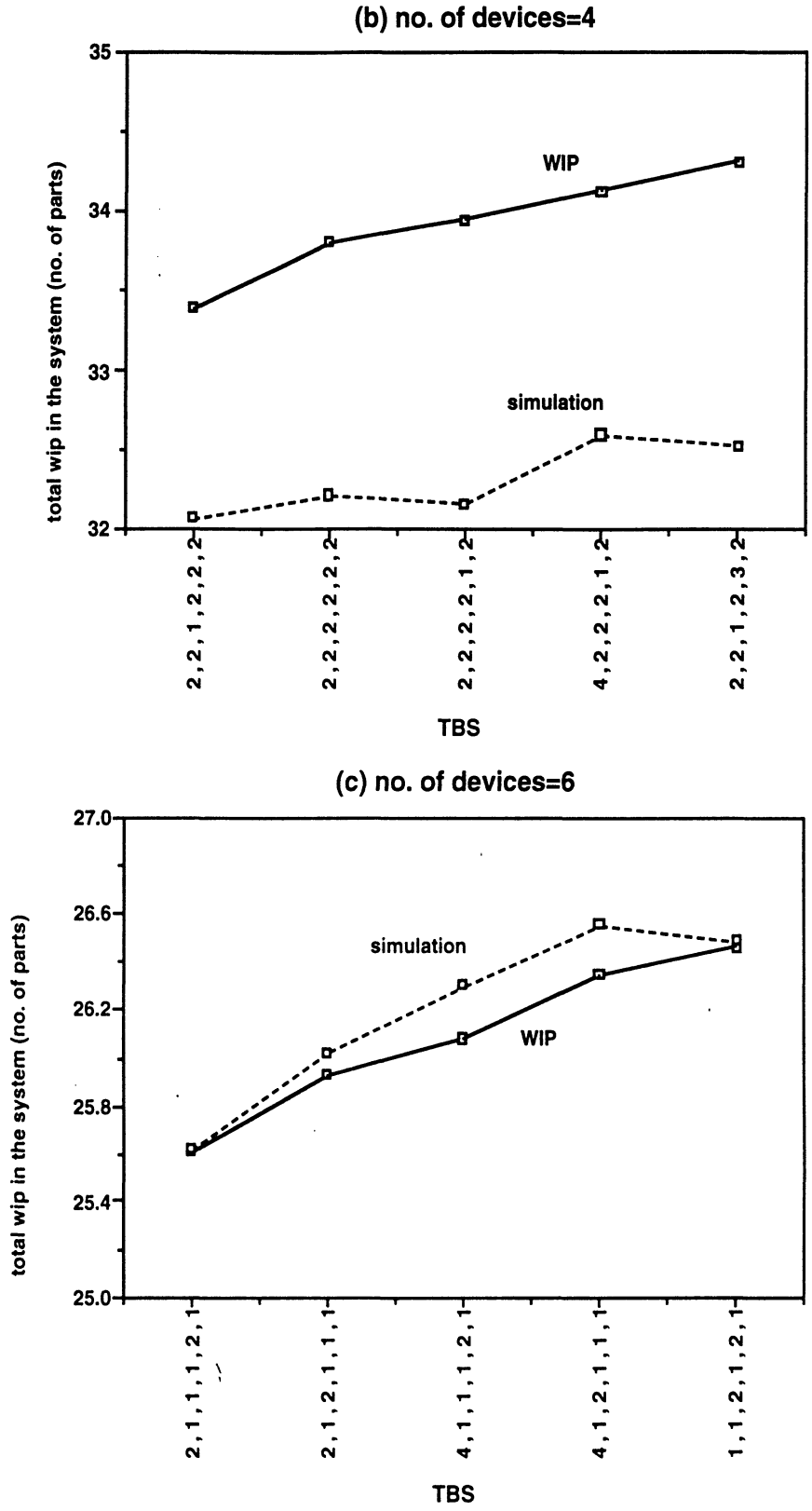


Figure 10: *(contd.) Solutions obtained with parts that have equal throughput requirements*

Table 6: *Comparison of solutions obtained by the WIP model and simulation*

	Best 10 solutions by the WIP model				Best 10 solutions by simulation			
	Rank	T.B.S.	Obj. value	Obj. value in simulation	T.B.S.	Obj. value	95% C.I.	%error (col. 5-col.7)/col. 7
L a y o u t 1	1	2 3 3 2 2 2	44.011	39.525	2 3 3 2 2 2	39.525	2.189	0.00%
	2	2 3 3 2 3 2	44.157	39.835	2 3 3 3 2 2	39.531	1.900	0.77%
	3	2 3 3 3 2 2	44.223	39.531	2 3 3 2 2 3	39.796	1.971	-0.67%
	4	3 3 3 2 2 2	44.271	40.238	2 3 3 2 3 2	39.835	2.402	1.01%
	5	2 3 3 2 2 3	44.412	39.796	2 4 3 2 2 2	39.971	1.329	-0.44%
	6	2 2 3 3 3 2	44.431	40.663	3 3 3 3 2 2	40.127	1.718	1.34%
	7	2 2 3 2 3 3	44.445	40.776	2 4 3 2 3 2	40.226	1.535	1.37%
	8	3 2 3 3 3 2	44.456	40.315	3 3 3 2 2 2	40.238	2.078	0.19%
	9	2 2 3 3 3 3	44.467	41.036	2 3 3 3 3 2	40.239	2.282	1.98%
	10	2 2 3 3 2 3	44.504	41.042	3 2 3 3 3 2	40.315	1.511	1.80%
	Avg.			40.276		39.980		
L a y o u t 2	1	3 2 2 3 2	99.351	87.490	2 3 2 3 2	86.727	4.200	0.88%
	2	2 3 2 3 2	100.105	86.727	4 2 2 3 2	87.115	2.743	-0.45%
	3	2 2 3 3 2	100.411	87.581	3 2 2 3 2	87.490	3.710	0.10%
	4	4 2 2 3 2	100.628	87.115	3 2 3 3 2	87.559	3.059	-0.51%
	5	3 2 3 3 2	100.730	87.559	2 2 3 3 2	87.581	3.567	-0.03%
	6	3 2 2 3 3	101.065	88.401	3 2 2 4 2	87.604	4.182	0.91%
	7	3 3 2 3 2	101.209	88.041	3 3 2 3 2	88.041	4.418	0.00%
	8	3 3 2 2 2	101.482	88.973	2 2 2 4 2	88.073	4.313	1.02%
	9	2 2 2 3 3	101.590	89.317	3 2 2 3 3	88.401	4.548	1.04%
	10	3 2 2 4 2	101.637	87.604	2 3 2 4 2	88.552	4.196	-1.07%
	Avg.			87.881		87.714		
L a y o u t 3	1	4 4 4 4 3	62.501	56.374	3 3 4 3 4	53.596	4.345	5.18%
	2	3 4 4 4 3	62.548	56.085	3 4 3 3 4	53.761	5.693	4.32%
	3	3 4 3 3 4	62.646	53.761	2 4 3 3 4	54.195	5.617	-0.80%
	4	4 4 3 4 3	62.692	57.042	4 4 3 3 4	54.453	5.774	4.75%
	5	3 3 4 4 4	62.746	55.183	2 5 3 3 4	54.515	6.194	1.23%
	6	4 3 3 4 4	62.768	56.556	4 4 2 3 4	54.555	5.947	3.67%
	7	3 3 3 4 4	62.796	55.514	3 4 4 3 4	54.685	5.197	1.52%
	8	5 4 4 4 3	62.847	57.287	2 4 4 3 4	54.717	5.220	4.70%
	9	4 4 3 3 4	62.851	54.453	2 6 2 3 4	54.958	5.385	-0.92%
	10	4 3 4 4 4	62.878	57.139	3 3 4 4 4	55.183	4.429	3.54%
	Avg.			55.939		54.462		

6.4 Genetic Algorithm

Although we used exhaustive enumeration to determine the "optimal" solutions in the preliminary results, needless to say it takes a very long time when the number of parts is large. To search the solution space more systematically and in a more efficient manner, we decided to apply a Genetic Algorithm (GA) which seems suitable for our problem structure. GAs are search algorithms developed by Holland [11] and his colleagues, based on the mechanics of natural selection and genetics. A simple GA is composed of three operations: *reproduction, crossover* and *mutation.* (Readers may refer to Goldberg [9].) The advantage of a GA is that it can handle complex objective functions, even discontinuous functions, which represent a major obstacle for classical optimization techniques. Also, the concepts are simple and easy to program yet powerful. The disadvantage of a GA is that it is not guaranteed to find the global optimal solution and the quality of the solution depends on the population size, crossover probability, and mutation probability. Also, some problems may require long computation times, especially if the objective function is time consuming to evaluate.

We tested several variations of a simple GA to evaluate the quality of the resulting solutions. After testing various problems, we constructed a simple GA with "elitist reproduction" and "biased mutation." Elitist reproduction is a technique to pass over the best solutions from the previous generation to the next generation. Mutation is the occasional (with small probability) random alteration of the value of a string position. In the biased mutation, we alter the value of a string based on the current value of the string instead of altering randomly. For instance, if a string is picked to be mutated and it's value lies in the lower half interval of the feasible region, then the value of that string is altered to a random integer in the upper half interval of the feasible region and vice versa. We stop the algorithm either when it reaches the maximum number of iterations (1000 in our test cases) or the 15th current best solution is not improved for 50 iterations. With the above settings, the GA we developed obtained the same "optimal" solutions we obtained with exhaustive enumeration for all the test problems (see Table 7).

7 CONCLUSIONS

We have demonstrated that the TBS affects both the expected WIP in the input queues and the output queues and proposed an analytical model to determine the "optimal" TBS. Subsequently, we have shown that the the "optimal" solution determined by the WIP model is comparable to solutions obtained by simulation. We are currently working on testing more problems and extending the application of the WIP model. We are testing the cases where all the parts are required to have the same TBSs and testing the performance of the WIP model under other than the

Table 7: *"Optimal" solutions obtained by Genetic Algorithm*

Layout		1 device		3 devices		5 devices	
		Enumeration	G.A.	Enumeration	G.A.	Enumeration	G.A.
1	T.B.S.	**7,8,8,7,7,7**	**7,8,8,7,7,7**	**2,3,3,2,2,2**	**2,3,3,2,2,2**	**1,2,2,1,1,1**	**1,2,2,1,1,1**
	Total WIP	**110.79**	**110.79**	**44.01**	**44.01**	**31.54**	**31.54**
2	T.B.S.	**8,7,7,8,6**	**8,7,7,8,6**	**3,2,2,3,2**	**3,2,2,3,2**	**2,1,2,2,1**	**2,1,2,2,1**
	Total WIP	**252.925**	**252.925**	**99.3507**	**99.3507**	**70.893**	**70.893**
3	T.B.S.	**11,11,11,11,10**	**11,11,11,11,10**	**4,4,4,4,3**	**4,4,4,4,3**	**2,2,3,2,2**	**2,2,3,2,2**
	Total WIP	**160.88**	**160.88**	**62.5**	**62.5**	**42.9147**	**42.9147**

Number of population = 50

Crossover probability = 0.7

Mutation probability = 0.01

Elitist reproduction = 8%

FCFS dispatching rule. Also, we are testing dynamic transfer batch sizing, where a *range* of TBSs is used instead of a fixed value. In this paper, we assumed that the number of devices is given. However, the model can be easily extended to determine the "optimal" number of devices in conjunction with the "optimal" TBSs.

References

[1] Bertrand, J. W. M., "Multiproduct Optimal Batch Size with In-Process Inventories and Multi Work Centers," *IIE Transactions*, Vol. 17, No. 2, pp. 157–163, 1985.

[2] Bozer, Y. A., M. S. Cho and M. M. Srinivasan, "Expected Waiting Times in Single-Device Trip-Based Material Handling Systems," Technical Report 90-17, Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, 1990, to appear in the *European Journal of Operational Research.*

[3] Cho, M. S., "Design and Performance Analysis of Trip-Based Material Handling Systems in Manufacturing," Ph.D. Dissertation, Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, 1990.

[4] Chow, W. M., "Design for Line Flexibility," *IIE Transactions*, Vol. 18, No. 1, pp. 95–108, 1986.

[5] Chow, W. M., "An Analysis of Automated Storage and Retrieval Systems in Manufacturing Assembly Lines," *IIE Transactions*, Vol. 18, No. 2, pp. 204–214, 1986.

[6] Egbelu, P. J., "The Use of Non-Simulation Approaches in Estimating Vehicle Requirements in an Automated Guided Vehicle Based Transport System," *Material Flow*, Vol. 4, Nos. 1–2, pp. 17–32, 1987.

[7] Egbelu, P. J., "Economic Design of Unit Load-Based FMSs Employing AGVs for Transport," *Intl. Journal of Production Research*, Vol. 31, No. 12, pp. 2753–2775, 1993.

[8] Egbelu, P. J., "Concurrent Specification of Unit Load Sizes and Automated Guided Vehicle Fleet Size in Manufacturing System," *Intl. Journal of Production Economics*, Vol. 29, pp. 49–64, 1993.

[9] Goldberg, D. E., *Genetic Algorithms*, Addison-Wesley Publishing Co., New York, 1989.

[10] Grasso, E. T. and J. M. A. Tanchoco, "Unit Load and Material Handling Considerations in Material\Requirements Systems," *Material Flow*, Vol. 1, pp. 79–87, 1983.

[11] Holland, J. H., *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, MI, 1975.

[12] Kuehn, P. J., "Approximate Analysis of general Queueing Networks by Decomposition," *IEEE Transactions on Communications*, Vol. COM-27, No. 1, pp. 113–126, 1979.

[13] Nozaki, S. and S. Ross, "Approximations in Finite Capacity Multiserver Queues with Poisson Arrivals," *Journal of Applied Probability*, 15, pp.826–834, 1978.

[14] Ross, S. M., *Introduction to Probability Theory*, Academic Press Inc., FL, 1985.

[15] Solberg, J.J., "The Optimal Planning of Computerized Manufacturing Systems: CAN-Q User's Guide," School of Industrial Engineering, Purdue University, West Lafayette, IN, 1980.

[16] Solberg, J. J., "Capacity Planning with a Stochastic Workflow Model," *AIIE Transactions*, Vol. 13, No. 2, pp. 116–122, 1981.

[17] Solot, P. and J. M. Bastos, "MULTIQ: A Queueing Model for FMSs with Several Paallet Types," *Journal of the Operational Research Society*, Vol. 39, No. 9, pp. 811–821, 1988.

[18] Srinivasan, M. M., Y. A. Bozer and M. S. Cho, "Trip-Based Material Handling Systems: Throughput Capacity Analysis," *IIE Transactions*, Vol 26 No. 1, pp. 70–89, 1994.

[19] Srinivasan, M. M. and Y. A. Bozer, "Which One is Responsible for WIP in Manufacturing: The Workstations or the Material Handling System," *Intl. Journal of Production Research*, Vol. 30, No. 6, pp. 1369–1399, 1992.

[20] Yao, D. D. and J. A. Buzacott, "Modeling the Performance of Flexible Manufacturing Systems," *Intl. Journal of Production Research*, Vol. 23, No. 5, pp. 945–959, 1985.

[21] Yao, D. D. and J. A. Buzacott, "Models of Flexible Manufacturing Systems with Limited Local Buffers," *Intl. Journal of Production Research*, Vol. 24, No. 1, pp. 107–118, 1986.

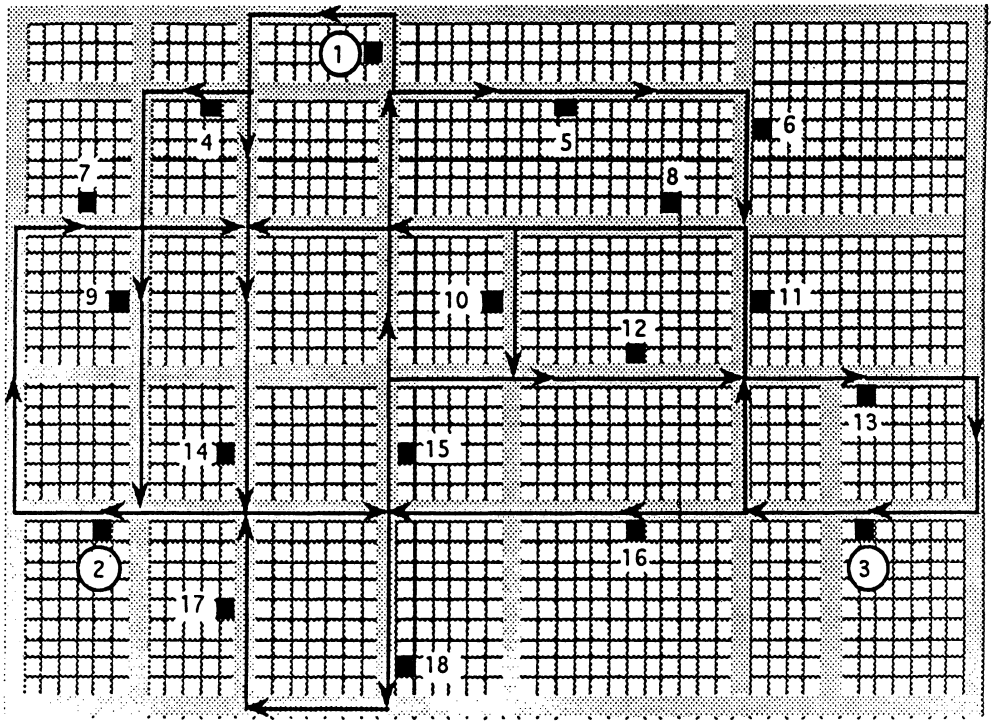
[22] Yao, D. D. and J. A. Buzacott, "Modeling a Class of Flexible Manufacturing Systems with Reversible Routing," *Operations Research*, Vol. 35, No. 1, pp. 87–93, 1987.

APPENDIX

Table A1: *Job routes and throughput requirements #1, layout 2*

part type	parts/min	Job route 1.L2
1	0.027	1 - 8 - 6 - 10 - 11 - 3 -
2	0.041	1 - 4 - 9 - 7 - 14 - 17 - 2
3	0.041	2 - 14 - 15 - 10 - 12 - 13 - 3
4	0.055	3 - 16 - 18 - 15 - 17 - 2 -
5	0.027	3 - 11 - 8 - 5 - 6 - 13 - 3

Device speed:from 35 (with 3 devices) to 15 (with 7 devices) distance units/
P/D time = 5.16 (with 3 devices) to 12 (with 7 devices) secs
Processor utilization=0.75



Each grid represents one distance unit.

I/O station processing station

Figure A1: *Layout 2*

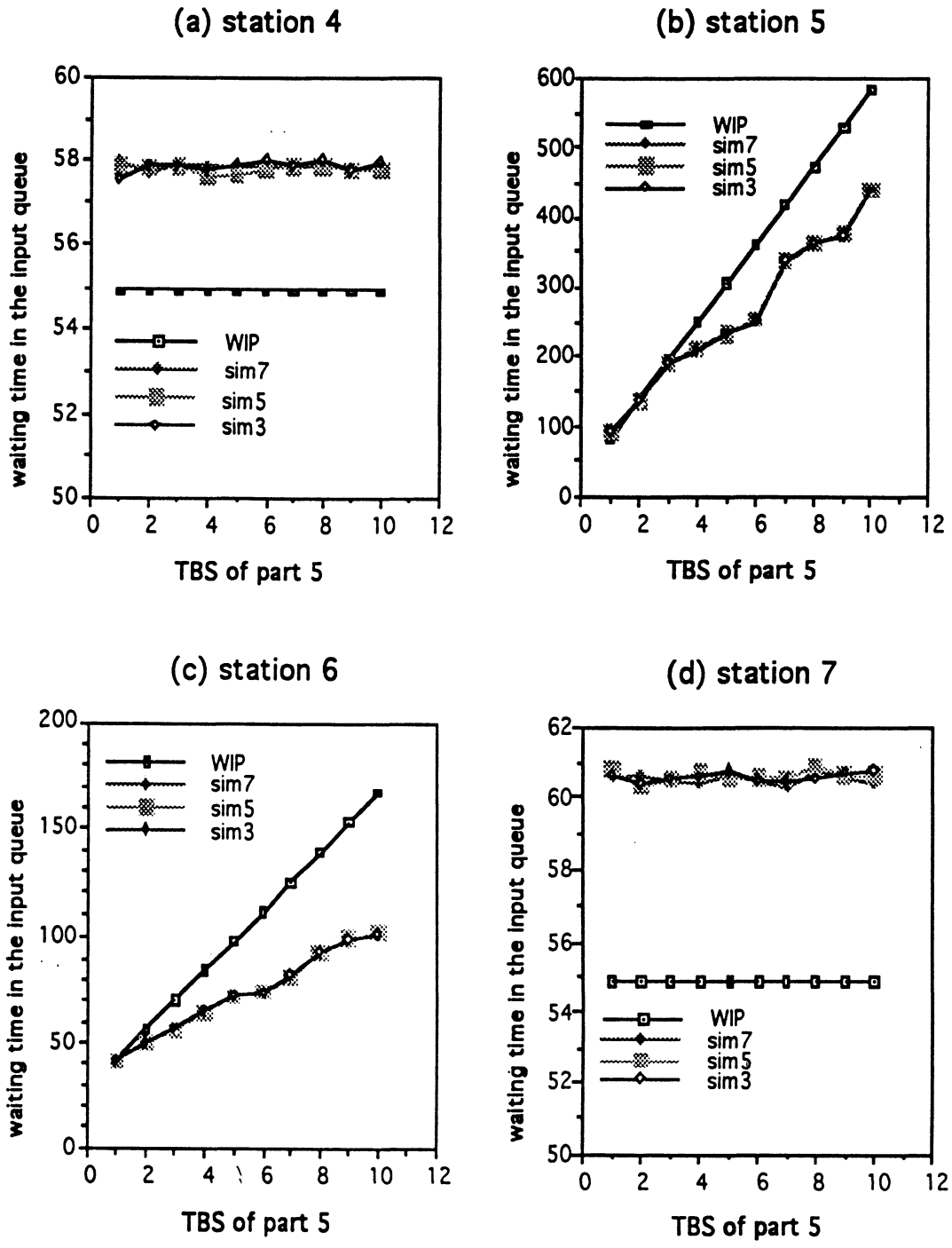


Figure A2: *Expected waiting times in the input queues, layout 2*

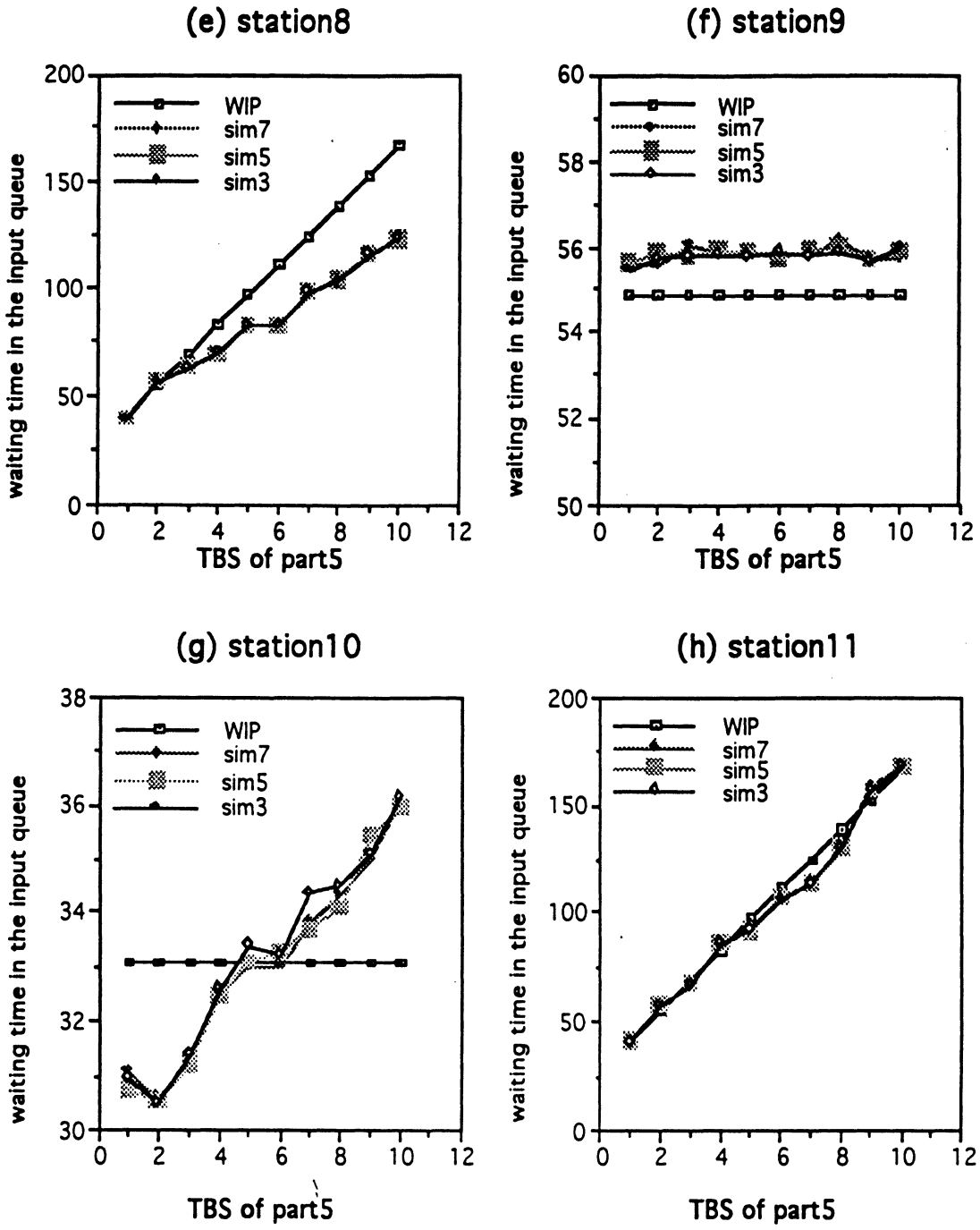


Figure A2: *(contd.) Expected waiting times in the input queues, layout 2*

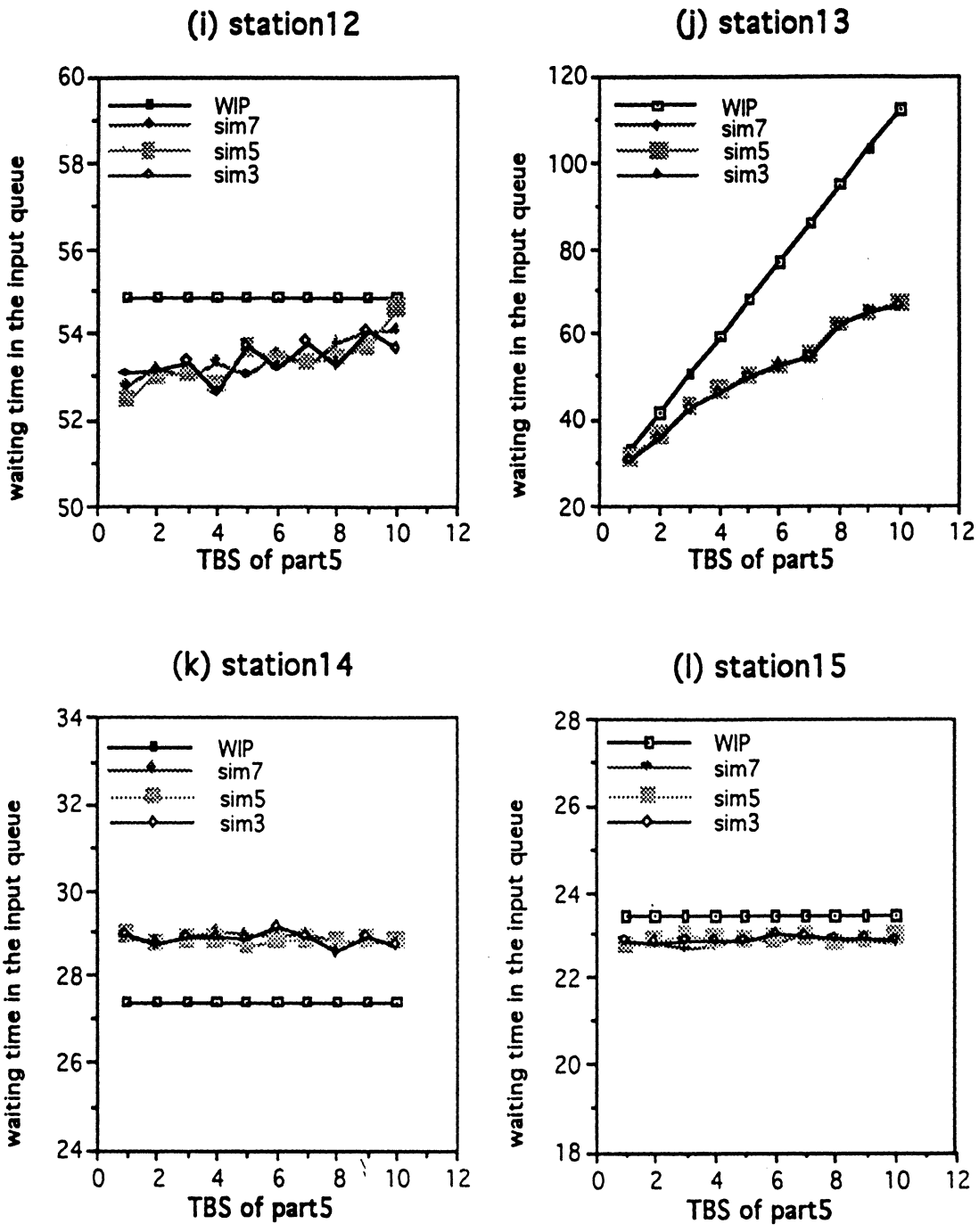


Figure A2: *(contd.) Expected waiting times in the input queues, layout 2*

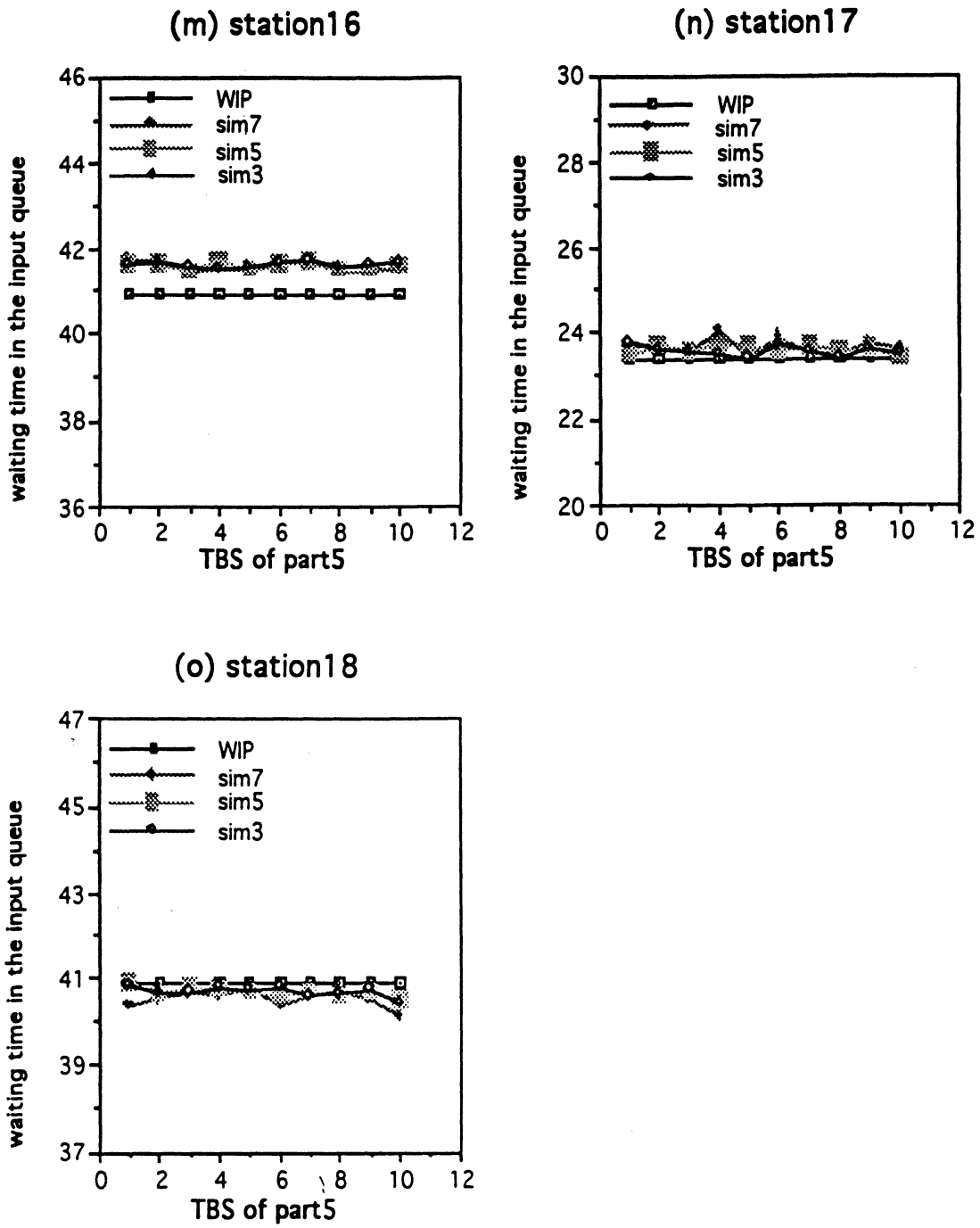


Figure A2: *(contd.) Expected waiting times in the input queues, layout 2*

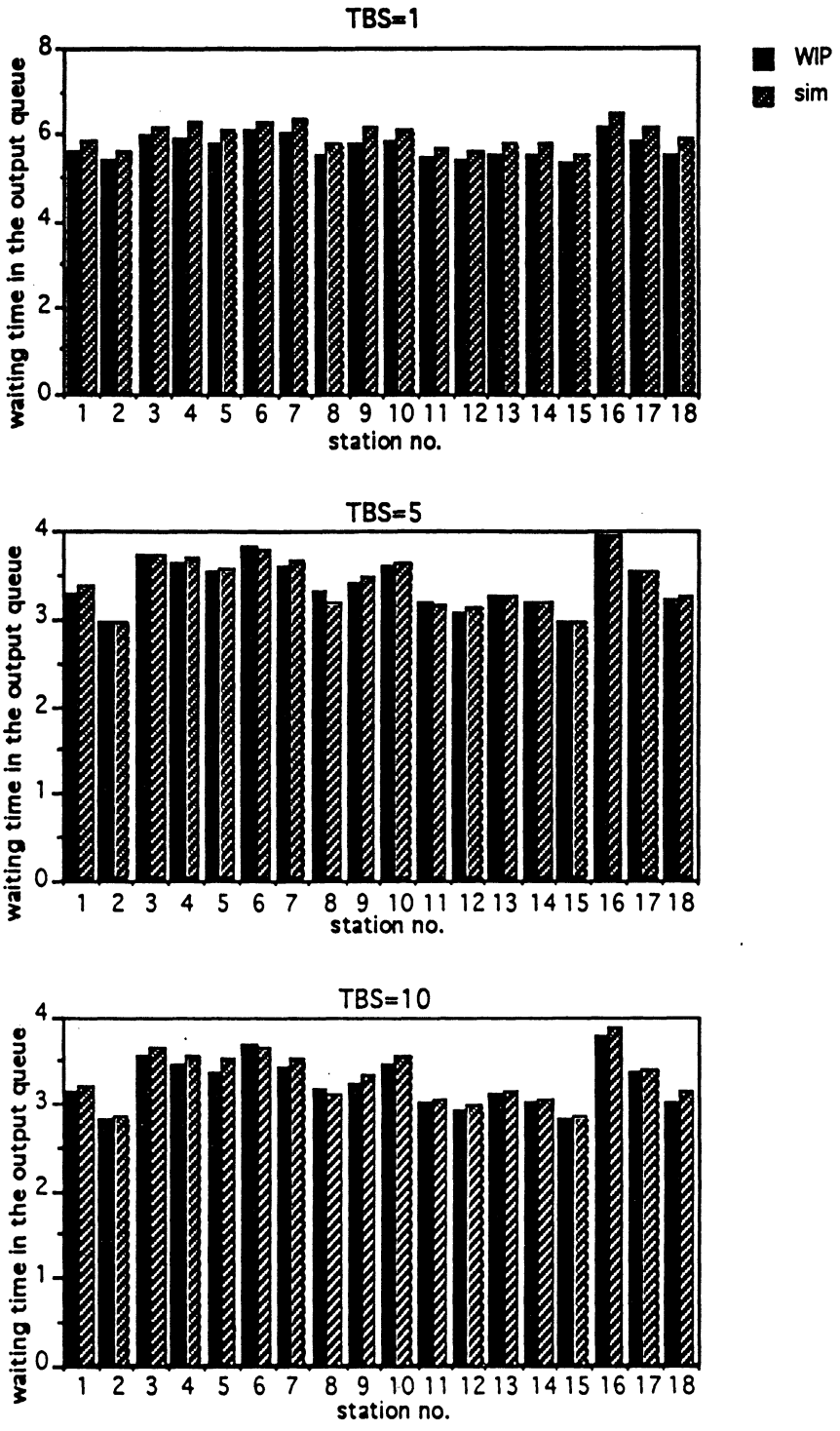


Figure A3: *Expected waiting times in the output queues, layout 2 with 3devices*

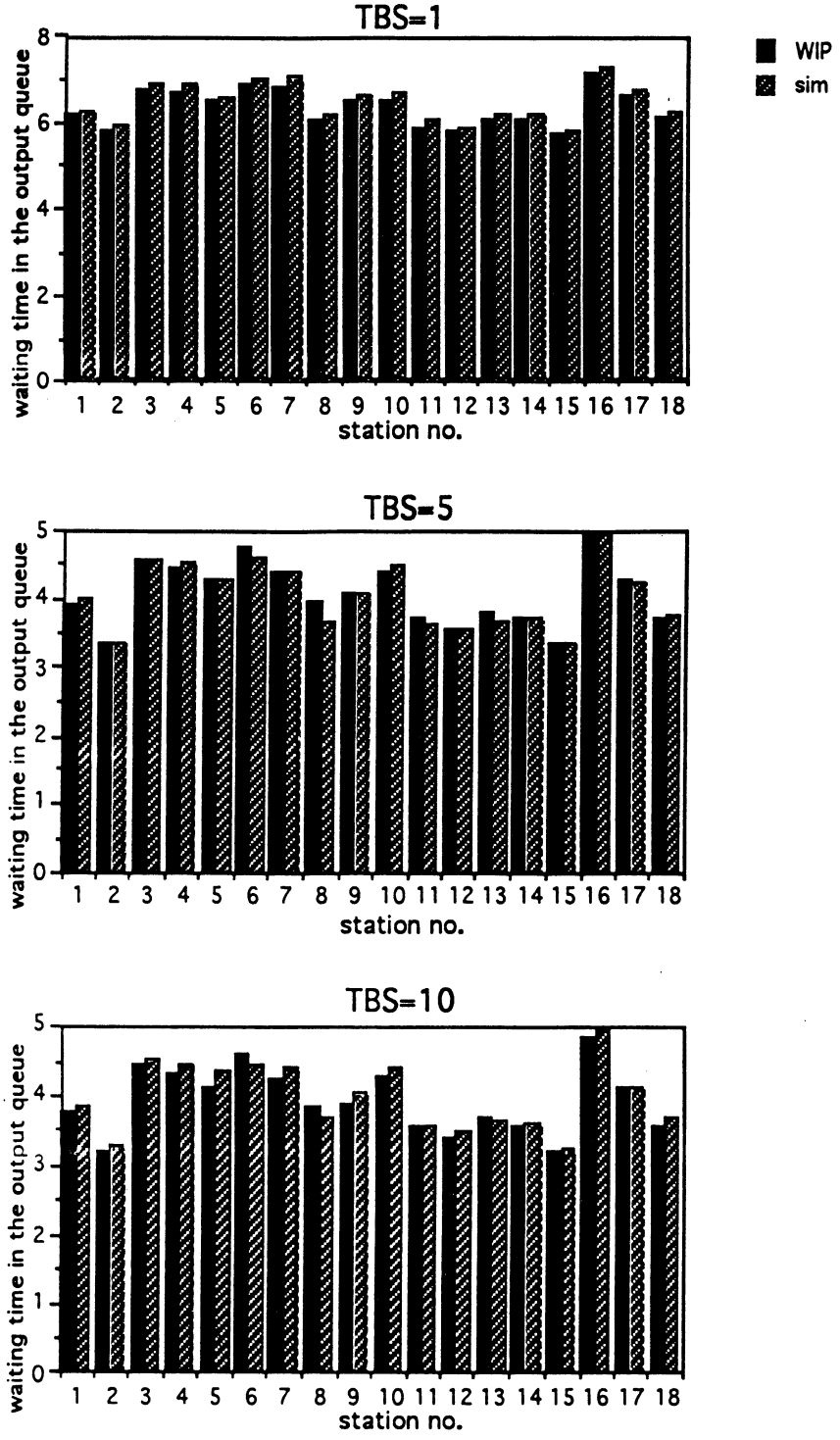


Figure A4: *Expected waiting times in the output queues, layout 2 with 5 devices*

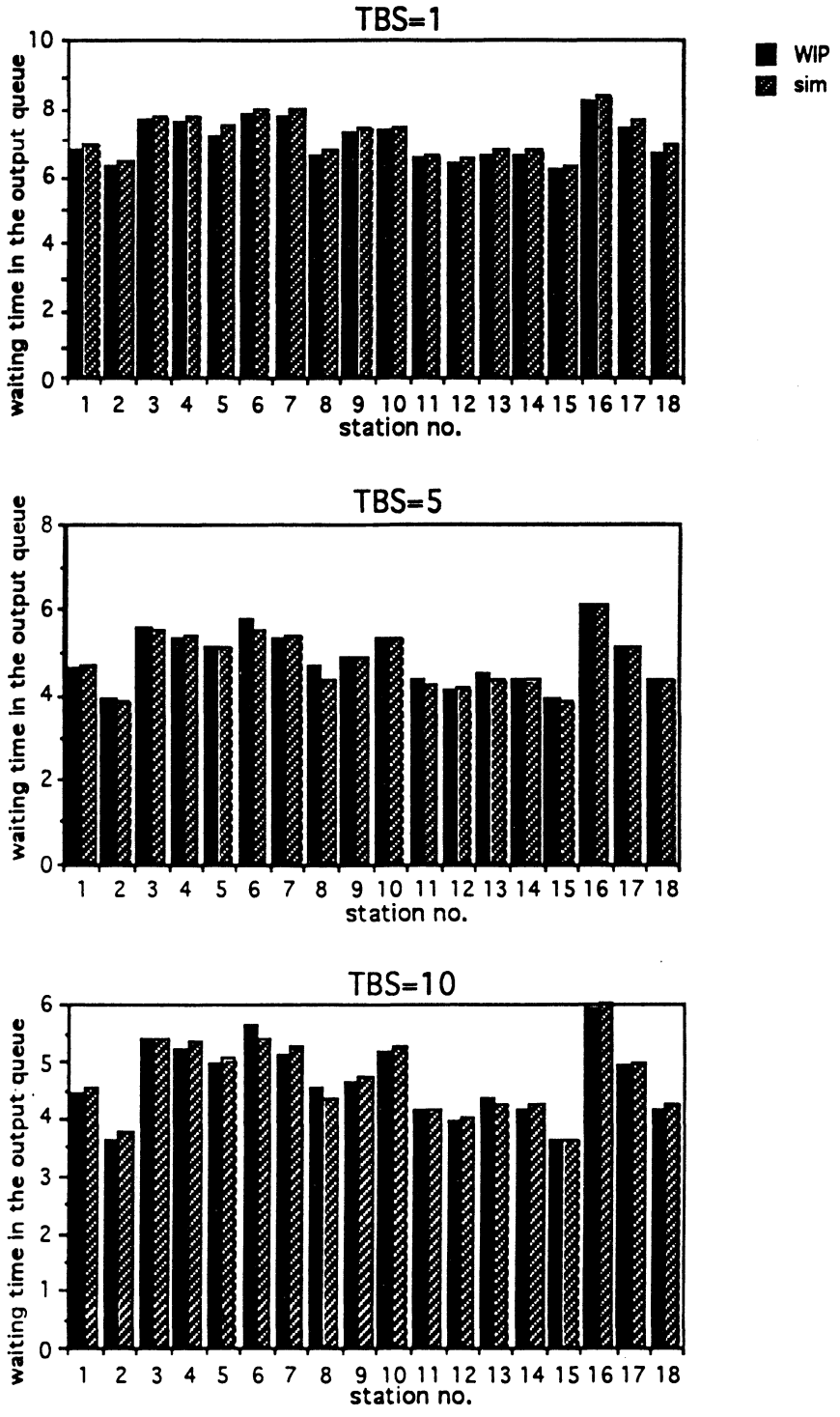


Figure A5: *Expected waiting times in the output queues, layout 2 with 7 devices*

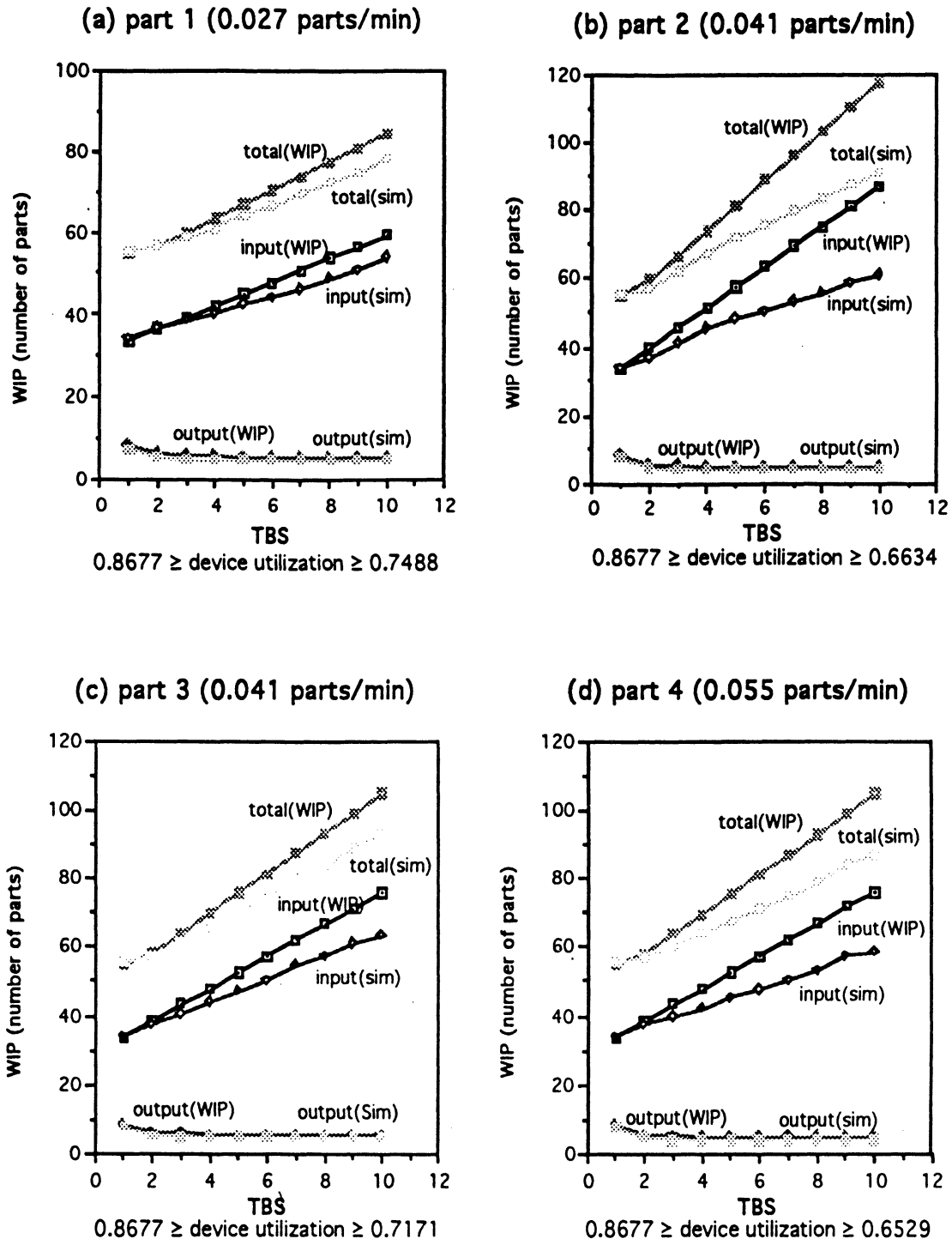


Figure A6: *Expected total WIP in the system, layout 2 eith 7 devices*

Table A2: *Job routes and throughput requirements #1, layout 3*

part type	parts/min	Job route 1.L3*
1	0.02	1 - 4 - 3 - 6 - 2
2	0.08	1 - 7 - 6 - 5 - 8 - 2
3	0.06	1 - 5 - 7 - 2
4	0.04	1 - 8 - 3 - 4 - 2
5	0.07	1 - 3 - 7 - 4 - 6 - 8 - 5 - 2

Device speed:from 270 (with 5 devices) to 150 (with 9 devices) distance units/min
P/D time = 8.34 (with 3 devices) to 15 (with 7 devices) secs
Processor utilization=0.75
*We changed station label in the original layout so that stations 1 and 2 become I/O stations.

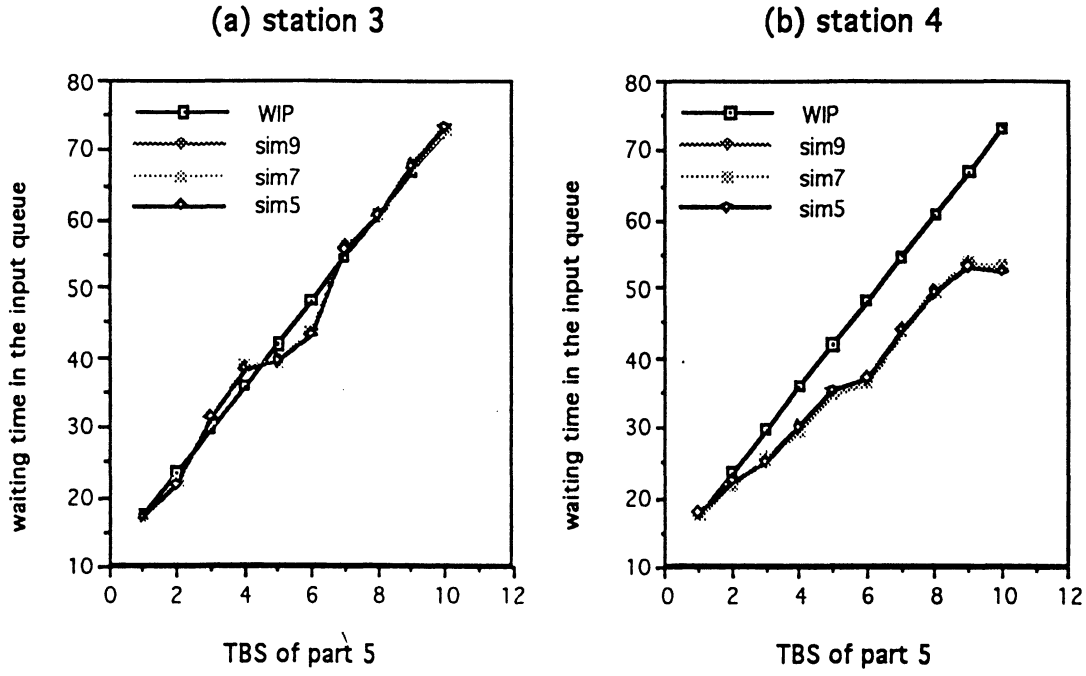


Figure A7: *Expected waiting times in the input queues, layout 3*

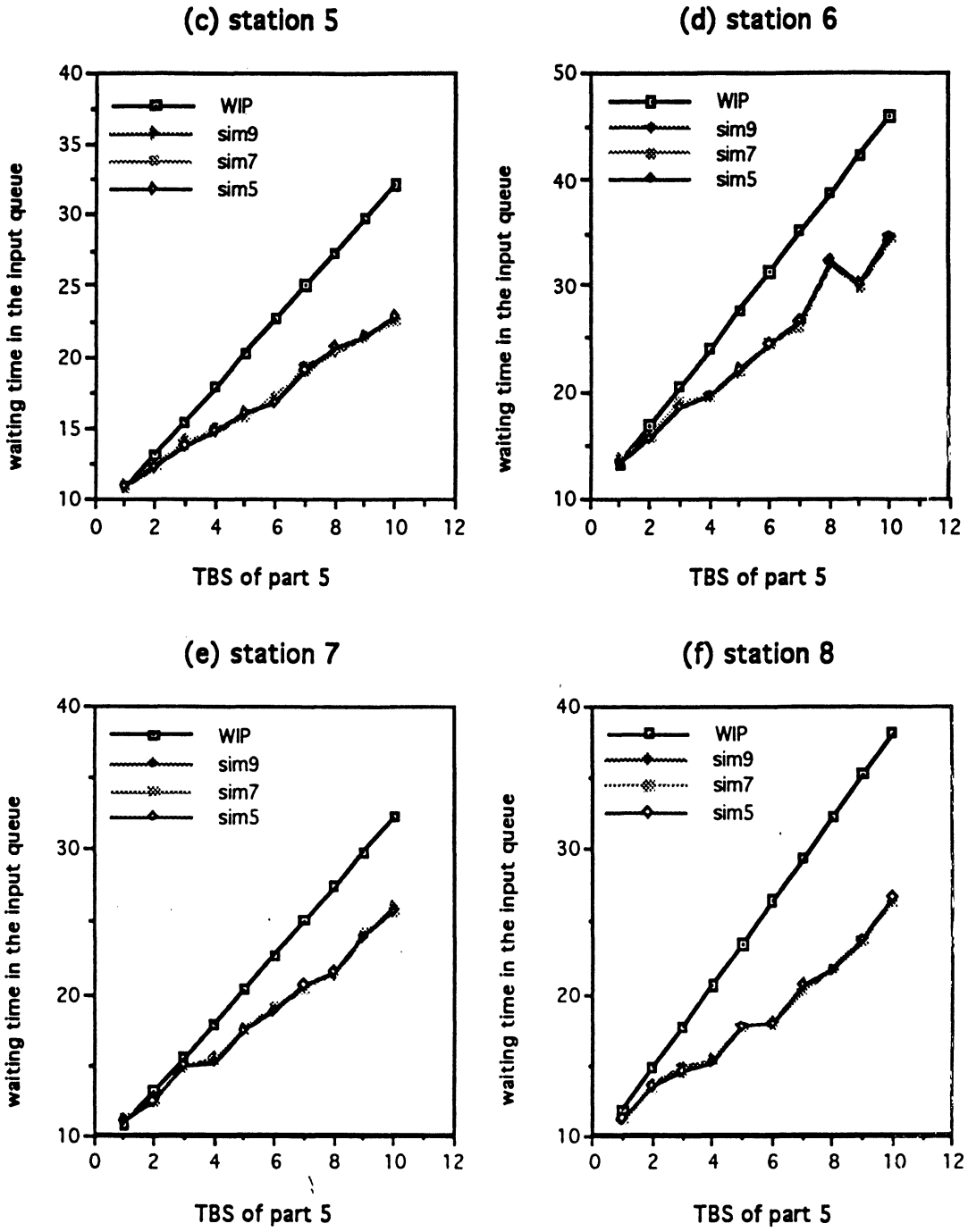


Figure A7: *(contd.) Expected waiting times in the input queues, layout 3*

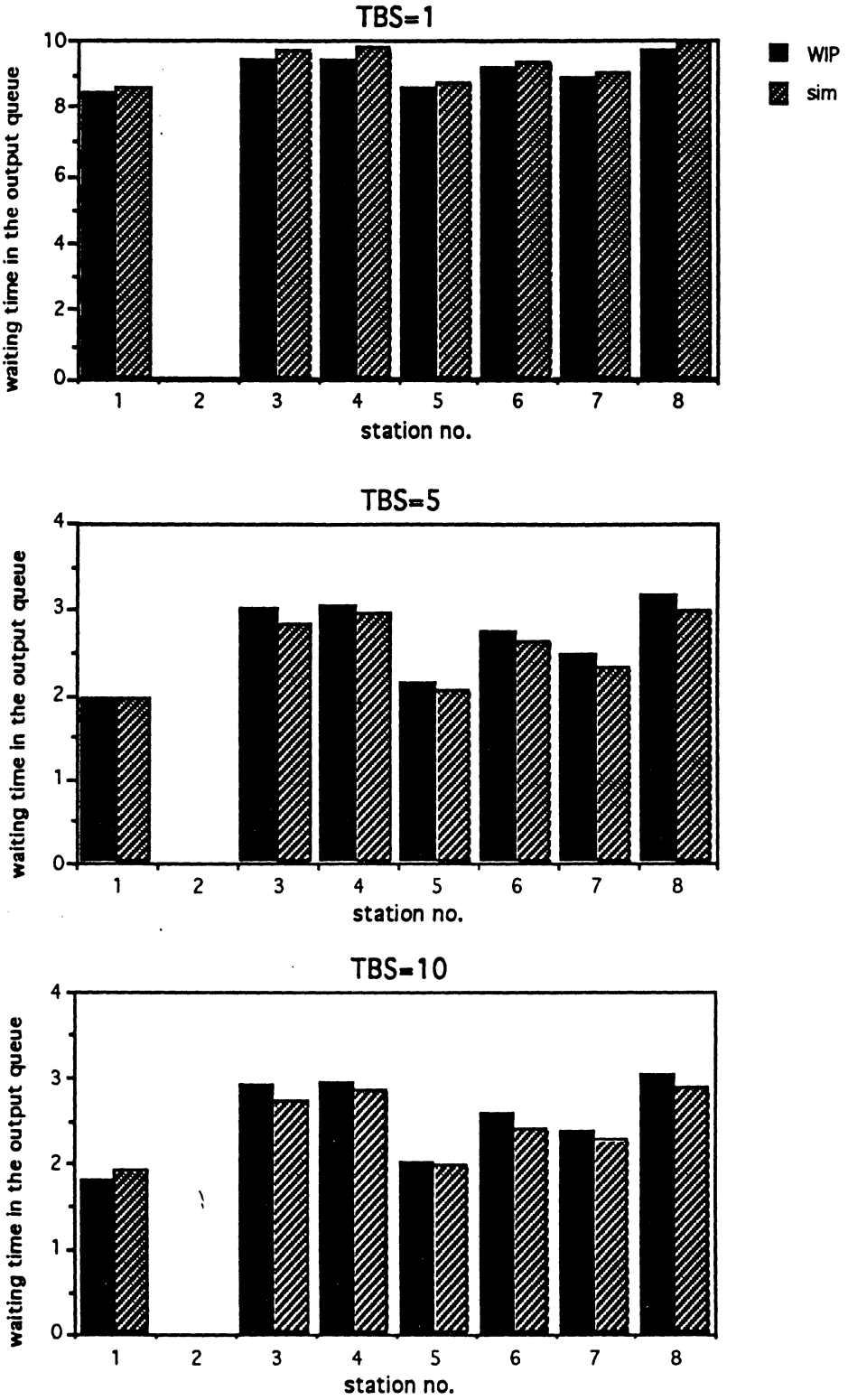


Figure A8: *Expected waiting times in the output queues, layout 3 with 5 devices*

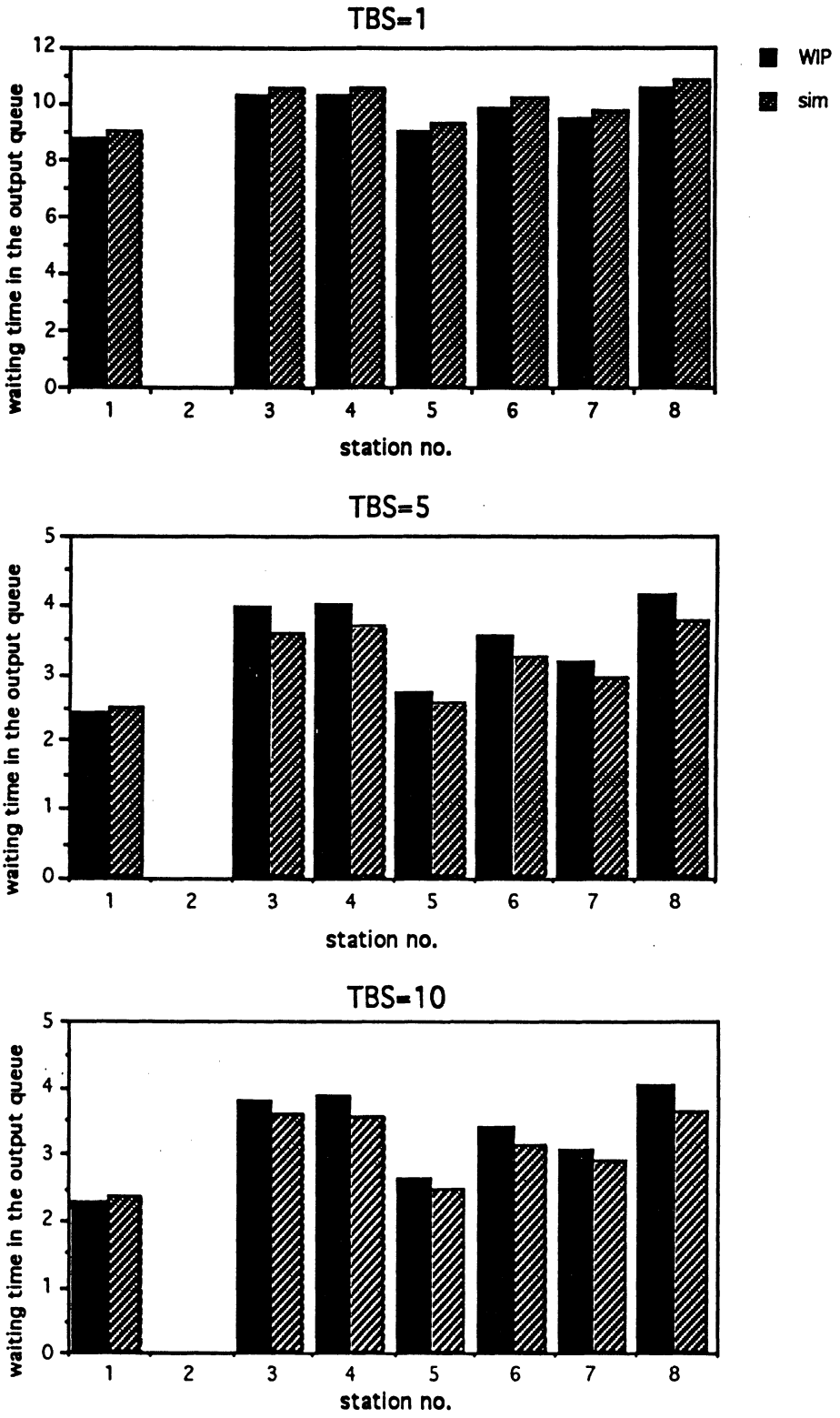


Figure A9: *Expected waiting times in the output queues, layout 3 with 7 devices*

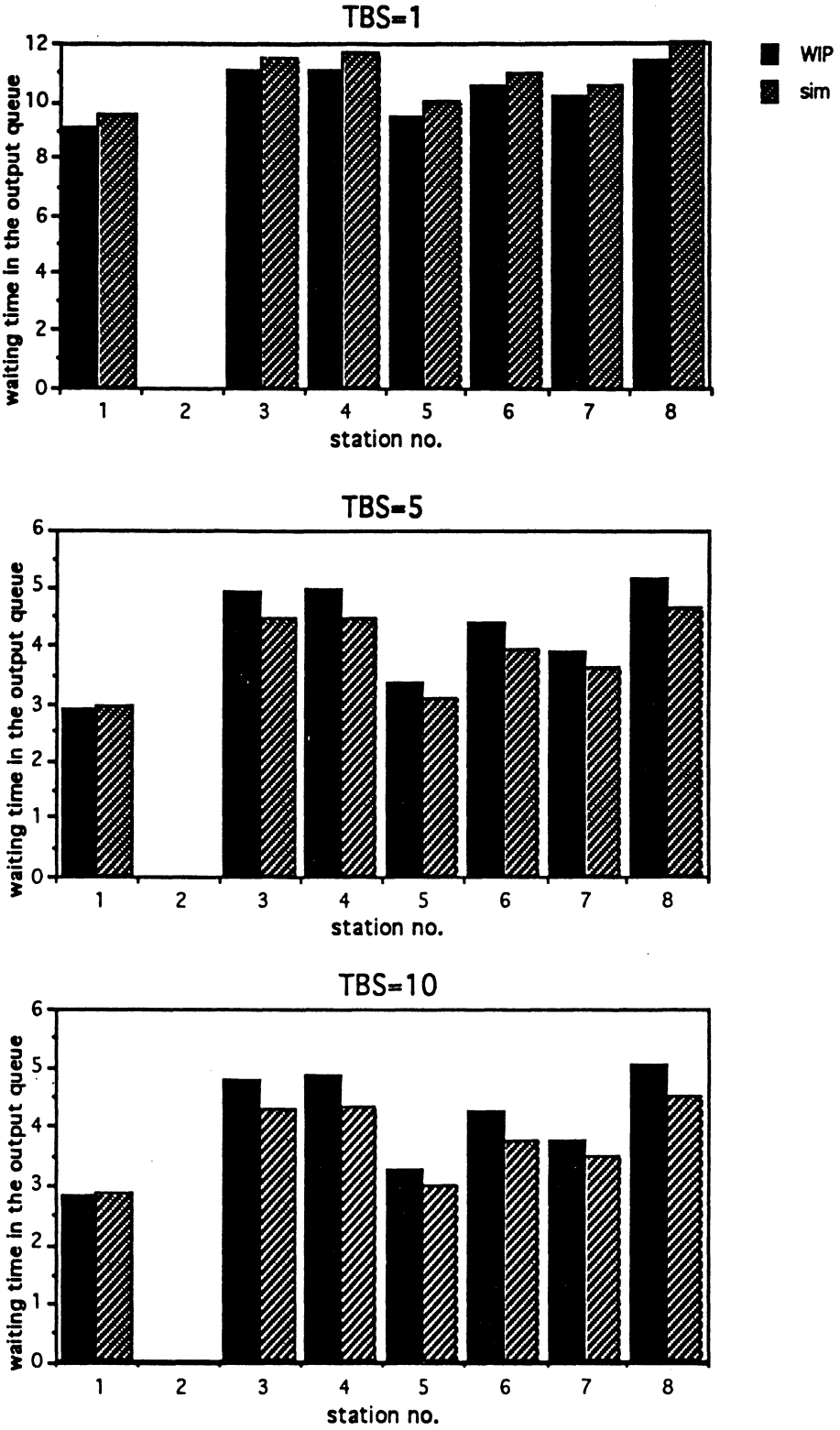


Figure A10: *Expected waiting times in the output queues, layout 3 with 9 devices*

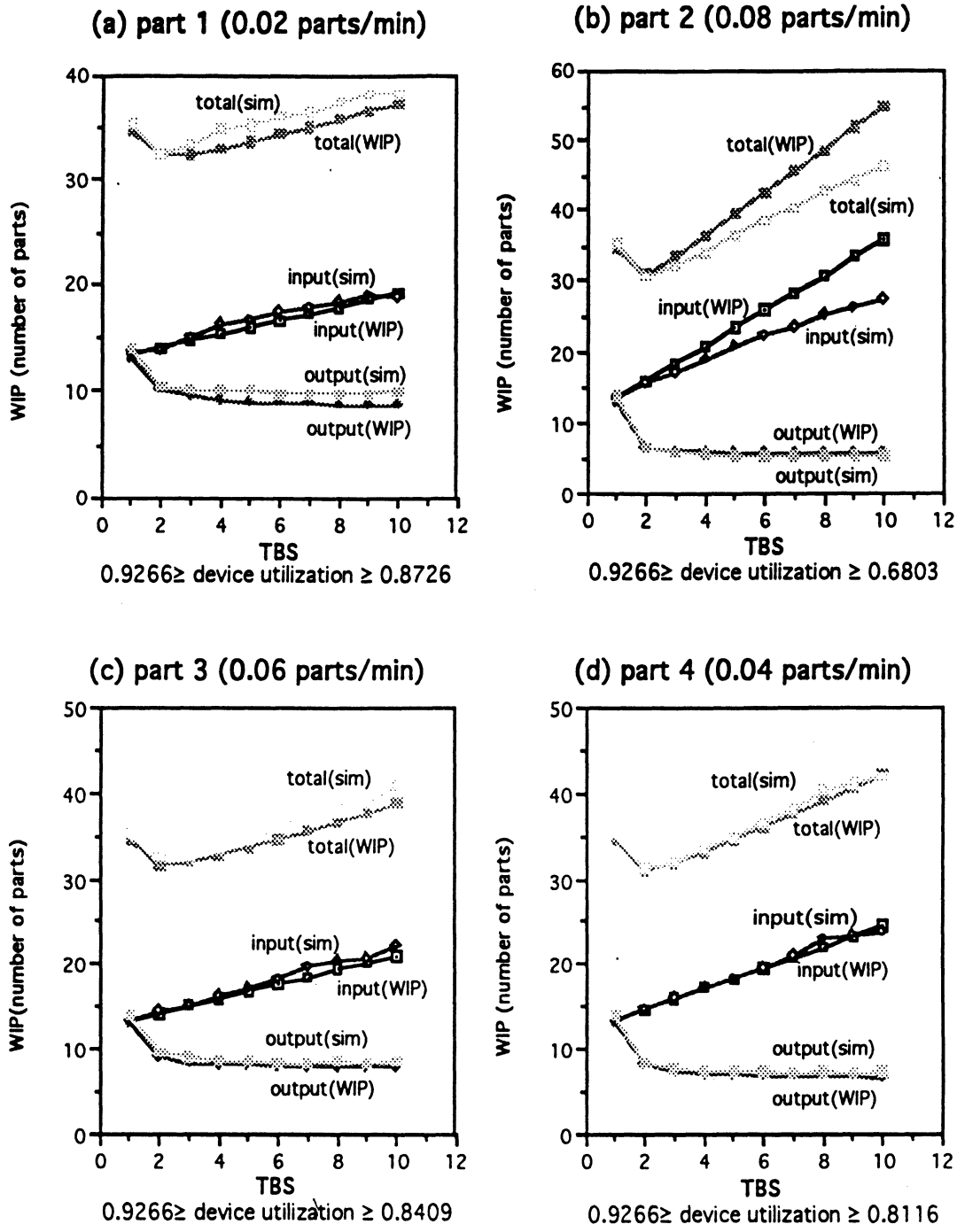


Figure A11: *Expected total WIP in the system, layout 3*

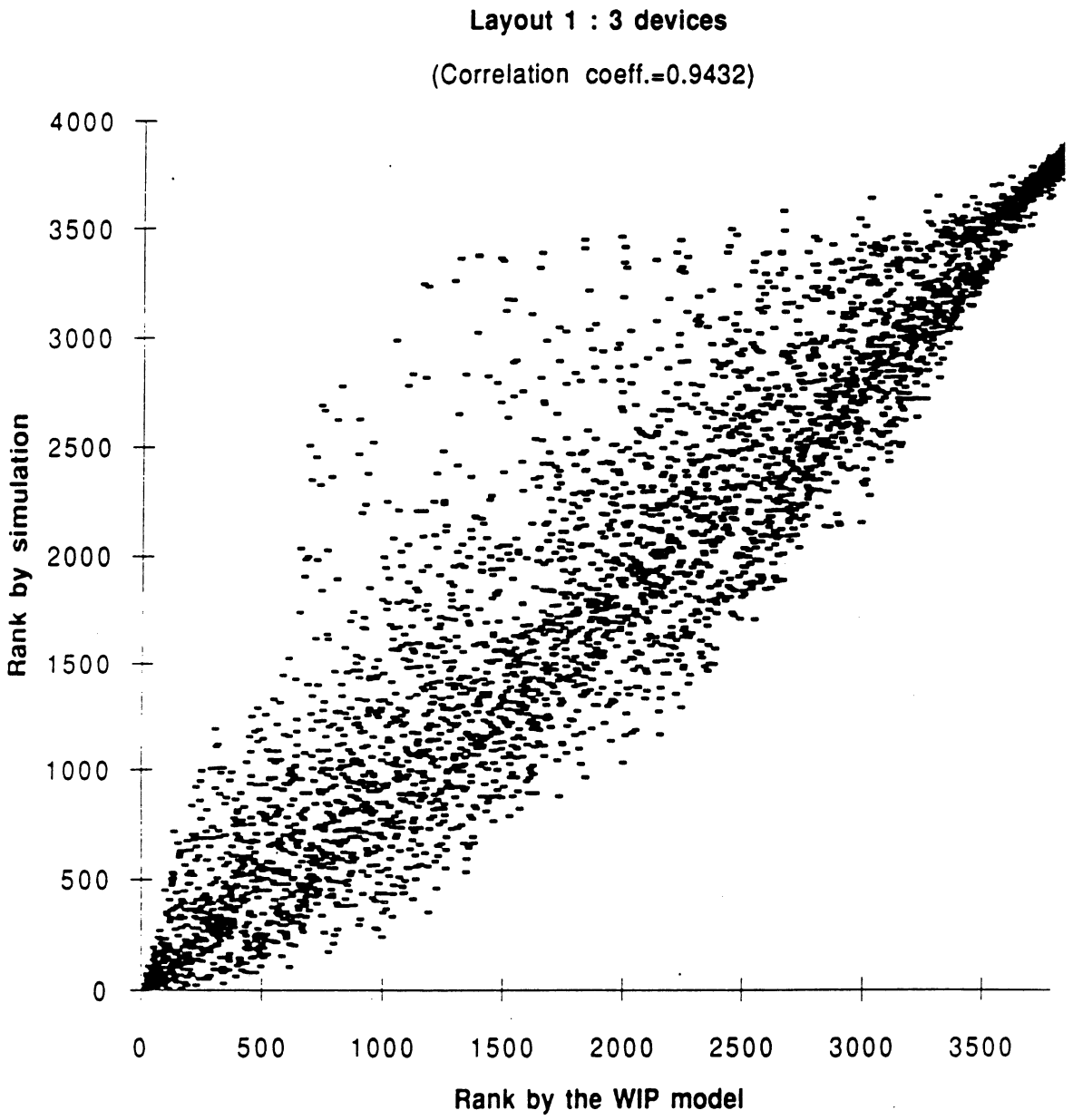


Figure A12: *Ranks by the WIP model and simulation, layout 1*

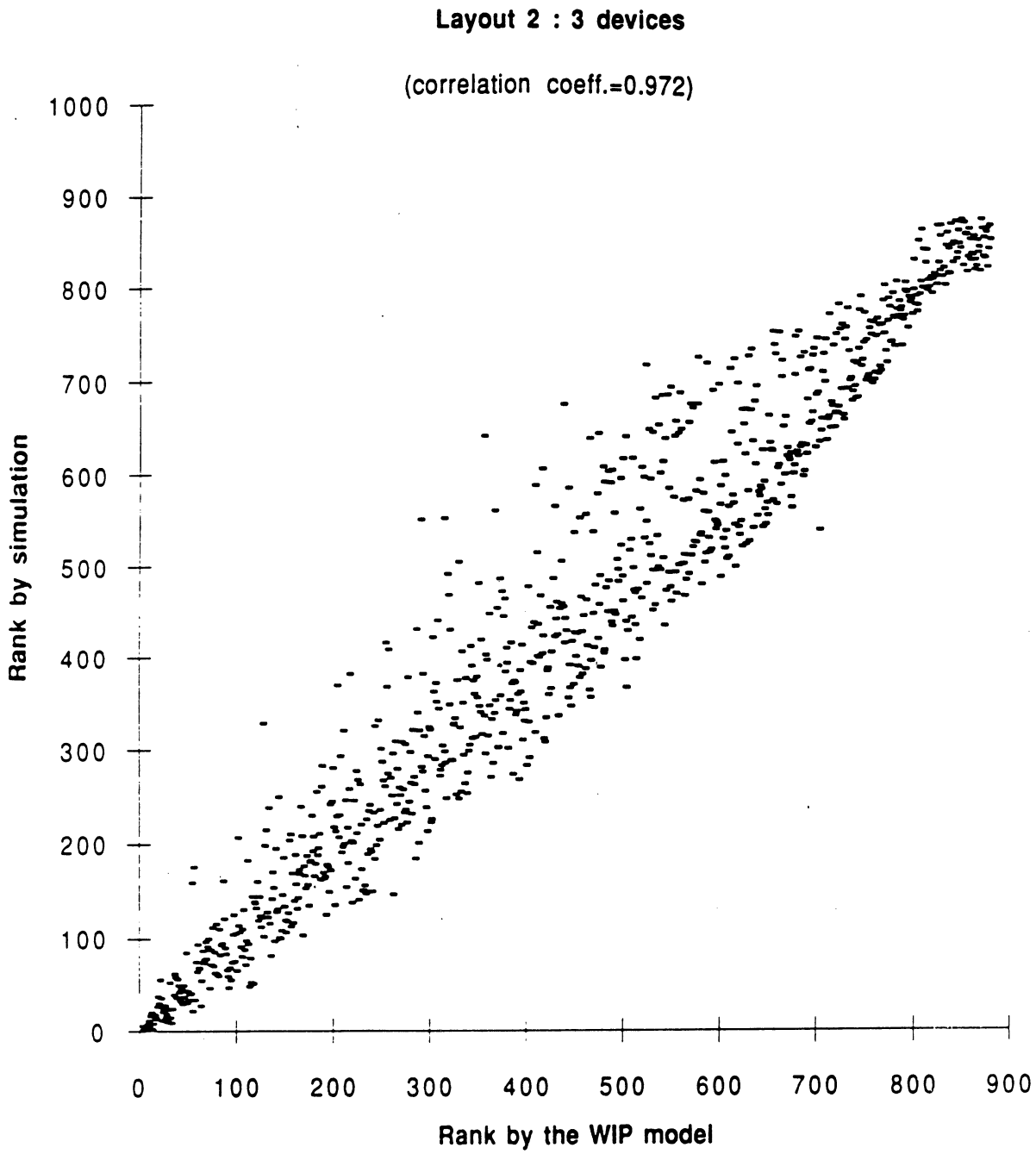


Figure A13: *Ranks by the WIP model and simulation, layout 2*

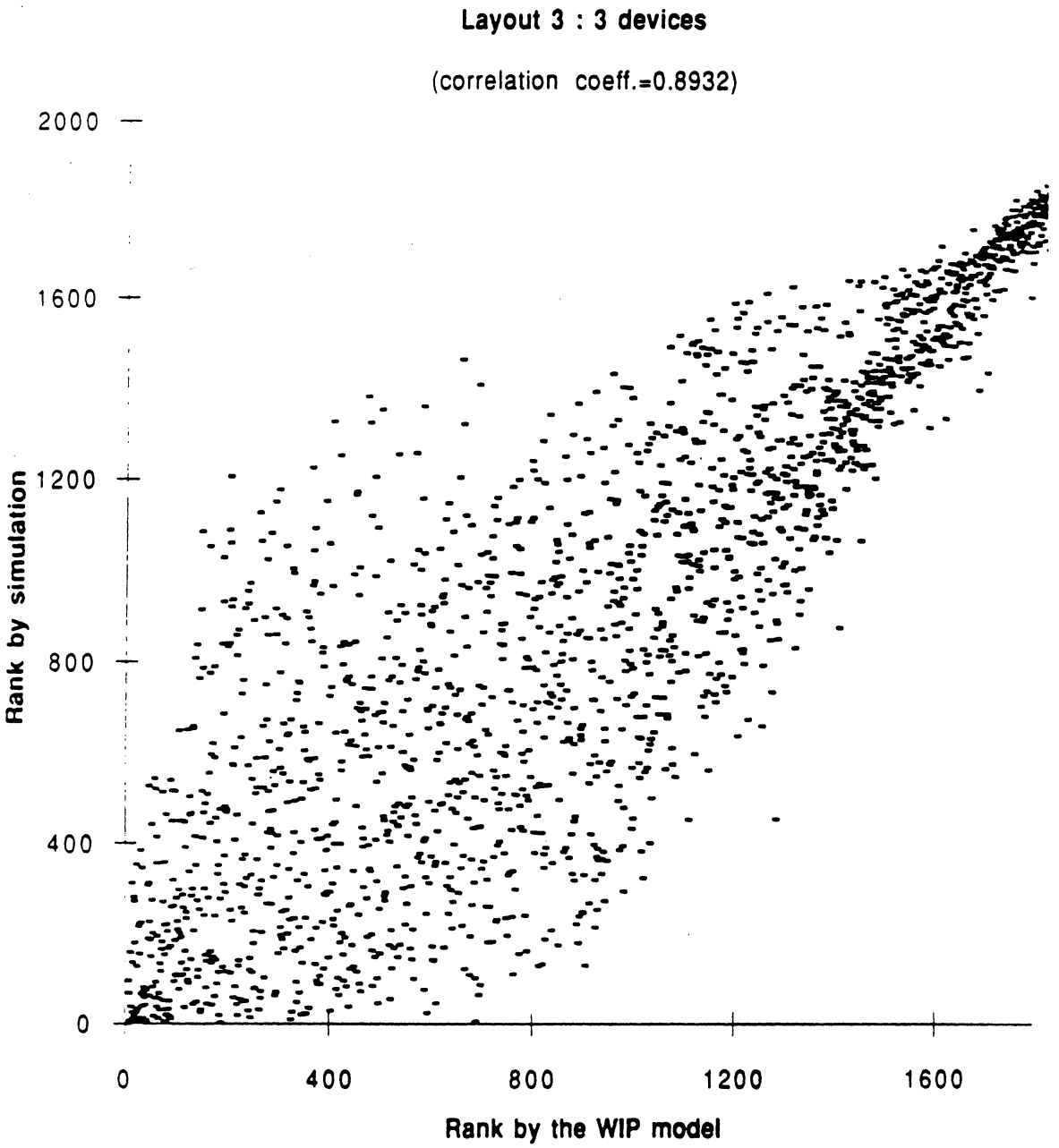


Figure A14: *Ranks by the WIP model and simulation, layout 3*