# INTELLIGENT DISPATCHING RULES FOR TRIP-BASED MATERIAL HANDLING SYSTEMS

Yavuz A. Bozer
and
Chih-kuan Yen
Department of Industrial & Operations Engineering
The University of Michigan
Ann Arbor, Michigan 48109-2117

Technical Report 95-16

September 1995

J. of Manufacturing syst.

# INTELLIGENT DISPATCHING RULES FOR TRIP-BASED MATERIAL HANDLING SYSTEMS[1]

Yavuz A. Bozer

Chih-kuan Yen

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

## ABSTRACT

It has been shown in the literature that the manner in which empty devices in a trip-based material handling system are dispatched affects the overall productivity of the handling system. Existing dispatching rules: 1. do not assign a move request until a device deposits a load and becomes empty; 2. with the exception of one rule, do not allow an empty device to be reassigned to another move request; 3. do not consider all of the available information (such as the status of other devices) before assigning a move request; 4. may lead to sizable empty travel (also known in industry as "dead heading") which reduces the efficiency of the system. In this paper, we present two new dispatching rules, namely, modified shortest-travel-time-first (MOD STTF) and bidding-based-dynamic-dispatching ($B^2D^2$), which address the above shortcomings and take advantage of information that should be readily available in centrally controlled systems. The latter rule also uses a new concept for device dispatching in that all the devices (empty or otherwise) are allowed to "bid" on incoming move requests. Simulation results indicate that, compared to existing rules, both MOD STTF and $B^2D^2$ improve system performance considerably.

---

[1] This study was partially supported by Dr. Bozer's Presidential Young Investigator Award under NSF Grant DDM-8858562.

# 1 Introduction and Assumptions

Material handling is an essential component of manufacturing. If the handling system does not deliver the right parts at the right time, it often leads to significant losses due to machine idling, missed due dates, and elevated work-in-process (WIP). Widespread use of just-in-time (JIT) deliveries on the factory floor has further increased the need for a responsive and efficiently run material handling system. However, since material handling is generally viewed as a "non-value added" operation, it typically receives less capital and less investment priority than production equipment. Consequently, coupled with the cost of modern handling equipment (such as automated guided vehicles or ergonomically designed lift trucks), many manufacturing concerns have a strong incentive to use their existing (or proposed) handling systems to their maximum efficiency.

The particular material handling system we address in this study is a trip-based material handling system. According to Srinivasan, Bozer, and Cho [1], a trip-based handling system consists of one or more self-powered devices that operate independently and asynchronously. To move a unit load, i.e., to serve a move request, the device must perform a trip, which is composed of empty travel (from the current location of the device to the station that placed the move request) followed by loaded travel. It is assumed that a device moves only one unit load on each trip. Examples of trip-based material handling systems include unit load automated guided vehicles (AGVs), lift trucks, cranes, freight elevators, and manual handling systems (where a person represents the device).

Consider the manufacturing system shown in Figure 1, where there is one input/output (I/O) station and three processor stations. The I/O station serves as an entry/exit point for the system while a processor station represents either a machine, a group of machines, or a processing department. (Of course, there may be more than one I/O station and all the processor stations need not be identical.) Each station has a dedicated *input queue* and *output queue*. Loads arriving at a station are deposited at the input queue of that station, where they wait to be processed. Loads that are ready to be moved are placed in the output queue of that station, where they wait to be picked up by a device. (Material handling *within* a station is not considered.) Loads that arrive from outside the system enter the output queue of an I/O station. Loads that require no further processing are delivered to the input queue of an I/O station where they are assumed to leave the system instantaneously. Note that an empty (loaded) trip occurs from the input (output) queue of one station to the output (input) queue of the other.

There are a number of issues involved in the design and operation of a

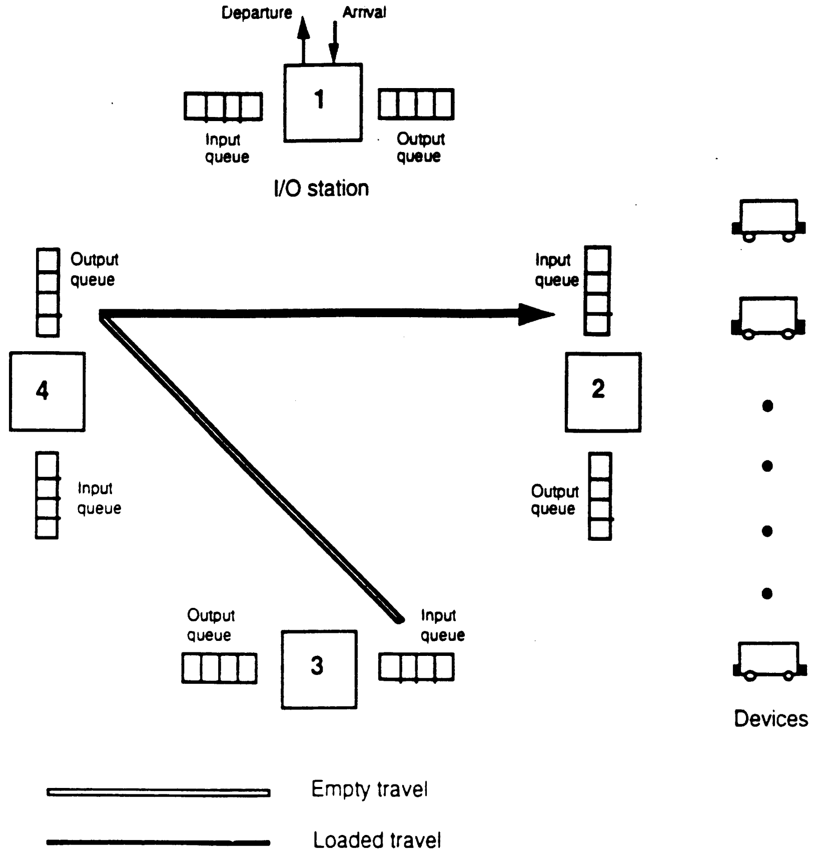


Figure 1: A Trip-Based Material Handling System

trip-based material handling system. The most notable design issues include the number of devices required and flow path design, while one of the key operational issues is device dispatching, which is basically concerned with assigning each move request to one of the devices. (Depending on the particular type of trip-based handling system involved, there could be other operational issues ranging from operator fatigue to battery charging; such issues are neyond the scope of our study.) Existing dispatching rules in the literature — and those typically used in industry — wait until a device becomes empty before making the above assignment. Therefore, device dispatching has generally been referred to as *empty* device dispatching. In this study, we take a more general view of dispatching and do not necessarily wait until a device is empty before assigning a move request to it.

Assuming that the flow path and the flow volumes are given, and that the empty or loaded travel routes among the stations are defined, the primary objective in device dispatching has been to reduce empty travel. Note that, with the above assumptions, total *loaded* travel per time unit that must be performed by all the devices is given and fixed. Hence, the efficiency of the handling system, and the resulting expected waiting times of move requests at the output queues, to a large extent depends on how well empty device travel is managed. Although empty trips are unavoidable in virtually all trip-based handling systems, empty travel is unproductive and it increases the time it takes for a device to serve a move request.

We make the following remaining assumptions for the study:

1. Multiple job types are processed through the system. Each job type, which is characterized by the sequence of stations it visits, is assumed to be of equal priority. The arrival rate of each job

type entering the system is fixed.

2. Devices follow a well-defined flow path (or aisle structure) while traveling; i.e., the path from one station to another is known and given. An "intersection" is formed when two flow paths (or aisles) cross each other.
3. The (empty or loaded) device travels at a constant speed independent of the job type it serves.
4. The devices do not interfere with one another; any congestion that may exist is negligible.
5. The processing time associated with each job on each processor is a random variable with a known distribution.
6. Loads in each input queue are processed on a FCFS basis.
7. The distance from the input queue to the output queue of a station is negligible.
8. Devices and machines never break down.

We present the above list to fully define the system we later simulate. As it will become evident in section 3, some of the above assumptions are made only for simplicity and they are *not* an inherent part of the two dispatching rules we present in this paper, namely, MOD STTF and $B^2D^2$. Since these rules (particularly $B^2D^2$) deviate significantly from existing rules and they attempt to use existing information more effectively, we refer to them as "intelligent" dispatching rules (although, strictly speaking, we have not used artificial intelligence techniques).

The remainder of the paper is organized as follows: in section 2 we review existing dispatching rules in the literature. In section 3, we describe the MOD STTF rule and the $B^2D^2$ rule. Simulation results aimed at comparing the performance of the above two rules against one another and against existing rules are presented in section refsec-pe. Lastly, we summarize the results and discuss possible future research directions in section 5.

## 2 Literature Review

There are a number of papers concerned with developing empty device dispatching rules. Egbelu and Tanchoco [2] proposed and evaluated the modified-first-come-first-served (MFCFS) rule, among others. The MFCFS rule differs from the first-come-first-served (FCFS) rule in that only the load at the head of an output queue can issue a move request from that station. (Note that MFCFS does not attempt to reduce empty device travel; rather, it seems concerned with allocating the material handling capacity more equitably among the stations.) Using simulation, the authors compared the performance of

MFCFS with other rules such as random-work-center (RW), shortest-travel-time-first (STTF), longest-travel-time-first (LTTF), and maximum-outgoing-queue-size (MOQS). Simulation results suggest that the MFCFS rule is superior to other dispatching rules if the input and output queue sizes are finite, whereas, the STTF rule performs competitively with MFCFS if queue characteristics are not considered. However, the results are based on only one layout with no replications, and the performance measures are not comprehensive. For example, the expected WIP in the input and output queues, and the expected device utilization, are not reported.

Also, under STTF, some output queues backed up since a device would deliver a load at station $i$ and then serve another station even if there was an unassigned move request waiting at the output queue of station $i$. This was primarily due to the layout and the fact that the output queue was not necessarily located close to the input queue of a station. Such results show that the performance of the STTF rule can be layout dependent and some stations may be "orphaned."

Simulation was also used by Russell and Tanchoco [3] to evaluate four empty vehicle dispatching rules: largest number in queue (LNQ), longest waiting time (LWT), preferred order by nearest load (POR), and random assignment (RAN). The results indicate that the LWT rule had the smallest mean flow time while the LNQ rule reduced the average maximum queue length. However, in both measures the best rule did not show a significant difference over other rules, including STTF.

Hodgson, King, and Monteith [4] use a Markov Decision Process (MDP) to develop a control strategy, namely, RULE, for material handling systems with one unit load or double-load AGV. RULE is based on the characterization of the optimal control policy obtained from the MDP. Under RULE, when a device becomes empty, a score is computed for each move request and the empty device is assigned to the move request with the maximum score (we refer the reader to [4] for details). An empty device assigned to a move request can be reassigned to another move request since the above score is recomputed at each station the empty vehicle must pass through.

The score computed for each move request is partly determined by three parameter values specified by the user. The performance of RULE is likely to be affected by these parameter settings, which are difficult to generalize. The authors use simulation to compare RULE with STTF for small systems with one unit load or one double-load AGV. For a unit load AGV, the results indicate that RULE is at most 1.20% better than STTF in terms of throughput and it improves the average output queue length by at most 5.4%. In a subsequent study, King, Hodgson, and Monteith [5] use simulation to further compare RULE with STTF in layouts with 7 to 14 stations and one or two AGVs. The maximum improvement in the average (output) queue length they report is approximately 14%.

Bartholdi and Platzman [6] proposed and analyzed a decentralized dispatching rule, namely, first-encountered-first-served (FEFS), for AGV systems where all the stations are arranged around a simple closed loop. Under FEFS, an empty AGV travels along the loop to poll the stations and it serves the first move request it encounters. For a static system (where a fixed set of move requests are given a priori), the authors show that, under FEFS, the AGV will travel no more than once around the loop beyond the optimal number of revolutions required to serve all the move requests. Under dynamic conditions, they showed that the AGV will travel no more than twice the minimum number of revolutions possible. Although the FEFS rule is a simple, decentralized rule that may be effective in closed loop systems with one or few AGVs, it is not as effective as centralized rules used in general (i.e., non-circular) configurations (see, for example, Egbelu and Tanchoco [2[illegible]

Han and McGinnis [7] developed the most significant move (MSM) rule to dispatch a microload storage/retrieval (S/R) machine with finite input and output buffers. The MSM rule uses known processing times to determine the due date at a given pickup station and computes a priority index for each station. The pickup station with the highest priority index is then selected. The performance of MSM was compared against the MFCFS and STTF rules via simulation. The authors empirically showed that, with buffer sizes of one unit load each, the MSM rule performs better in terms of throughput. However, when the buffer size is increased, the throughput difference between the above rules is reduced. The MSM rule was not evaluated in a non-deterministic setting with multiple devices (since microload AS/R systems contain a single S/R machine in each aisle).

Srinivasan, Bozer and Cho [1] proposed an alternate modification to the FCFS rule, namely, the MOD FCFS rule, where they attempt to reduce unnecessary empty device travel that results with the FCFS rule. Under the MOD FCFS rule, upon delivering a load at station $i$, the empty device first inspects the output queue of station $i$. If its empty, the device serves the "oldest" move request in the system (as in FCFS); otherwise, the device is assigned to the load at the output queue of station $i$. The authors present an analytical model to estimate system throughput under the MOD FCFS rule. According to their simulation results, the MOD FCFS rule outperformed the FCFS and MFCFS rules, and it was comparable to the STTF rule.

Sabuncuoglu and Hommertzheim [8] proposed a dynamic dispatching algorithm (DDA) for scheduling machines and AGVs in a FMS with finite queue capacities. The DDA rule consists of four hierarchical levels to assign jobs to AGVs. The four levels are: 1. identifying stations whose input and output buffers are full, 2. identifying the load in the central buffer area with the most available space in the destination input queue, 3. identifying the starving workstations, 4. identifying a load which has the highest chance of being processed earliest at the next workstation. To schedule a job on a machine, a priority index, based on known processing times and two subjective scaling factors, is computed for each candidate job in the input queue.

The authors use simulation to evaluate the performance of DDA against two alternative machine/AGV scheduling rules, namely, shortest-processing-time (SPT)/STTF and SPT/MOQS. They observed that when the machine utilization is over 80%, the DDA rule has the smallest mean flow time. However, they did not investigate whether the improvement in mean flow time was a result of better machine scheduling or better AGV dispatching. Furthermore, the jobs are assigned to the AGVs only one at a time.

In short, existing device dispatching rules: 1. do not assign a move request until a device deposits a load and becomes empty; 2. with the exception of RULE, do not allow an empty device to be reassigned to another move request (reassignment under RULE is considered only at the stations an empty device must pass through); 3. do not consider all of the available information (such as the status of other devices, which should be readily available in centrally controlled systems) before assigning a move request; 4. may lead to sizable empty travel (also known in industry as "dead heading").

Furthermore, results in the literature indicate that, except possibly for special cases — such as the single-device microload AS/R system or a layout where certain stations may be inadvertently "orphaned" — the STTF rule generally performs better than the other rules. Along with FCFS, the STTF rule is also one of the rules used frequently in industry (particularly in AGV systems). Hence, in this study we will use the STTF rule as our "benchmark." (For small systems with few stations and one device, RULE outperforms STTF by a small margin. Since the parameter settings in RULE are specified by the user, we do not know how RULE would compare with STTF in general. However, since we were able to obtain sizable improvements over STTF, we will not compare our results with RULE.)

# 3 Dispatching Rules: MOD STTF and $B^2D^2$

In this section we present two new dispatching rules, namely, modified shortest-travel-time-first (MOD STTF) and bidding-based-device-dispatching ($B^2D^2$) to improve the efficiency of trip-based material handling systems by reducing empty device travel. As the name implies, MOD STTF is a modification of the STTF rule, which is a commonly used rule in industry (particularly in AGV dispatching). In contrast, $B^2D^2$ represents a new paradigm for device dispatching – it is based on each device (empty or otherwise) placing a "bid" for each move request as the move requests arrive at the output queues.

## 3.1 The MOD STTF Rule

The MOD STTF rule is similar to the STTF rule in that: 1. it assigns empty devices to move requests based on the location of the device relative to the move request(s), and 2. a device may be assigned to at most one move request at any given instant. However, MOD STTF deviates from STTF in two significant aspects: 1. an empty device may be reassigned to another move request, and 2. one empty device may "release" another empty device. Also, under MOD STTF, we investigate the impact of "parking" an idle device at a "strategic" point rather than letting the device remain idle at its last location.

To describe the MOD STTF rule, we first need to define "committed" and "uncommitted" empty devices. Suppose that device $d$ has just become empty at station $i$. Further suppose that the closest move request is at station $j$. If the distance from station $i$ to station $j$, say, $\tau_{ij}$, is less than or equal to a threshold value (that we define later), then device $d$ is committed to the move request at station $j$ and vice versa (i.e., the move request at station $j$ is committed to device $d$). If $\tau_{ij}$ is greater than the threshold, however, then device $d$ is assigned to the move request at station $j$ but there is no commitment; i.e., device $d$ starts traveling empty towards station $j$ without having committed itself — we refer to this as "uncommitted empty travel." Likewise, the move request at station $j$ is assigned but not committed to device $d$.

Note that, under the above scenario, when device $d$ becomes empty at station $i$, it will consider all the move requests in the system except the committed ones. Also, at any given instant: 1. a move request is either committed or uncommitted, and 2. an uncommitted move request is either assigned to a device or it is not. Likewise, at any instant, a device may be: 1. traveling loaded (including the pick-up/deposit operations), 2. traveling empty and committed, 3. traveling empty but uncommitted, 4. traveling empty to a parking point for idle devices, or 5. parked and idling at the parking point. A device cannot be assigned to a move request if all the output queues are empty or all the move requests in the system are committed; i.e., an unassigned device is defined as an idle device, whether it is traveling or parked. Of course, an unassigned device is uncommitted, by definition.

Consider next the scenario where device $d$ is traveling empty but uncommitted towards the move request at station $j$ (which we will label as move request $j$ for brevity). On its way to station $j$, if device $d$ passes through a station or an intersection, then it will reconsider its assignment; i.e., it will consider all the uncommitted move requests in the system (assigned or otherwise, including those that may have arrived after it was assigned to move request $j$) and identify the closest one.

Suppose the closest move request is at station $k$ ($k \neq j$). Further suppose that device $e$ is traveling empty but uncommitted towards move request $k$. If device $e$ is closer to station $k$ than device $d$, then device $d$ ignores move request $k$ and considers the next closest uncommitted move request. Otherwise, device $d$ is assigned to move request $k$ and device $e$ is "released," i.e., it becomes unassigned. Hence, under MOD STTF, an uncommitted move request may have at most one device traveling empty but uncommitted towards it. Likewise, no move request may be committed to more than one device and vice versa.

Obviously, the distance threshold used with MOD STTF (in deciding to commit a device or not) plays a key role. Instead of requiring a user-specified distance threshold, we propose to use distance information obtained directly from the layout. One possible approach is to sort the distances between all pairs of stations (in non-decreasing order) and use, say, the 60th percentile as the threshold. That is, if the distance from the empty device to the closest move request is greater than 60% of all the distance values in the layout, then the device would be assigned but not committed to it. An alternate approach is to set the threshold distance relative to the average distance traveled per loaded trip (which is straightforward to obtain from the data). For example, if the distance to the closest move request is greater than the average loaded trip, then the device would be assigned but not committed to it. We considered both of the above types of threshold values in our simulation experiment; the results will be shown in section 4.

In the above description of MOD STTF, a device reconsiders its assignment only when it travels through a station or intersection. Obviously, if all the (uncommitted) devices were allowed to reconsider their assignments every time a new move request arrived, MOD STTF is likely to perform better. However, considering that a device can change its route only at an intersection, we do not believe that the additional complexity generated by such a change would be justified (especially in systems where the flow path is unidirectional).

Letting TH denote the distance threshold, the MOD STTF rule can be formally presented as follows: whenever a move request arrives at, say, output queue $i$, the following sequence is executed:

1. If there are no idle (i.e., unassigned) devices, STOP. Otherwise, consider each unassigned device (traveling or parked), and label the closest one as device $d$. (If an unassigned device is traveling, the distance to output queue $i$ is measured from the next intersection on the device's route.) Let the distance from device $d$ to output queue $i$ be $\tau_i$.

2. If $\tau_i \leq TH$, then commit device $d$ to move request $i$ and vice versa; STOP. Otherwise, assign device $d$ to move request $i$ and initiate uncommitted empty travel; STOP.

Whenever: 1. a loaded device, say, device $d$, deposits a load at, say, station $i$; 2. an uncommitted (empty) device, say, device $d$, reaches a station or an intersection (say, point $i$); or 3. an uncommitted device at point $i$ is released by another device; then the following sequence is executed:

1. If there are no uncommitted move requests (or they have all been already considered), dispatch device $d$ to the parking point for idle devices; STOP. Otherwise, identify the (next) closest uncommitted move request, say, move request $j$, and denote its distance to point $i$ as $\tau_{ij}$.

2. If there is no empty device already assigned to move request $j$, go to 3. Otherwise, denote the assigned empty device by $e$ and go to 4.

3. If $\tau_{ij} \leq TH$, then commit device $d$ to move request $j$ and vice versa; STOP. Otherwise, assign device $d$ to move request $j$ without committing

either one and initiate uncommitted empty travel; STOP.

4. Let the next intersection on device $e$'s route be denoted by $k$. Also, let the distance from intersection $k$ to move request $j$ be denoted by $\tau_{kj}$. If $\tau_{ij} \geq \tau_{kj}$, go to 1. Otherwise, release device $e$ and go to 3.

Under MOD STTF, idle devices wait at a parking point. An obvious parking point is the station where the device delivered its last load. An alternate approach is to park idle devices at stations that place the most move requests; that is, idle devices can be dispatched to those stations with the highest move request arrival rates. Even if idle devices are somewhat evenly allocated among such stations, our preliminary simulation results indicated that such a parking strategy has little or no noticeable impact on system performance; therefore, we did not further pursue this approach.

A third approach is based on parking all the idle devices at a single point, say, point $p$, such that the expected empty travel time from point $p$ to any station in the system is minimized. Using the move request arrival rate at each station as a "weight," and rectilinear travel distances, it is straightforward to compute the optimum location of point $p$, which is a "minisum" location (see [9]). Of course, if point $p$ does not overlap with the flow path, one needs to construct "contour lines" [9]. For simplicity, however, instead of using contour lines, we picked the closest station (to the optimal minisum location) as the parking point.

While it may not be desirable to send all the idle devices to the same point, in a properly designed trip-based handling system, very seldom one would find many idle devices waiting at the parking point. (If this is not the case, then there is excess device capacity and dispatching/parking strategies are not as critical.) We also stress that, an idle device may be assigned or committed to a move request if it arrives before the device reaches the parking point. In section 4 we will show the impact of the above "minisum parking" strategy.

## 3.2 The $B^2D^2$ Rule

Existing dispatching rules wait until a device becomes empty and then assign only one move request to it. While MOD STTF may also assign a move request when a device is released (by another device) or it is heading to the parking point, it also waits until the device is ready to receive its next assignment and then it assigns (or commits) only one move request. Furthermore, all the dispatching rules assign move requests to devices without explicitly considering the status of the other devices. Assigning the move requests only one at a time and doing it only when a device is ready for an assignment offers some inherent flexibility in that a move request is not prematurely committed to a device (and vice versa). It also tends to "automatically" balance the workload among the devices since a device cannot be assigned another move request until it serves the current one. However, one can also argue that such an approach is very "myopic" and may lead to unnecessary empty travel especially since the assignments are made without considering the status of the other devices.

As an alternate, we propose the B$^2$D$^2$ rule where all the devices place a "bid" for each move request. In fact, with B$^2$D$^2$, we do not wait until a device is ready for its next assignment. Instead, as soon as a move request is received, all the devices in the system (whether they are traveling loaded, empty, or idle) are allowed to place a bid. Once all the bids are received, we attempt to assign or commit the move request to the device with the lowest bid, subject to the distance threshold we described earlier under MOD STTF.

More specifically, under B$^2$D$^2$, we maintain two sets of lists for each device $d$: one set, say, $C_d$, represents all the move requests that have been committed to device $d$, while the other set, say, $UC_d$, represents all the move requests that have been assigned but not committed to device $d$. To keep the rule simple and manageable, we assume that: 1. all the move requests in $C_d$ are served in the order in which they were placed in $C_d$ (i.e., we do not attempt to further reduce empty travel by resequencing the entries in $C_d$ each time a new request is added to it), and 2. the set $UC_d$ may contain at most one move request.

When move request $i$ arrives, the bid placed by device $d$ is equal to the total (remaining) distance it must travel to serve all the move requests currently in $C_d$, plus the empty travel distance from the destination station of the last move request in $C_d$ to output queue $i$. If $C_d$ is empty but $UC_d$ is not, then device $d$'s bid is equal to the empty travel distance to output queue $i$ from the next intersection on device $d$'s route. Lastly, if both $C_d$ and $UC_d$ are empty, then device $d$'s bid is equal to the empty travel distance from station $j$ (where device $d$ became idle) to output queue $i$.

Suppose device $d$ is the lowest bidder. If the above empty travel distance to output queue $i$ is less than the threshold distance, then move request $i$ is committed to device $d$, i.e., it is added to $C_d$; otherwise, device $d$ is assigned but not committed to move request $i$ provided that $UC_d$ is empty. If $UC_d$ is not empty, then we attempt to "swap" move request $i$ with the one already in $UC_d$ if it reduces empty travel; if it does not, then device $d$ is declared ineligible for move request $i$ and we consider the next smallest bid.

After all the bids have been considered, if move request $i$ has not been committed to or assigned to any device, then it is placed in a list, say, $UA$, until the next move request arrives. At that point, all the move requests (i.e., the ones in $UA$, plus the one that just arrived) are offered, one at a time, for bidding and the above process is repeated for each move request. Hence, bidding occurs only when a move request arrives, and all the devices are allowed to bid on the new request as well as all the move requests in $UA$.

When device $d$ has served all the move requests in $C_d$, it begins serving the move request in $UC_d$ on an uncommitted basis. If $UC_d$ is empty, then device $d$ becomes idle at its last delivery station. Lastly, whenever we add a move request to $C_d$, we remove the move request in $UC_d$ (if any) and allow all the devices, including device $d$, to bid on it again. Although this may slightly increase the number of bids we process per unit time, it ensures that a move request does not unnecessarily wait in $UC_d$ if it can be committed to another device.

Before we formally present the $B^2D^2$ rule, we need the following notation. Let:

$TH$ = distance threshold
MR $i$ = move request $i$
$D$ = set of devices
$UA$ = set of unassigned move requests
$BID_{id}$ = bid placed by device $d$ for MR $i$
$ET_{id}$ = empty distance device $d$ must travel to reach MR $i$

$C_d$ = set of MRs committed to device $d$

$UC_d$ = set of MRs assigned to but not committed to device $d$

(may not contain more than one MR)

$DS_d$ = destination station of the last MR in $C_d$

Using the above notation, and the bidding process described earlier, the flow chart for $B^2D^2$ is shown in Figure 2.

# 4 Performance Evaluation

In this section we will evaluate the performance of the MOD STTF rule and the $B^2D^2$ rule relative to STTF and MOD FCFS via simulation. As we remarked earlier, since the STTF rule is one of the most competitive rules in the literature, we will use the STTF results as a "benchmark." All the "percent improvements" we report are relative to STTF.

In order to perform as complete an evaluation as possible, we report device statistics as well as work-in-process (WIP) results for each rule we tested. We also used four different layouts (L-1 through L-4) to test the rules; L-4 was taken from [1]. All the data for L-1, L-2, and L-3 are shown Appendix I, except for the distance matrix for L-3, which is too large to

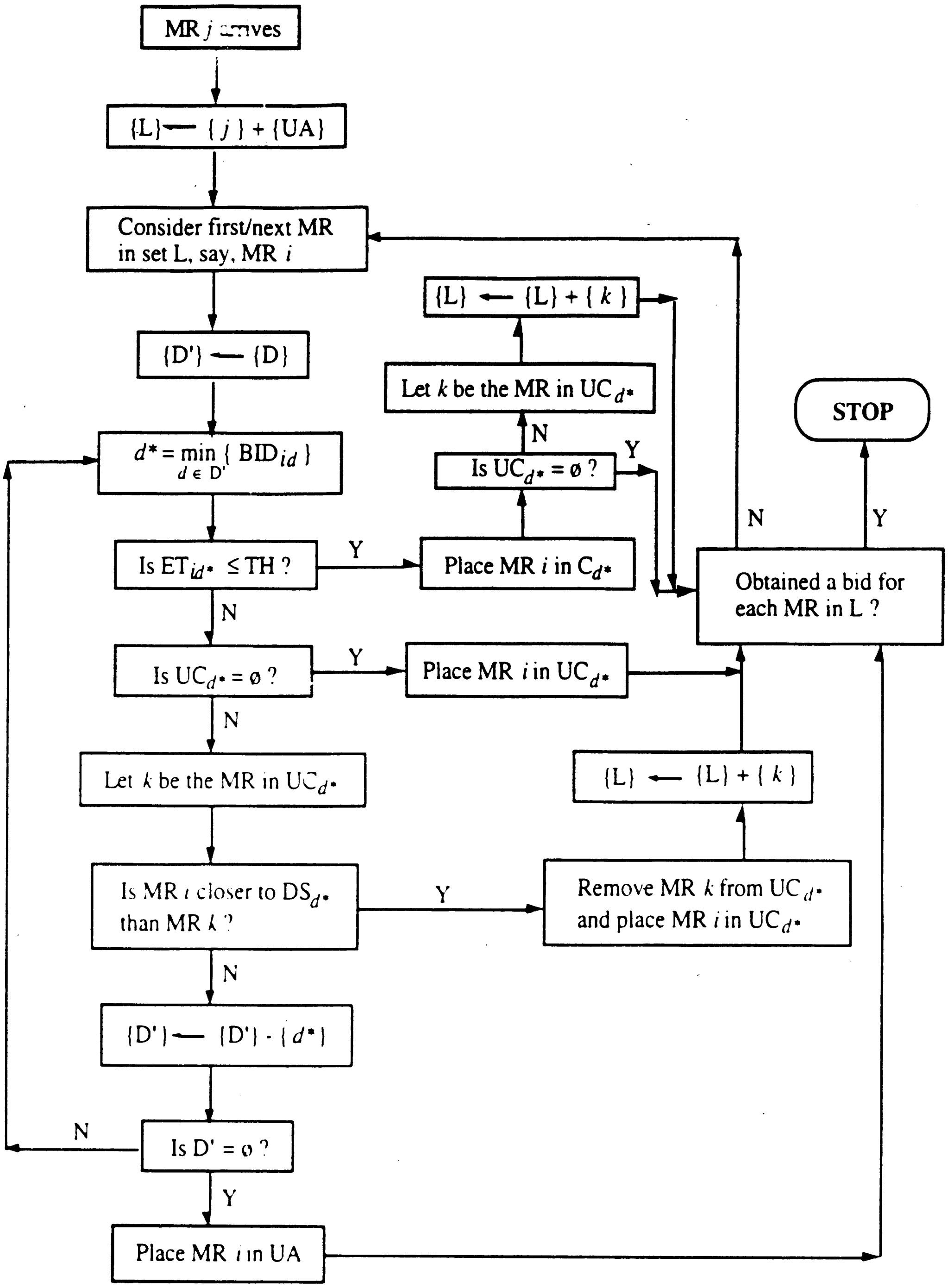


Figure 2: Flow Chart for $B^2D^2$

present here.

For each device we present the: 1. Fraction of time the device is performing committed empty travel, 2. Fraction of time the device is performing uncommitted empty travel, 3. Fraction of time the device is traveling empty (committed or otherwise), and 4. Average device utilization (which includes all empty or loaded device travel, plus load pickup and deposit times). Of course, with STTF and MOD FCFS, all empty travel must be interpreted as committed empty travel since there is no device reassignment.

For WIP results, we present the: 1. average time a load spends in the system (including the time it spends in the input queues and the time required at each processor station to process the load), 2. average time a load spends in an output queue waiting for a device, 3. maximum output queue size observed in the system, and 4. average time a load spends in an input queue waiting for a processor. Although the second and third statistics are the ones that are most likely affected by the performance of the handling system, we report all four of the above statistics since it presents a more complete WIP picture. We also included the maximum output queue size in our evaluation to identify any station that may possibly be "orphaned" by any of the rules we tested.

The results we report were obtained from a simulation model written in GPSS/H. Each layout/rule combination was simulated for 10.000 loaded trips per device per replication. Ten replications were performed for each layout except L–3, for which we used 5 replications due to its large size. All the confidence intervals we report are 95% intervals. In addition to those we presented in section 1, we assume that jobs from outside the system arrive according to a Uniform distribution and that all the processing times are uniformly distributed. To ensure that there are no bottleneck processors, processing capacities are adjusted to achieve an average processor utilization in the range of 70% to 80%. For simplicity, we assume that the handling times are deterministic and that the device speed is equal to one distance unit per unit time; load deposit and pick up times are assumed to be negligible.

The results of the simulation experiment are presented in Table 1, where we show the percent improvement achieved in reducing empty travel along with the aforementioned device and WIP statistics. For both MOD STTF and $B^2D^2$, we report results obtained with various distance threshold values. The first distance threshold value shown for each layout reflects the average loaded trip length in distance units (du's). For example, for L–1, the average loaded trip is approximately equal to 12 du's. The remaining distance values are based on the (sorted) list of all the distance values obtained from the layout. For L–1, 19 (29) du's corresponds approximately to the

15th (30th) percentile. For L–2, 45 du's corresponds approximately to the 30th percentile. For L–3, 17 (25) du's corresponds approximately to the 20th (40th) percentile. Lastly, for L–4, 28 (41) du's corresponds approximately to the 20th (35th) percentile. In addition, with MOD STTF, we report results obtained with "minisum parking" versus letting the idle device stay at its last delivery station.

The results are fairly consistent across the four layouts we tested. A smaller threshold value generally produces better results. (Further reductions in the threshold values, however, did not lead to noticeable improvements.) We also observe that "minisum parking" under the MOD STTF rule leads to little or no improvement in system performance (which was the primary reason we did not test "minisum parking" under the $B^2D^2$ rule).

Perhaps the most significant conclusion we draw from Table 1, however, is that both MOD STTF and $B^2D^2$ outperform STTF by a substantial margin in terms of reducing both empty device travel and the average output queue waiting time. Furthermore, the maximum queue lengths we observe under all the rules seem very reasonable and do not indicate any "orphaned" stations.

Another conclusion we draw from Table 1 is that $B^2D^2$ performs better than MOD STTF. Although at first $B^2D^2$ may appear to be more complex, we believe it is a rule that can be easily implemented because the "bidding" concept has intuitive appeal; it is easy to program and easy to explain to a plant manager or production supervisor.

Table 1 also shows that MOD FCFS is not a competitive rule. It is outperformed by the STTF rule. (We must stress, however, that the MOD FCFS rule lends itself more readily to analytical modeling; we refer the reader to [2] for details.) Lastly, in all the layout/rule combinations we tested, the average input queue waiting time does not appear to be sensitive to the dispatching rule used. As a result, savings in average time in system (i.e., total expected WIP) are not as dramatic. However, in Table 1, if we take the minimum average output queue waiting time obtained under $B^2D^2$ and compare it with the average output queue waiting time obtained under STTF, we find that the percent reduction in average waiting time ranges from 25% to 18% for the four layouts. This would result in substantial savings due to less WIP in the output queues; it may even allow the elimination of one or more devices. Note that, the expected device utilization under STTF ranges from 80% to 93%, while it ranges from 62% to 79% under $B^2D^2$, which indicates excess handling capacity.

To further test the excess handling capacity offered by $B^2D^2$, we maintained the same number of devices in each layout but we gradually increased the throughput requirement (by reducing the interarrival time of the jobs

# Table 1: Simulation Results

(a) Layout 1 (16 stations/3 devices)

| Distance Threshold | Comm. Empty | Uncomm. Empty | **Total Empty** | Percent Imprvmt. | Device Utilzn. | Average Time in System | Avg. Out Que. W.T. | Max Out Queue | Avg. Input Que. W.T. |
|---|---|---|---|---|---|---|---|---|---|
| **MOD STTF 12** | 0.1525 ± 0.0002 | 0.3653 ± 0.0001 | **0.5178 ± 0.0002** | 9.73% | 85.16% | 82.76 ± 0.52 | **2.90 ± 0.01** | 2 | 2.80 ± 0.05 |
| **MOD STTF 12*** | 0.1414 ± 0.0002 | 0.3598 ± 0.0001 | **0.5012 ± 0.0001** | 12.62% | 83.50% | 82.11 ± 0.49 | **2.75 ± 0.01** | 2 | 2.78 ± 0.07 |
| **MOD STTF 19** | 0.3244 ± 0.0002 | 0.1965 ± 0.0001 | **0.5209 ± 0.0001** | 9.19% | 85.47% | 83.10 ± 0.63 | **2.97 ± 0.01** | 2 | 2.80 ± 0.06 |
| **MOD STTF 19*** | 0.3172 ± 0.0002 | 0.1917 ± 0.0001 | **0.5089 ± 0.0001** | 11.28% | 84.77% | 82.45 ± 0.53 | **2.81 ± 0.01** | 2 | 2.75 ± 0.09 |
| **MOD STTF 29** | 0.4519 ± 0.0002 | 0.0883 ± 0.0001 | **0.5402 ± 0.0003** | 5.82% | 87.40% | 83.57 ± 0.46 | **3.00 ± 0.01** | 2 | 2.78 ± 0.06 |
| **MOD STTF 29*** | 0.4418 ± 0.0001 | 0.0736 ± 0.0001 | **0.5154 ± 0.0002** | 10.15% | 84.92% | 83.18 ± 0.45 | **2.85 ± 0.01** | 2 | 2.79 ± 0.08 |
| $B^2D^2$ 12 | 0.1449 ± 0.0001 | 0.3088 ± 0.0001 | **0.4537 ± 0.0001** | 20.90% | 78.77% | 79.94 ± 0.38 | **2.57 ± 0.01** | 2 | 2.84 ± 0.09 |
| $B^2D^2$ 19 | 0.3221 ± 0.0002 | 0.1401 ± 0.0002 | **0.4623 ± 0.0002** | 19.40% | 79.53% | 80.35 ± 0.32 | **2.71 ± 0.01** | 2 | 2.79 ± 0.10 |
| $B^2D^2$ 29 | 0.4449 ± 0.0002 | 0.0386 ± 0.0002 | **0.4835 ± 0.0002** | 15.71% | 81.66% | 81.36 ± 0.47 | **2.94 ± 0.01** | 2 | 2.75 ± 0.02 |
| STTF | 0.5736 ± 0.0002 | - | **0.5736 ± 0.0002** | - | 90.74% | 84.73 ± 0.32 | **3.42 ± 0.01** | 2 | 2.80 ± 0.06 |
| MOD FCFS | 0.6188 ± 0.0001 | - | **0.6188 ± 0.0001** | - | 95.26% | 87.81 ± 0.27 | **4.28 ± 0.01** | 2 | 2.79 ± 0.11 |

* park at minisum station 8

(b) Layout 2 (18 stations/5 devices)

| Distance Threshold | Comm Empty | Uncomm Empty | **Total Empty** | Percent Imprvmt | Device Utilzn. | Average Time in System | Avg. Out Que W T | Max Out Queue | Avg Input Que W T |
|---|---|---|---|---|---|---|---|---|---|
| **MOD STTF 29** | 0.2094 ± 0.0002 | 0.2087 ± 0.0001 | **0.4181 ± 0.0002** | 22.16% | 77.89% | 165.03 ± 0.62 | **4.72 ± 0.01** | 3 | 6.25 ± 0.07 |
| **MOD STTF 29*** | 0.2177 ± 0.0002 | 0.1909 ± 0.0001 | **0.4087 ± 0.0002** | 23.91% | 76.95% | 163.87 ± 0.47 | **4.20 ± 0.01** | 3 | 6.26 ± 0.05 |
| **MOD STTF 45** | 0.3390 ± 0.0002 | 0.1076 ± 0.0001 | **0.4466 ± 0.0002** | 16.85% | 80.74% | 165.29 ± 0.37 | **4.80 ± 0.01** | 3 | 6.26 ± 0.11 |
| **MOD STTF 45*** | 0.3415 ± 0.0003 | 0.0960 ± 0.0001 | **0.4375 ± 0.0003** | 18.54% | 79.83% | 164.26 ± 0.28 | **4.31 ± 0.01** | 3 | 6.20 ± 0.06 |
| $B^2D^2$ 29 | 0.1945 ± 0.0003 | 0.1299 ± 0.0002 | **0.3244 ± 0.0004** | 39.60% | 68.47% | 163.52 ± 0.59 | **4.07 ± 0.01** | 2 | 6.22 ± 0.03 |
| $B^2D^2$ 45 | 0.2923 ± 0.0001 | 0.0515 ± 0.0001 | **0.3437 ± 0.0002** | 36.01% | 70.40% | 163.57 ± 0.62 | **4.14 ± 0.01** | 2 | 6.20 ± 0.06 |
| STTF | 0.5371 ± 0.0001 | - | **0.5371 ± 0.0001** | - | 89.80% | 172.39 ± 0.61 | **5.76 ± 0.01** | 3 | 6.25 ± 0.08 |
| MOD FCFS | 0.5974 ± 0.0001 | - | **0.5974 ± 0.0001** | - | 95.82% | 185.12 ± 0.39 | **7.42 ± 0.01** | 4 | 6.28 ± 0.06 |

* park at minisum station 15

# Table 1: (continued) Simulation Results

(c) Layout 3 (50 stations/15 devices)

| Distance Threshold | Comm. Empty | Uncomm. Empty | **Total Empty** | Percent Imprvmt. | Device Utilzn. | Average Time in System | Avg. Out Que. W.T. | Max Out Queue | Avg. Input Que. W.T. |
|---|---|---|---|---|---|---|---|---|---|
| MOD STTF 11 | 0.2330 ± 0.0002 | 0.2428 ± 0.0001 | **0.4758 ± 0.0002** | 18.72% | 77.65% | 128.78 ± 0.62 | **1.60 ± 0.01** | 4 | 2.64 ± 0.07 |
| MOD STTF 11* | 0.2359 ± 0.0002 | 0.2558 ± 0.0002 | **0.4917 ± 0.0002** | 16.01% | 79.24% | 129.63 ± 0.58 | **1.63 ± 0.01** | 4 | 2.65 ± 0.06 |
| MOD STTF 17 | 0.3359 ± 0.0002 | 0.1541 ± 0.0002 | **0.4900 ± 0.0003** | 16.30% | 79.07% | 130.36 ± 0.79 | **1.66 ± 0.01** | 4 | 2.65 ± 0.06 |
| MOD STTF 17* | 0.3470 ± 0.0002 | 0.1519 ± 0.0001 | **0.4989 ± 0.0002** | 14.78% | 79.96% | 131.63 ± 0.86 | **1.70 ± 0.01** | 4 | 2.66 ± 0.05 |
| MOD STTF 25 | 0.4448 ± 0.0002 | 0.0781 ± 0.0002 | **0.5229 ± 0.0003** | 10.68% | 82.36% | 136.79 ± 0.67 | **1.88 ± 0.01** | 4 | 2.68 ± 0.06 |
| MOD STTF 25* | 0.4658 ± 0.0002 | 0.0718 ± 0.0002 | **0.5376 ± 0.0003** | 8.17% | 83.83% | 137.45 ± 0.81 | **1.95 ± 0.01** | 4 | 2.65 ± 0.06 |
| $B^2D^2$ 11 | 0.2192 ± 0.0001 | 0.1251 ± 0.0001 | **0.3443 ± 0.0002** | 41.19% | 64.50% | 129.58 ± 0.71 | **1.60 ± 0.01** | 4 | 2.69 ± 0.08 |
| $B^2D^2$ 17 | 0.2799 ± 0.0002 | 0.0703 ± 0.0001 | **0.3502 ± 0.0002** | 40.18% | 65.09% | 129.96 ± 0.36 | **1.62 ± 0.01** | 4 | 2.70 ± 0.05 |
| $B^2D^2$ 25 | 0.3268 ± 0.0001 | 0.0289 ± 0.0001 | **0.3557 ± 0.0001** | 39.24% | 65.64% | 130.32 ± 0.68 | **1.65 ± 0.01** | 4 | 2.71 ± 0.11 |
| STTF | 0.5854 ± 0.0002 | - | **0.5854 ± 0.0002** | - | 88.61% | 148.28 ± 0.46 | **2.21 ± 0.01** | 4 | 2.66 ± 0.09 |
| MOD FCFS | 0.6568 ± 0.0001 | - | **0.6568 ± 0.0001** | - | 95.75% | 159.24 ± 0.37 | **4.39 ± 0.01** | 4 | 2.69 ± 0.08 |

* park at minisum station 23

(d) Layout 4 (20 stations/8 devices)

| Distance Threshold | Comm Empty | Uncomm Empty | **Total Empty** | Percent Imprvmt | Device Utilzn. | Average Time in System | Avg. Out Que W.T | Max Out Queue | Avg. Input Que W.T |
|---|---|---|---|---|---|---|---|---|---|
| MOD STTF 20 | 0.2181 ± 0.0002 | 0.2147 ± 0.0002 | **0.4329 ± 0.0002** | 22.39% | 80.90% | 84.04 ± 0.06 | **2.68 ± 0.01** | 3 | 3.48 ± 0.03 |
| MOD STTF 20* | 0.2154 ± 0.0003 | 0.2227 ± 0.0003 | **0.4381 ± 0.0004** | 21.46% | 81.42% | 84.15 ± 0.09 | **2.69 ± 0.01** | 3 | 3.48 ± 0.05 |
| MOD STTF 28 | 0.3106 ± 0.0002 | 0.1401 ± 0.0001 | **0.4507 ± 0.0002** | 19.20% | 82.69% | 84.15 ± 0.08 | **2.70 ± 0.01** | 3 | 3.47 ± 0.02 |
| MOD STTF 28* | 0.3122 ± 0.0002 | 0.1454 ± 0.0001 | **0.4576 ± 0.0002** | 17.96% | 83.38% | 84.28 ± 0.09 | **2.71 ± 0.01** | 3 | 3.47 ± 0.05 |
| MOD STTF 41 | 0.4163 ± 0.0002 | 0.0646 ± 0.0001 | **0.4809 ± 0.0002** | 13.79% | 85.71% | 86.03 ± 0.09 | **3.03 ± 0.01** | 3 | 3.50 ± 0.03 |
| MOD STTF 41* | 0.4237 ± 0.0001 | 0.0715 ± 0.0001 | **0.4952 ± 0.0002** | 11.22% | 87.14% | 86.81 ± 0.10 | **3.10 ± 0.01** | 3 | 3.50 ± 0.06 |
| $B^2D^2$ 20 | 0.1681 ± 0.0002 | 0.0736 ± 0.0001 | **0.2417 ± 0.0001** | 56.67% | 61.73% | 81.82 ± 0.18 | **2.14 ± 0.01** | 2 | 3.46 ± 0.09 |
| $B^2D^2$ 28 | 0.2118 ± 0.0001 | 0.0336 ± 0.0001 | **0.2454 ± 0.0002** | 56.01% | 62.15% | 82.36 ± 0.24 | **2.21 ± 0.01** | 2 | 3.42 ± 0.06 |
| $B^2D^2$ 41 | 0.2437 ± 0.0002 | 0.0084 ± 0.0001 | **0.2521 ± 0.0002** | 54.80% | 62.82% | 82.60 ± 0.23 | **2.24 ± 0.01** | 2 | 3.44 ± 0.05 |
| STTF | 0.5578 ± 0.0001 | - | **0.5578 ± 0.0001** | - | 93.41% | 91.98 ± 0.07 | **4.10 ± 0.01** | 3 | 3.40 ± 0.09 |
| MOD FCFS | 0.6092 ± 0.0001 | - | **0.6092 ± 0.0001** | - | 98.55% | 101.76 ± 0.12 | **5.91 ± 0.01** | 5 | 3.42 ± 0.08 |

* park at minisum station 12

arriving from outside the system). The results are shown in Table 2, where the first column for each rule/layout combination corresponds to the original throughput level while the second and third columns represent an increase in the throughput level. As long as they both meet throughput, MOD STTF is superior to STTF. However, it is clear that $B^2D^2$ has the strongest performance.

We replaced deterministic handling times with uniformly distributed handling times and repeated the above experiment in its entirety (including the inflated throughput levels). The results we observed were virtually identical to the ones we reported above for deterministic handling times.

## 5 Conclusions and Future Work

We present two new dispatching rules, namely, the modified shortest-travel-time-first (MOD STTF) rule and the bidding-based- dynamic-dispatching ($B^2D^2$) rule, to improve the efficiency of trip-based material handling systems. The MOD STTF rule uses a distance-based threshold to determine if an empty device should be committed to a move request or not. Under MOD STTF, a device which is assigned but not committed to a move request is allowed to travel towards that move request but it may be reassigned to another move request at any station or intersection that it passes through. The

## Table 2: Simulation Results with Inflated Throughput Rates

### (a) Device Utilization

| Dispatching Rules | L - 1 (16s/3devices) | | | L - 2 (18s/5devices) | | | L - 3 (50s/15devices) | | | L - 4 (20s/8devices) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MOD FCFS | 95.26% | 99.40% | unstable | 95.82% | 99.74% | unstable | 95.75% | 99.99% | unsatble | 98.55% | 99.93% | unstable |
| STTF | 90.74% | 96.27% | unstable | 89.80% | 96.53% | unstable | 88.61% | 97.88% | unstable | 93.41% | 97.88% | unstable |
| MOD STTF | 83.50% | 91.64% | unstable | 76.95% | 88.67% | unstable | 77.65% | 90.86% | unstable | 80.90% | 90.85% | unstable |
| BBDD | 78.77% | 88.37% | 98.89% | 68.47% | 82.39% | 98.62% | 64.50% | 77.14% | 98.14% | 61.73% | 74.66% | 98.93% |

### (b) Percentage of Time Devices Travel Empty

| Dispatching Rules | L - 1 (16s/3devices) | | | L - 2 (18s/5devices) | | | L - 3 (50s/15devices) | | | L - 4 (20s/8devices) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MOD FCFS | 61.88% | 60.18% | unstable | 59.74% | 57.20% | unstable | 65.68% | 67.22% | unstable | 60.92% | 55.64% | unstable |
| STTF | 57.36% | 57.06% | unstable | 53.71% | 53.99% | unstable | 58.54% | 62.48% | unstable | 55.78% | 53.59% | unstable |
| MOD STTF | 50.12% | 52.37% | unstable | 40.87% | 45.96% | unstable | 47.58% | 55.54% | unstable | 43.29% | 46.19% | unstable |
| BBDD | 45.37% | 49.15% | 32.33% | 32.44% | 39.85% | 41.57% | 34.43% | 40.42% | 46.38% | 24.17% | 30.37% | 23.75% |

### (c) Time in System

| Dispatching Rules | L - 1 (16s/3devices) | | | L - 2 (18s/5devices) | | | L - 3 (50s/15devices) | | | L - 4 (20s/8devices) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MOD FCFS | 87.81 | 85.62 | unstable | 185.12 | 173.78 | unstable | 159.24 | 160.57 | unstable | 101.76 | 99.25 | unstable |
| STTF | 84.73 | 75.51 | unstable | 172.39 | 152.56 | unstable | 148.28 | 140.23 | unstable | 91.98 | 77.09 | unstable |
| MOD STTF | 82.11 | 72.20 | unstable | 163.87 | 144.71 | unstable | 128.78 | 115.91 | unstable | 84.04 | 73.87 | unstable |
| BBDD | 79.94 | 70.88 | 93.86 | 163.52 | 142.70 | 145.67 | 129.58 | 115.85 | 120.47 | 81.82 | 66.14 | 71.55 |

### (d) Average Output Queue Waiting Time

| Dispatching Rules | L - 1 (16s/3devices) | | | L - 2 (18s/5devices) | | | L - 3 (50s/15devices) | | | L - 4 (20s/8devices) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MOD FCFS | 4.28 | 5.80 | unstable | 7.42 | 10.04 | unstable | 4.39 | 5.51 | unstable | 5.91 | 7.04 | unstable |
| STTF | 3.42 | 3.86 | unstable | 5.76 | [illegible] | unstable | 2.21 | 2.43 | unstable | 4.10 | 4.42 | unstable |
| MOD STTF | 2.75 | 3.12 | unstable | [illegible] | [illegible] | unstable | 1.60 | 1.79 | unstable | 2.68 | 2.85 | unstable |
| BBDD | 2.57 | 2.98 | 9.57 | [illegible] | 4.81 | 10.56 | 1.60 | 1.78 | 3.21 | 2.14 | 2.46 | 5.25 |

### (e) Max Output Queue Size

| Dispatching Rules | L - 1 (16s/3devices) | | | L - 2 (18s/5devices) | | | L - 3 (50s/15devices) | | | L - 4 (20s/8devices) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MOD FCFS | 2 | 3 | unstable | 4 | 5 | unstable | 4 | 6 | unstable | 5 | 6 | unstable |
| STTF | 2 | 3 | unstable | 3 | 4 | unstable | 4 | 6 | unstable | 3 | 4 | unstable |
| MOD STTF | 2 | 3 | unstable | 3 | 3 | unstable | 4 | 5 | unstable | 3 | 4 | unstable |
| BBDD | 2 | 3 | 7 | 2 | 2 | 7 | 4 | 5 | 9 | 2 | 3 | 7 |

$B^2D^2$ rule, on the other hand, is a novel application of the "bidding" concept to trip-based handling systems. Under $B^2D^2$, each device places a bid based on its current workload, and the system tries to assign or commit each move request to the lowest bidding device provided that a distance-based threshold is met. Each device is allowed to bid on each (uncommitted) move request.

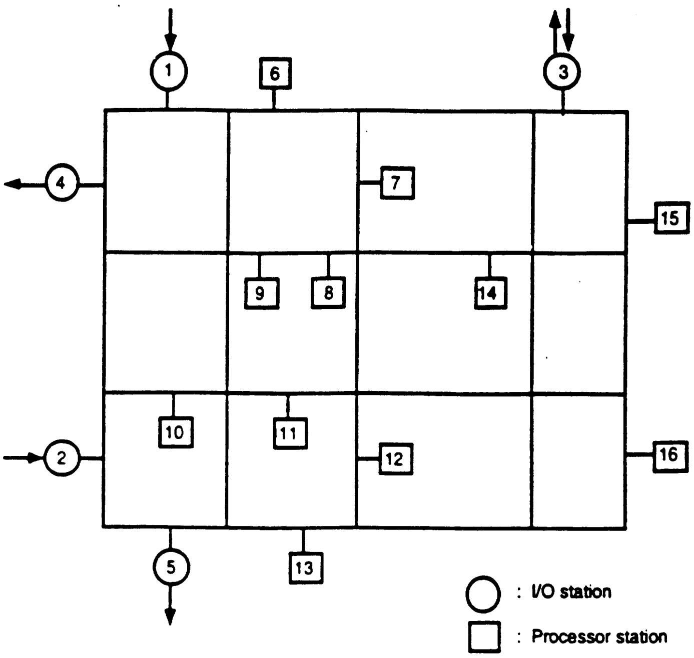
Simulation results indicate that both of the above rules perform considerably better than the shortest-travel-time-first (STTF) rule, which is a very competitive rule and widely used in industry. Although both rules are more elaborate than STTF and similar rules, they are not unreasonably complex. In fact, we believe the bidding concept has intuitive appeal.

Our study can be extended in several directions. For example, one can include the impact of device downtimes and reallocate the workload of a down device to the others in the most effective manner possible. Also, we assumed throughout the study that all the move requests have equal priority. One may develop new rules, or modify the ones presented here, to capture the overall priority assigned to each job processed and moved through the system. See, for example, Egbelu [10], who presents a "demand driven" rule where move requests are not all treated equally. Another possibility is to incorporate the processing times into the dispatching rule by "anticipating" the move requests whenever possible.

# APPENDIX I

## Input Data for Layout 1

**(i) Layout**



**(ii) Distance Matrix**

| Station No | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 30 | 30 | 10 | 40 | 8 | 20 | 22 | 17 | 29 | 30 | 40 | 41 | 34 | 43 | 60 |
| 2 | 30 | 0 | 60 | 20 | 10 | 38 | 40 | 32 | 27 | 11 | 20 | 30 | 21 | 44 | 57 | 50 |
| 3 | 30 | 60 | 0 | 40 | 60 | 22 | 20 | 28 | 33 | 49 | 40 | 40 | 49 | 16 | 13 | 30 |
| 4 | 10 | 20 | 40 | 0 | 30 | 18 | 30 | 22 | 17 | 21 | 30 | 40 | 41 | 34 | 47 | 60 |
| 5 | 40 | 10 | 60 | 30 | 0 | 38 | 40 | 32 | 27 | 19 | 20 | 20 | 11 | 44 | 57 | 40 |
| 6 | 8 | 38 | 22 | 18 | 38 | 0 | 12 | 20 | 15 | 27 | 28 | 32 | 39 | 26 | 35 | 52 |
| 7 | 20 | 40 | 20 | 30 | 40 | 12 | 0 | 8 | 13 | 29 | 20 | 20 | 29 | 14 | 27 | 40 |
| 8 | 22 | 32 | 28 | 22 | 32 | 20 | 8 | 0 | 5 | 21 | 18 | 18 | 27 | 12 | 25 | 38 |
| 9 | 17 | 27 | 33 | 17 | 27 | 15 | 13 | 5 | 0 | 16 | 17 | 23 | 28 | 17 | 30 | 43 |
| 10 | 29 | 11 | 49 | 21 | 19 | 27 | 29 | 21 | 16 | 0 | 9 | 19 | 20 | 33 | 46 | 39 |
| 11 | 30 | 20 | 40 | 30 | 20 | 28 | 20 | 18 | 17 | 9 | 0 | 10 | 19 | 24 | 37 | 30 |
| 12 | 40 | 30 | 40 | 40 | 20 | 32 | 20 | 18 | 23 | 19 | 10 | 0 | 9 | 24 | 37 | 30 |
| 13 | 41 | 21 | 49 | 41 | 11 | 39 | 29 | 27 | 28 | 20 | 19 | 9 | 0 | 33 | 46 | 29 |
| 14 | 34 | 44 | 16 | 34 | 44 | 26 | 14 | 12 | 17 | 33 | 24 | 24 | 33 | 0 | 13 | 26 |
| 15 | 43 | 57 | 13 | 47 | 57 | 35 | 27 | 25 | 30 | 46 | 37 | 37 | 46 | 13 | 0 | 17 |
| 16 | 60 | 50 | 30 | 60 | 40 | 52 | 40 | 38 | 43 | 39 | 30 | 30 | 29 | 26 | 17 | 0 |

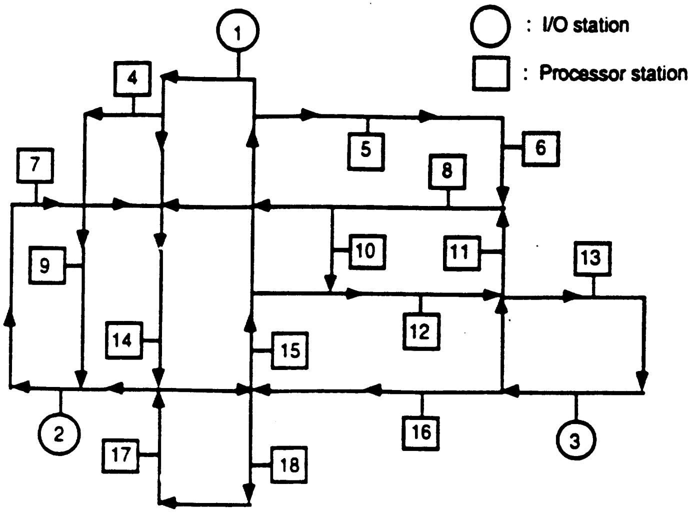
**(iii) Routing Matrix and Interarrival Times (mins)**

| Job Type/ Route | Sequence of Stations Visited | | | | | | Intarr Time |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 6 | 7 | 8 | 9 | 4 | 150.00 |
| 2 | 2 | 10 | 11 | 12 | 13 | 5 | 150.00 |
| 3 | 3 | 15 | 16 | 12 | 14 | 3 | 300.00 |

# APPENDIX I

## Input Data for Layout 2

**(i) Layout**



**(ii) Distance Matrix**

| Station No | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 43 | 96 | 15 | 75 | 87 | 67 | 82 | 30 | 95 | 74 | 64 | 77 | 32 | 46 | 109 | 66 | 51 |
| 2 | 80 | 0 | 109 | 95 | 88 | 100 | 24 | 95 | 31 | 108 | 87 | 77 | 90 | 45 | 59 | 122 | 79 | 64 |
| 3 | 51 | 63 | 0 | 66 | 59 | 71 | 87 | 26 | 81 | 39 | 18 | 48 | 21 | 62 | 30 | 13 | 50 | 35 |
| 4 | 64 | 28 | 93 | 0 | 72 | 84 | 52 | 79 | 15 | 92 | 71 | 61 | 74 | 29 | 43 | 106 | 63 | 48 |
| 5 | 46 | 68 | 77 | 61 | 0 | 12 | 92 | 21 | 76 | 34 | 55 | 45 | 58 | 57 | 71 | 90 | 91 | 76 |
| 6 | 34 | 56 | 65 | 49 | 42 | 0 | 80 | 9 | 64 | 22 | 43 | 33 | 46 | 45 | 59 | 78 | 79 | 64 |
| 7 | 56 | 20 | 85 | 71 | 64 | 76 | 0 | 71 | 7 | 84 | 63 | 53 | 66 | 21 | 35 | 98 | 55 | 40 |
| 8 | 25 | 47 | 56 | 40 | 33 | 45 | 71 | 0 | 55 | 13 | 34 | 24 | 37 | 36 | 50 | 69 | 70 | 55 |
| 9 | 93 | 13 | 122 | 108 | 101 | 113 | 37 | 108 | 0 | 121 | 100 | 90 | 103 | 58 | 72 | 135 | 92 | 77 |
| 10 | 54 | 76 | 43 | 69 | 62 | 74 | 100 | 29 | 84 | 0 | 21 | 11 | 24 | 65 | 73 | 56 | 93 | 78 |
| 11 | 33 | 55 | 64 | 48 | 41 | 53 | 79 | 8 | 63 | 21 | 0 | 32 | 45 | 44 | 58 | 77 | 78 | 63 |
| 12 | 43 | 65 | 32 | 58 | 51 | 63 | 89 | 18 | 73 | 31 | 10 | 0 | 13 | 54 | 62 | 45 | 82 | 67 |
| 13 | 70 | 82 | 19 | 85 | 78 | 90 | 106 | 45 | 100 | 58 | 37 | 67 | 0 | 81 | 49 | 32 | 69 | 54 |
| 14 | 35 | 11 | 64 | 50 | 43 | 55 | 35 | 50 | 42 | 63 | 42 | 32 | 45 | 0 | 14 | 77 | 34 | 19 |
| 15 | 21 | 43 | 50 | 36 | 29 | 41 | 67 | 36 | 51 | 49 | 28 | 18 | 31 | 32 | 0 | 63 | 66 | 51 |
| 16 | 38 | 50 | 67 | 53 | 46 | 58 | 74 | 53 | 68 | 66 | 45 | 35 | 48 | 49 | 17 | 0 | 37 | 22 |
| 17 | 37 | 13 | 66 | 52 | 45 | 57 | 37 | 52 | 44 | 65 | 44 | 34 | 47 | 48 | 16 | 79 | 0 | 21 |
| 18 | 52 | 28 | 81 | 67 | 60 | 72 | 52 | 67 | 59 | 80 | 59 | 49 | 62 | 63 | 31 | 94 | 15 | 0 |

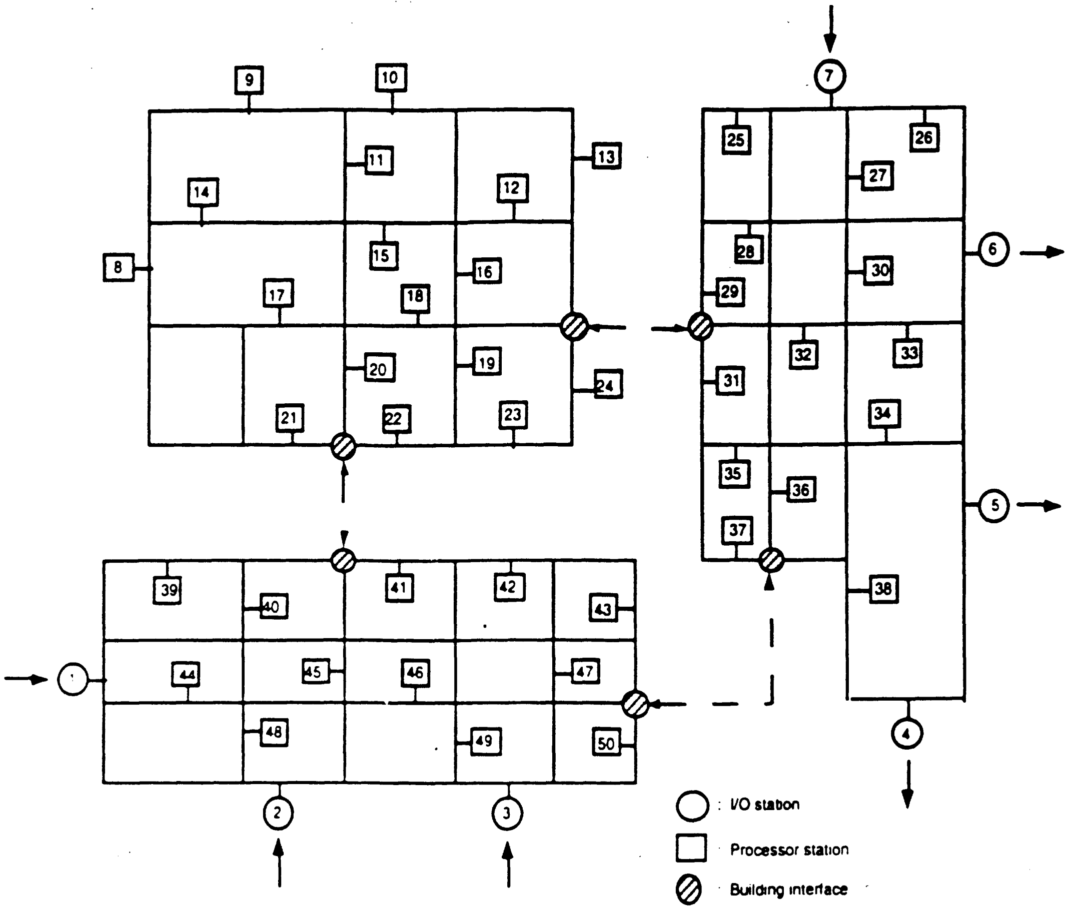
**(iii) Routing Matrix and Interarrival Times (mins)**

| Job Type/ Route | Sequence of Stations Visited | | | | | | | Intarr Time |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 8 | 6 | 10 | 11 | 3 | | 610.50 |
| 2 | 1 | 4 | 9 | 7 | 14 | 17 | 2 | 402.60 |
| 3 | 2 | 14 | 15 | 10 | 12 | 13 | 3 | 402.60 |
| 4 | 3 | 16 | 18 | 15 | 17 | 2 | | 297 00 |
| 5 | 3 | 11 | 8 | 5 | 6 | 13 | 3 | 610 50 |

# APPENDIX I

## Input Data for Layout 3

### (i) Layout



### (ii) Routing Matrix and Interarrival Times (mins)

| Job Type/Route | Sequence of Stations Visited | | | | | | | | | | | | | | | Interarr. Time |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 39 | 40 | 46 | 41 | 21 | 16 | [illegible] | [illegible] | [illegible] | 30 | 6 | | | | 126.13 |
| 2 | 2 | 45 | 40 | 20 | 17 | 8 | 9 | 10 | 13 | 11 | 15 | 34 | 5 | | | 252.26 |
| 3 | 3 | 49 | 47 | 43 | 42 | 41 | 22 | 19 | 16 | 12 | 29 | 28 | 27 | 26 | 6 | 252.26 |
| 4 | 1 | 44 | 45 | 21 | 17 | 8 | 9 | 11 | 15 | 19 | 23 | 24 | 32 | 33 | 5 | 252.26 |
| 5 | 2 | 48 | 46 | 47 | 42 | 20 | 8 | 14 | 11 | 10 | 13 | 31 | 37 | 38 | 4 | 252.26 |
| 6 | 3 | 49 | 47 | 43 | 42 | 20 | 19 | 23 | 24 | 29 | 25 | 27 | 6 | | | 252.26 |
| 7 | 1 | 44 | 45 | 40 | 21 | 17 | 18 | 16 | 15 | 10 | 13 | 31 | 34 | 5 | | 252.26 |
| 8 | 3 | 50 | 36 | 32 | 33 | 34 | 38 | 4 | | | | | | | | 126.13 |
| 9 | 7 | 25 | 28 | 30 | 32 | 35 | 37 | 38 | 4 | | | | | | | 252.26 |

# References

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

[2] P. J. Egbelu, and J. M. A. Tanchoco, "Characterization of Automatic Guided Vehicle Dispatching Rules," *International Journal of Production Research,* Vol. 22, No. 3, 359–374 (1984).

[3] D. S. Russell, and J. M. A. Tanchoco, "An Evaluation of Vehicle Dispatching Rules and Their Effect on Shop Performance," *Material Flow,* Vol. 1, 271–280 (1984).

[4] T. J. Hodgson, R. E. King, S. K. Monteith, and S. R. Schultz, "Developing Control Rules for an AGV using Markov Decision Processes," *Material Flow,* Vol. 4, 85–96 (1987).

[5] R. E. King, T. J. Hodgson, and S. K. Monteith, "Extracting Heuristic Control Rules for AGVs' using Markov Decision Processes," *Belgian Journal of OR, Stat. and Comp. Sci.* , Vol. 27, No. 2, 3–17 (1987).

[6] J. J. Bartholdi, III, and L. K. Platzman, "Decentralized Control of a Fixed Route Automatic Guided Vehicle System," *IIE Transactions,*

Vol. 21, No. 1, 76–81 (1989).

[7] M. H. Han, and L. F. McGinnis, "Control of Material Handling Transporter in Automated Manufacturing," *IIE Transactions.* Vol. 21, No. 2, 184–190 (1989).

[8] I. Sabuncuoglu, and D. L. Hommertzheim, "Dynamic Dispatching Algorithm for Scheduling Machines and Automated Guided Vehicles in a Flexible Manufacturing System," *International Journal of Production Research,* Vol. 30, No. 5, 1059–1079 (1992).

[9] R. L. Francis, L. F. McGinnis, and J. A. White, " Facility Layout and Location: An Analytical Approach," *Prentice Hall* (1992).

[10] P. J. Egbelu, "Pull Versus Push Strategy for Automated Guided Vehicle Load Management in a Batch Manufacturing System," *Journal of Manufacturing Systems*, Vol. 6, No. 3, 209–221 (1987).