TRIP-BASED MATERIAL HANDLING SYSTEMS: THROUGHPUT CAPACITY ANALYSIS

Mandyam M. Srinivasan
Yavuz A. Bozer
Myeon-Sig Cho

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

Technical Report 89-32

November 1989

Revised October 1990

Trip-Based Material Handling Systems: Throughput Capacity Analysis

Mandyam M. Srinivasan

Yavuz A. Bozer

Myeon-Sig Cho

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

Abstract

In this paper we present a general-purpose analytical model to compute the approximate throughput capacity of a trip-based material handling system used in a manufacturing setting. A wide variety of handling systems, including freight elevators, cranes, microload automated storage/retrieval (AS/R) systems, industrial lift trucks, and automated guided vehicle (AGV) systems can be modeled as trip-based handling systems. To our knowledge, this model is the first analytical model to explicitly consider an empty device dispatching rule. The model is first developed for a single-device system (such as a crane) and subsequently, with a simple modification, it is extended to multiple-device systems (such as lift trucks and AGVs). Using this model one can rapidly evaluate a wide range of handling and layout alternatives for given flow data. Hence, the model would be most effective when used early in the design phase to narrow down the set of alternative handling systems and configurations prior to simulation.

1. Introduction

In most manufacturing systems, the material handling system plays a critical role since it is primarily responsible for providing the right material at the right place, and at the right time. (See Tompkins and White [14] for a full definition of material handling.) A poorly designed material handling system interferes with the efficient operation of a manufacturing concern and in the long–term it may lead to a substantial loss in productivity.

The significance and role of the material handling system are understood better today and, both in research and practice, more time is being devoted to the design and analysis of handling systems. However, primarily due to a lack of general–purpose analytical design models, many handling systems used today are designed through simulation models.

Although simulation is a powerful analysis tool, without the right direction provided by pre–simulation analytical models such as the one presented here, simulation becomes an expensive and time consuming undertaking. This is true especially early in the design phase where the analyst is usually faced with a wide array of fairly diverse handling alternatives that appear to "do the same job." Furthermore, the successful use of simulation requires the right combination of hardware, software, and technical expertise.

In this paper we present an analytical model that one can use to rapidly determine the throughput capacity of a wide range of handling and layout alternatives (for given flow data) without necessarily investing in simulation hardware and software. We do not, however, view this model as a substitute for simulation since it is an approximate model and simulation is still required to "fine tune" the system and/or further analyze specific dynamic interactions in preferred systems identified by the analytical model.

The remainder of the paper is organized as follows: after a brief literature review, in section 3 we will describe the problem setting along with relevant definitions (including that of "trip–based" handling systems) and assumptions. In section 4, we will develop the model for a single–device system. This model represents the bulk of the analytical work since its extension to a multiple-device system is performed through a straightforward modification presented in section 5. In section 6, we present several numerical examples and use simulation to evaluate the performance of the analytical model. Lastly, in section 7, we summarize the results of the study and identify some areas for future research.

2. Literature Review

An increasing number of papers concerned with material handling systems continues to appear in the literature. Given the wide range of interests represented by these papers, we will limit the following review to those studies related to throughput capacity estimation in a manufacturing environment. We will assume that the reader is familiar with basic material handling terminology and equipment types. (For an excellent review of material handling equipment, along with sample illustrations, the reader may refer to Tompkins and White, Chapter 6 [14].)

To the best of our knowledge, in the manufacturing arena, there is no general–purpose analytical model which can be used to determine the throughput capacity of a wide–range of material handling systems. Those that are reported in the literature are developed for specific types of material handling equipment, primarily microload automated storage/retrieval (AS/R) systems and (pick–and–drop) automated guided vehicle (AGV) systems.

Consider first AGV systems. A deterministic analytical model to obtain a lower bound on the number of AGVs required is presented by Maxwell and Muckstadt [11] who study a system with unit load carriers operating on a unidirectional guidepath. After computing the "net flow" for each workstation, the authors model the empty vehicle flow as a transportation problem where the supply (demand) nodes correspond to workstations where empty vehicles are generated (needed). Leung et al. [10] extend the above model to the case where different types of vehicles are used within the same system.

For a similar AGV system, Egbelu [6] presents four relatively simple analytical models to estimate the number of vehicles required. We will focus on only three of them here since the fourth model (due to Koff [9]) requires an user–supplied estimate on the fraction of time a vehicle is blocked or idle. Two of the models, say, model 1 and model 2, are due to Beisteiner [1] while the third one, say, model 3, is due to Egbelu [6]. The three models differ primarily in the approach used for approximating empty vehicle travel. Model 1 simply assumes that the fraction of time a vehicle travels empty is equal to the fraction of time it travels loaded. (For given flow data and travel times, the latter is straightforward to compute.)

Model 2 is similar to the one proposed by Maxwell and Muckstadt [11] in that the net flow is computed for each workstation and the empty vehicles are assumed to flow from "supply" nodes to "demand" nodes as described earlier. However, rather than solving a transportation problem, in model 2 an *empty* vehicle traveling from any supply node to any demand node is assumed to travel a distance equal to the average distance traveled by a *loaded* vehicle. (The reader may refer to [1] or [6] for details.) In model 3, the expected number of empty vehicle trips per unit time from station i to station j is assumed to be given by E(# of deliveries to station i) E(# of pick–ups from station j) / E(total # of pick–ups in the system). Once the number of empty trips between all pairs of stations is estimated, one can compute the fraction of time the vehicle would travel empty, provided there is no congestion in the system and the empty travel time from i to j is known.

None of the above four models explicitly considers an empty vehicle dispatching rule. That is, empty vehicle travel is computed independent of the empty vehicle dispatching rule used in the system. Therefore, using a fixed layout, Egbelu presents simulation results for each model to characterize the adequacy of these models for various dispatching rules (which are presented in an earlier study by Egbelu and Tanchoco [7]). The author observes that the estimated number of vehicles obtained from model 1 agrees with the value obtained via simulation under the shortest–travel–time (or distance)–first (STTF) rule, where an empty vehicle is dispatched to the closest (in time or distance) unassigned move request. Since model 1 assumes that the fraction of time a vehicle travels empty is equal to the fraction of time that it travels loaded, one has to conclude that the above result observed for model 1 is mostly coincidental; that is, the layout and the pattern of move requests have, in all likelihood, created a situation where the travel time required for the empty vehicle to travel to the nearest move request was equal, on the average, to the travel time required for the loaded vehicle to deliver that move request to its destination.

It is observed in [6] that the number of AGVs required under the STTF rule is at least 30% less than the number of AGVs required under other dispatching rules tested, including the modified first–come–first–served (MFCFS) rule.[1] However, as discussed in [7], the performance of the STTF rule is quite sensitive to the location of the stations. If a particular pick–up point is not close to any of the deposit points, it is likely that it will not receive an empty vehicle for an extended period of time. This may also explain the fact that, for a different layout with six vehicles, Egbelu and Tanchoco [7] empirically observed the opposite: that is, given infinite or finite buffer sizes at each station, the MFCFS rule outperformed the STTF rule. (The STTF rule also leads to "shop locking"; please refer to [6] or [7].)

The estimated number of AGVs obtained from model 3, on the other hand, is close to that obtained from simulation provided that the STTF or the MOQS (maximum–outgoing–queue–size first) dispatching rules are not used. Lastly, model 2 underestimates the vehicle requirements for all dispatching rules tested.

In general, Egbelu [6] concludes that the performance of the analytical models depend on the vehicle dispatching rule adopted for the system. While this observation certainly seems valid, it is

[1] Under the MFCFS rule, a station places its move requests only one at–a–time. That is, even if a station has two or more loads waiting to be moved, it can have at most one "active" move request. If necessary, the next move request at a station is made "active" when the oldest request at that station is served.. All "active" move requests are served on a FCFS basis.

difficult to generalize it since the simulation experiment is based on a single layout with no replications. Hence, for arbitrary layouts, no consistent relationship has been established between model performance and dispatching rules. This is perhaps one of the drawbacks of "quick–and–dirty" analytical models that do not explicitly take into account a specific dispatching rule.

Consider next, microload AS/R systems. Chow [2] presents an approximate analytical model to predict the utilization of the S/R machine (and the mean waiting time for move requests) by modeling the system as an M/G/1/FCFS queue. That is, the arrival of the move requests (from the workstations) are assumed to follow the Poisson distribution while the S/R machine has a general service time distribution. Assuming that the S/R machine never finds the destination buffer full, approximate values for the first and second moments of the service time are obtained from the flow matrix by a simple probabilistic argument. Once the approximate service time distribution is defined, however, the S/R machine is assumed to serve each move request according to this distribution regardless of the actual origin and destination of a move request (and the current position of the S/R machine).

Although the performance of the approximate model is not fully explored in the paper, the FCFS dispatching rule leads to unnecessary empty travel for the S/R machine. In a subsequent paper, Chow [3] uses a simulation model to evaluate the impact of alternative dispatching rules for the S/R machine.

Toro-Ramos and McGinnis ([15], [16]), study the performance of a microload AS/R system with finite input and output buffers. In [15] the authors assume that the S/R machine gets blocked if the input buffer of the destination station is full. In [16] they treat the case where each station has an alternate storage area. If the input buffer of the receiving station is full, then the S/R machine temporarily stores the container in the alternate storage area. Empty S/R machine travel is estimated as in model 3 described earlier. (Recall that the empty device dispatching rule is not explicitly considered in model 3.) The authors model the system as a network of queues with a central server station.

Tanchoco et al. [13] and Wysk et al. [17] use CAN–Q (see Solberg [12]) to estimate the number of AGVs required. Tanchoco et al. test the performance of the CAN–Q based model by comparing the results to those obtained from a simulation model. In the analytical model empty vehicle travel time is not explicitly considered while in the simulation model an empty vehicle is dispatched to the closest non–empty move request queue. For systems with only four stations, the authors observe that the CAN–Q based model generally underestimates the number of AGVs required. Consequently, they suggest that this model be used only as an approximate tool to obtain a lower bound on the number of AGVs required prior to an extensive simulation study.

On the other hand, Wysk et al. [17] consider empty vehicle travel time in their model. However, the performance of the CAN–Q based model for estimating vehicle requirements is not evaluated via simulation. It is important to stress that, in CAN–Q based models, no empty vehicle dispatching rule is explicitly taken into account and the travel time between each pair of stations is assumed to be the same.

Yao and Buzacott [18], and [19] model a flexible manufacturing system as a network of queues with a central server station which represents the material handling system. (Loads traveling from one processor station to another go through the central server station.) In both papers, it is difficult to capture the performance of the material handling system with sufficient accuracy since delivery times between all stations are assumed to be the same regardless of where a load is picked up and where it needs to be delivered next. Furthermore, it is assumed that a load is (next) delivered to station j with probability p_j which is independent of the station where it was last processed.

In short, although there are several studies concerned with the design of microload AS/R systems and AGV systems, none of the models reported in the literature explicitly capture the empty vehicle dispatching rule. Hence, the results are difficult to generalize and no consistency in model performance has been established. Furthermore, some of them (such as those based on CAN–Q or a central server) oversimplify the material handling system in terms of delivery times and routing probabilities.

In the next section we will define trip–based material handling systems and the problem setting for the proposed analytical model. We must stress that our objective is to develop an analytic model based on an efficient empty vehicle dispatching rule. As shown in section 6, the dispatching rule in question is not only easy to implement but it is comparable in throughput performance to other dispatching rules discussed earlier in this section.

3. Problem Setting: Definitions and Assumptions

3.1. Trip–Based Material Handling Systems

A trip–based material handling system consists of one or several self–powered "devices" that are capable of operating *independently* in an *asynchronous* fashion. Each move request in the system is served by one of the devices. In order to serve a move request, a device has to perform a trip. Each trip is composed of empty travel followed by loaded travel. The former accounts for the time it takes for the empty device to travel from its current location to the station which placed the move request while the latter represents the time it takes the loaded device to deliver the load to the next station. The loaded travel time also includes the load handling time which consists of a pick–

up and deposit operation. On each trip the device is assumed to handle only one load (which represents one move request).

A device becomes empty as soon as it deposits the load at its destination station. At this point, if there are other move requests in the system, the empty device is assigned to one of them according to the empty device dispatching rule described in section 3.3. Otherwise, the device remains idle at its last delivery point until it is assigned to another move request.

Note that, at certain times, there may be one or more idle devices in the system (which implies that there are no move requests awaiting service). When a move request eventually arrives and finds two or more devices idle, it will select (or "call") one of the idle devices according to the idle device selection rule which is also described in section 3.3.

Note that many handling systems used in the fabrication or assembly of discrete parts can be modeled as a trip–based handling system. Examples of single–device systems include freight elevators, bridge or gantry cranes, and microload AS/R systems. (In the case of microload AS/R systems, the model will not work well if the S/R machine finds the input buffer full a non-negligible portion of the time.) Examples of multiple-device systems include a fleet of industrial lift trucks, a fleet of unit load (pick–and–drop) AGVs, and manual systems (where people move material by, say, pushing carts or hand trucks).

Programmable "smart" monorails (inverted or otherwise), on the other hand, satisfy most of the properties of trip–based handling systems. That is, each carrier is self–propelled and any carrier can be programmed to pick up any load in the system. However, in a trip–based material handling system a device should be able to "pass or overtake" other devices when necessary. In a monorail, a carrier cannot pass another carrier that is traveling on the same segment of the track.

One may argue that the above assumption significantly limits the applicability of the proposed model. This is not the case. In an AGV system, it is possible for a vehicle to temporarily leave the guidepath to pass another vehicle. Furthermore, progress made in self–guided vehicles (that operate with no guidepath) supports the "independent operation" assumption. Operator driven lift trucks (or people pushing carts) have always had the capability to pass each other as long as sufficient aisle space is provided. Whenever two–way traffic is anticipated in an aisle, it must be wide enough to allow two devices (traveling in the same or opposite direction) to pass each other. (See, for example, Tompkins and White [14], p. 93.)

3.2. , The Manufacturing System

The manufacturing system is assumed to consist of a set of stations (or nodes). There is a pick–up/deposit (P/D) point associated with each station. For each station, the deposit point is represented by an input buffer, and the pick–up point by an output buffer. (Refer to Figure A1 in

Appendix III.) The device delivers loads at the input buffers and picks them up from the output buffers. Each pick–up point immediately follows the corresponding deposit point. If the distance from the deposit point to the pick–up point of a station is non–negligible, then the model can still be used after a simple change. (Refer to the last paragraph in Appendix III.) The device is assumed to take the shortest path from its origin to destination. However, as seen later, the analytical model will work with any user–defined empty and loaded travel time matrix.

Note that a move request (which represents one load) is defined by two variables: the point of origin (which corresponds to the output buffer where the load is waiting) and the point of destination (which corresponds to the input buffer of the next station). Recall that the device moves only one move request (or load) at a time.

There are two types of stations: input/output (I/O) stations and processor stations. Loads from outside the system enter through one of the I/O stations and – when all the operations have been completed – they exit through one of the I/O stations. Incoming loads arrive at the output buffer of an I/O station while outgoing loads are deposited at the input buffer of an I/O station where they instantly leave the system. That is, no processing takes place at an I/O station.

A processor station, on the other hand, represents either one machine, or a group of machines (a cell), or a department. Loads to be processed are removed from the corresponding input buffer; later, when processing is complete, they are placed in the corresponding output buffer. (Material handling needs within a station is beyond the scope of our study.) Although load characteristics may change after processing, for ease of exposition we assume that, as far as the material handling system is concerned, flow is conserved at each processor station. It is straightforward to extend the model to handle situations where flow is not conserved at each processor station.

Given that each load must be processed through certain stations in the above system, the *required throughput* is usually defined as the number of loads that the devices must move per unit time. The *throughput capacity* of the system, however, is defined only for a given number of devices and it is expressed as the maximum number of loads that the devices are capable of moving per unit time.

3.3. Device Dispatching

The empty device dispatching rule we developed for this study is a simple modification of the FCFS rule. Upon delivering a load at station i, an empty device first inspects the output buffer of station i. If one or more move requests are found, the device is assigned to one of them. Otherwise, the device serves the oldest move request in the system (regardless of its location). We will refer to this rule as the modified FCFS (or MOD FCFS) rule which is different than another modification suggested by Egbelu and Tanchoco [7]. (See MFCFS described earlier in section 2.)

Note that the MOD FCFS rule attempts to reduce unnecessary empty travel by allowing the device to override the FCFS rule whenever it finds another move request at the destination point. At the same time, it attempts to evenly distribute device time among the stations since an empty device is assigned to the oldest move request in the system whenever an empty trip is inevitable.

Recall that an empty device will become idle if no move requests are present. Hence, when a load eventually arrives at the output buffer of a station, it may find one or more devices idle. In the simulation model, if two or more idle devices are found, we assume that the new move request is assigned to the device which has been idle for the longest time period. This rule, namely, the longest idle vehicle (LIV) rule is one of the rules proposed by Egbelu and Tanchoco [7] who also considered the NV (nearest vehicle) rule and the RV (random vehicle) rule.

Although the NV rule may seem to be more appealing than the LIV rule, as shown empirically in [6] and [7], the idle device selection rule generally has little or no impact on the throughput capacity of the system since it is usually invoked very seldom. In other words, when the minimum or near–minimum required number of devices are used, the probability of finding two or more idle devices in the system diminishes quite rapidly. On the other hand, if more than the minimum required number of devices are used, then the idle device selection rule will be invoked more often; however, since excess capacity is present, using the "most efficient" selection rule for idle devices is not essential.

Note that the MOD FCFS rule is a centralized rule in the sense that a device must be told where the oldest move request in the system is located. In trip–based handling systems, one way to achieve this is to use radio dispatched devices. Generally speaking, radio dispatched devices improve productivity, response times, and inventory accuracy. The benefits of radio dispatching and examples of successful applications have been reported in [4], [20], [21], and [22].

4. The Single–Device Model

We consider a system with M stations. Let ϑ denote the set of processor stations, and let Ω denote the set of I/O stations in the system. (Recall that every station in the system is assumed to have both an input and an output buffer.) The rate at which loads arrive at the output buffer of station i is denoted by λ_i, and these rates are assumed to be given. To perform a trip, the device picks up a load from the output buffer of a station and delivers it to the input buffer of some other station. Let Λ_i denote the rate at which the device delivers loads to the input buffer of station i. Recall that we assume Λ_i equals λ_i in steady state at the processor stations. (This implies that a processor station may never be a bottleneck in the system.) For I/O stations, Λ_i need not equal λ_i in general. However, from conservation of flow, provided that the device is able to meet the demand placed on it , we must have $\sum_{i\in\Omega} \Lambda_i = \sum_{i\in\Omega} \lambda_i$. When a load is delivered at the input buffer of an I/O station, it is assumed to immediately exit from the system.

In the following discussion, unless specified otherwise, the index for any summation is assumed to be over the range 1 through M. Let p_{ij} denote the probability that a load, which is picked up by the device from the output buffer of station i, is destined for station j. The p_{ij} terms can be easily computed from the given flow data for the stations. (It is implicit that $p_{ii} = 0$.) The values for Λ_i are obtained as the unique solution to the system of equations:

$$\Lambda_i = \sum_j \lambda_j p_{ji}, \quad i \in \Omega, \qquad \text{and} \qquad \Lambda_i = \lambda_i, \quad i \in \vartheta. \tag{4.1}$$

Let λ_T denote the total arrival rate at the output buffers of all stations. Note that

$$\lambda_T = \sum_i \lambda_i = \sum_i \Lambda_i. \tag{4.2}$$

Whenever an empty device checks the *output buffer* of a station, we say that the device "inspects" station i. The time taken by the device to pick up a load from the output buffer of station i, transport it from station i to station j, unload it at station j, and then inspect station j, is collectively assumed to be a random variable with a known mean τ_{ij}. The empty device travel time between stations i and j, on the other hand, is a random variable with a known mean σ_{ij}, which includes the time taken to "inspect" station j. (Strictly speaking, when an empty device arrives at a station empty it *knows* that there is a load waiting at the station; however, we still use the term inspect for ease of exposition.) We assume that each station has ample input and output buffers, so that no blocking of any form can occur.

Let α_f and α_e denote the proportion of time that the device is traveling with a load and traveling empty, respectively. Let $\rho = \alpha_f + \alpha_e$ denote the *utilization* of the device. If the device is to meet the required throughput, we must have $\rho < 1$, which in turn implies that both α_f and α_e, as well as their sum, must be less than 1. The term α_f is easily computed as

$$\alpha_f = \sum_i \lambda_i \sum_j p_{ij}\tau_{ij}. \tag{4.3}$$

Note that, if the value of α_f computed from equation (4.3) is greater than or equal to 1, the device will not meet throughput, and α_f can no longer be defined as a proportion. Also, if $\rho < 1$, we let $\alpha_I = 1 - \rho$ denote the proportion of time that the device is idle.

Our objective is to determine the expected device utilization for a given set of input parameters expressed by the λ_i values (which drive the throughput requirement) and the travel times. In the process, we derive a necessary and sufficient condition for the device to meet the required throughput. If desired, one can also evaluate the throughput capacity of the device by systematically increasing the λ_i values and/or the travel times until the above condition is violated. Note that, if the λ_i values are kept constant, one can also rapidly evaluate the material handling impact of alternative layouts by changing the corresponding travel times.

The movement of the device among the stations is described by the transition matrix R. In this matrix we explicitly need to distinguish between instances where the device arrived at station i with a load (the state i_f), and instances where it is dispatched (empty) to pick up a load at station i (the state i_e). In addition, state I denotes the idle state; although the idle state does not identify the station at which the device became idle, we shall see, subsequently, that no information is lost by this representation. Let $\varpi = \{1_e, 1_f, \ldots, M_e, M_f, I\}$ denote the set of possible states for the device. Each element r_{jk}, $j,k \in \varpi$, of the transition matrix R is obtained as follows.

Consider first, the movement of the device from state i_e to the state next visited by it. Clearly, the device arrived at station i to pick up a load. Hence, the only possible states the device can visit next are the states j_f ($j \neq i$), with probability p_{ij}.

Consider next, the case where the device has just delivered a load at the input buffer of station i. In other words, we are considering the movement of the device from state i_f to the next state visited by it. Following the MOD FCFS rule, on delivering a load at station i, the device immediately inspects the output buffer of station i. Let q_{i_f} denote the probability that the device finds the output buffer of station i empty upon inspection, given that it just delivered a load at station i. We assume that the instant at which a load is delivered at any station *is a random point in time.* With this assumption, the term q_{i_f} represents the probability that the output buffer of station i is empty at an arbitrary instant in time. This assumption also implies that the product $\prod_i q_{i_f}$ gives the probability that all the output buffers are empty at an arbitrary instant in time. Since the device becomes idle if all the output buffers are empty, we obtain the following expression for ρ:

$$\rho \quad = \quad 1 - \prod_i q_{i_f}. \tag{4.4}$$

Thus the device moves from state i_f to state I with probability $1-\rho$.

Let $\bar{q}_{i_f} = 1 - q_{i_f}$ denote the probability that the output buffer of station i is non-empty when the device inspects it, given that it just delivered a load at station i. In this case, the load that is picked up from the output buffer is destined for station j with probability p_{ij}, and therefore the device moves from state i_f to state j_f with probability $\bar{q}_{i_f} p_{ij}$.

Suppose that the device, upon delivering a load at station i, finds its output buffer empty, but that there is at least one other load elsewhere in the system. The probability of this occurrence is just $q_{i_f}(1-\prod_{j\neq i} q_{j_f}) = \rho-\bar{q}_{i_f}$. In this case we assume that the oldest waiting load is at station j, $j \neq i$, with probability $\lambda_j/(\lambda_T - \lambda_i)$; in other words, this probability is assumed to be proportional to the rate at which jobs arrive at the output buffer of station j (accounting, of course, for the fact that station i cannot be the station that is visited next). Hence, the device moves from state i_f to state j_e, $(j \neq i)$ with probability $(\rho-\bar{q}_{i_f})\lambda_j/(\lambda_T - \lambda_i)$.

Finally, if the device is idle, the probability that the next load to be moved originates at station i is approximated by λ_i/λ_T. In other words, the probability that the device moves from state I to state i_f is assumed to be proportional to the rate at which loads arrive at the output buffer of station i. Hence, the transition matrix R is given by

$$R = \begin{array}{c|cccccccc|}
 & 1_e & 1_f & 2_e & 2_f & \cdots\cdots & M_e & M_f & I \\
\hline
1_e & 0 & 0 & 0 & p_{12} & & 0 & p_{1M} & 0 \\
1_f & 0 & 0 & (\rho-\bar{q}_{1_f})\frac{\lambda_2}{\lambda_T-\lambda_1} & \bar{q}_{1_f}p_{12} & & (\rho-\bar{q}_{1_f})\frac{\lambda_M}{\lambda_T-\lambda_1} & \bar{q}_{1_f}p_{1M} & 1-\rho \\
2_e & 0 & p_{21} & 0 & 0 & & 0 & p_{2M} & 0 \\
2_f & (\rho-\bar{q}_{2_f})\frac{\lambda_1}{\lambda_T-\lambda_2} & \bar{q}_{2_f}p_{21} & 0 & 0 & & (\rho-\bar{q}_{2_f})\frac{\lambda_M}{\lambda_T-\lambda_2} & \bar{q}_{2_f}p_{2M} & 1-\rho \\
 & : & & & & & & & \\
: & & & & & & & & \\
M_e & 0 & p_{M1} & 0 & p_{M2} & & 0 & 0 & 0 \\
M_f & (\rho-\bar{q}_{M_f})\frac{\lambda_1}{\lambda_T-\lambda_M} & \bar{q}_{M_f}p_{M1} & (\rho-\bar{q}_{M_f})\frac{\lambda_2}{\lambda_T-\lambda_M} & \bar{q}_{M_f}p_{M2} & & 0 & 0 & 1-\rho \\
I & \frac{\lambda_1}{\lambda_T} & 0 & \frac{\lambda_2}{\lambda_T} & 0 & & \frac{\lambda_M}{\lambda_T} & 0 & 0 \\
\hline
\end{array} \qquad (4.5)$$

For ease of exposition, we will assume, without loss of generality, that station 1 is one of the I/O stations. We define a *cycle* as the time between two successive inspections at station 1, and let C_1 denote its expected value. We will refer to C_1 as the *cycle time*.

Let v_i, $i = 1,\ldots,M$, denote the expected number of times the device inspects the output buffer of station i during a cycle. (Note that, by this definition, $v_1 = 1$.) Among these visits, the device arrives at station i either empty (v_{i_e}) or loaded (v_{i_f}), i.e., $v_i = v_{i_e} + v_{i_f}$. Let v_I denote the expected number of times the device becomes idle during a cycle. With $\bar{v} = [(v_{i_e},v_{i_f}), v_I]$, these terms are obtained from the unique solution to the system of equations:

$$\bar{v}R = \bar{v}; \quad \text{with} \quad v_{1_e} + v_{1_f} = v_1 = 1. \qquad (4.6)$$

From equations (4.5) and (4.6), we get

$$v_{i_e} = \sum_{j\neq i} v_{j_f}(\rho - \bar{q}_{j_f})\frac{\lambda_i}{\lambda_T - \lambda_j} + v_I\frac{\lambda_i}{\lambda_T}, \tag{4.7.a}$$

$$v_{i_f} = \sum_{j\neq i} (v_{j_e} + v_{j_f}\bar{q}_{j_f})\, p_{ji}, \tag{4.7.b}$$

$$v_I = (1-\rho)\sum_{j} v_{j_f}. \tag{4.7.c}$$

Note that the expected number of loads delivered at station i during a cycle is exactly equal to the expected number of times the device arrives loaded at station i in the same period; that is

$$v_{i_f} = \Lambda_i C_1. \tag{4.8}$$

Substituting the value for v_{i_f} given by equation (4.8) in equation (4.7.a) and (4.7.c), and using equation (4.2), we obtain:

$$v_I = (1-\rho)\,\lambda_T\, C_1, \tag{4.9.a}$$

$$v_{i_e} = \sum_{j\neq i} \Lambda_j C_1 (\rho - \bar{q}_{j_f})\frac{\lambda_i}{\lambda_T - \lambda_j} + (1-\rho)\,\lambda_i C_1. \tag{4.9.b}$$

During a cycle, the device makes v_{i_e} empty visits and v_{i_f} loaded visits to station i, on the average. Every time the device visits a station empty, a load is picked up with probability 1. (Recall that under the MOD FCFS rule, the empty device is not dispatched to a station unless that station contains the oldest move request.) In contrast, when the device arrives loaded at station i, a load is picked up from the output buffer of this station with probability $\bar{q}_{i_f}$. Hence, during a cycle, the expected number of loads picked up by the device equals $v_{i_e} + v_{i_f}\bar{q}_{i_f}$. For the system to be stable, this must equal $\lambda_i C_1$, which is the expected number of jobs that arrive at the output buffer of station i during a cycle. Thus, for a stable system we must have

$$\lambda_i C_1 = v_{i_e} + v_{i_f}\bar{q}_{i_f}. \tag{4.10}$$

From equations (4.8) and (4.10), we obtain

__Lemma 4.1__

$$q_{i_f} = \frac{v_{i_f} - (\lambda_i C_1 - v_{i_e})}{v_{i_f}} = \frac{v_i - \lambda_i C_1}{\Lambda_i C_1}. \tag{4.11}$$

■

We now develop an expression for C_1 by considering the expected transition time for the device to move from one state to the next. Let T_{i_f} (T_{i_e}) denote the expected time from the instant at which the device arrives loaded (empty) at station i until the instant it visits the next state. Let T_I denote the expected amount of time that the device remains in the idle state each time it becomes idle. Then C_1 is obtained as:

$$C_1 \quad = \quad \sum_{k\in\varpi} v_k T_k. \qquad (4.12)$$

The terms T_k, $k \in \varpi$, are determined as follows. Consider first, the instant when an *empty* device arrives at station i. The possible next states are j_f, $j\neq i$. Hence,

$$T_{i_e} \quad = \quad \sum_{j\neq i} p_{ij}\tau_{ij}\,. \qquad (4.13.a)$$

Consider next, the instant when a *loaded* device arrives at station i. The state visited at the next transition instant is either j_f, j_e ($j \neq i$) or I, with probability $\bar{q}_{i_f}p_{ij}$, $(\rho-\bar{q}_{i_f})\lambda_j/(\lambda_T-\lambda_i)$, and $1-\rho$, respectively. If the next state visited is j_f (j_e), then the expected transition time to this state is simply τ_{ij} (σ_{ij}). Although it takes no time for the device to enter the idle state I (from the state i_f), once the device enters the idle state then the next state it enters is state j_e, $j=1,\ldots,M$, with probability λ_j/λ_T; in this case the time to move to station j from station i is just σ_{ij}. Therefore,

$$T_{i_f} \quad = \quad \sum_{j\neq i} \bar{q}_{i_f}\, p_{ij}\tau_{ij} + \sum_{j\neq i} (\rho-\bar{q}_{i_f})\frac{\lambda_j}{\lambda_T-\lambda_i}\sigma_{ij} + \sum_{j} (1-\rho)\frac{\lambda_j}{\lambda_T}\sigma_{ij}\,. \qquad (4.13.b)$$

Lastly, the expected time in the idle state until a load arrives at any output buffer is approximated as

$$T_I \quad = \quad \frac{1}{\lambda_T}. \qquad (4.13.c)$$

Let

$$\theta_i \quad = \quad \sum_j \lambda_j\sigma_{ij}, \qquad (4.14.a)$$

$$\chi \quad = \quad \sum_i \lambda_i\,\theta_i\,, \qquad \text{and} \qquad \hat{\chi} \quad = \quad \sum_i \Lambda_i\,\theta_i\,, \qquad (4.14.b)$$

and set

$$\varphi_i \quad = \quad \frac{\hat{\chi}}{\lambda_T} - \frac{\Lambda_i\,\chi}{\lambda_i\,\lambda_T}. \qquad (4.15)$$

From equations (4.8) through (4.15) we obtain the following expression for q_{i_f}, stated as:

Proposition 4.2:

$$q_{i_f} \quad = \quad \frac{\lambda_i}{\Lambda_i\,\chi}(\lambda_T-\lambda_i)\,(\rho-\alpha_f-\varphi_i) + \frac{\lambda_i}{\lambda_T}(1-\rho). \qquad (4.16)$$

■

A proof of Proposition 4.2 is provided in Appendix I.

We now determine the proportion of time the device travels empty. Note that the device travels empty only if it arrives loaded at a station, and finds no job waiting there. In the expression for T_{i_f}

given by equation (4.13.b), the second and the third summations, taken together, yields the expected time the device travels empty to pick up another load, for every loaded visit to station i. In a cycle of expected length C_1, the device makes $v_{i_f} = \Lambda_i C_1$ loaded visits to station i on the average, and so the expected empty travel generated by station i per cycle is given by E_i, where

$$E_i \quad = \quad \Lambda_i C_1 \Big(\sum_{j \neq i} (\rho - \bar{q}_{i_f}) \frac{\lambda_j}{\lambda_T - \lambda_i} \sigma_{ij} + \sum_j (1-\rho) \frac{\lambda_j}{\lambda_T} \sigma_{ij} \Big) \quad = \quad \frac{\Lambda_i \theta_i}{\lambda_T - \lambda_i} \Big(q_{i_f} - (1-\rho) \frac{\lambda_i}{\lambda_T} \Big) C_1 .$$

In arriving at the above expression, we used equation (4.14.a) and the fact that $\sigma_{ii} = 0$. Hence, the proportion of the time the device travels empty out of station i is obtained as $e_i = E_i/C_1$. Substituting for q_{i_f} using proposition 4.2, e_i is given by

$$e_i \quad = \quad \frac{\Lambda_i \theta_i}{\lambda_T - \lambda_i} \frac{\lambda_i}{\Lambda_i \chi} (\lambda_T - \lambda_i) (\rho - \alpha_f - \varphi_i) \quad = \quad \frac{\lambda_i \theta_i}{\chi} (\rho - \alpha_f - \varphi_i). \qquad (4.17)$$

Since e_i is non-negative by definition, we must have $\rho - \alpha_f - \varphi_i > 0$. Hence, to have a stable system with $\rho < 1$, we require $\alpha_f + \varphi_i < 1$, i=1,...,M. Let station k be such that $\varphi_k = \max_i \varphi_i$. In particular, we must have $\alpha_f + \varphi_k < 1$. This is expressed as Proposition 4.3. (It can be easily shown that $\varphi_k \geq 0$, and so we must have $\alpha_f + \varphi_k > 0$.)

Proposition 4.3: A necessary condition for the stability of the system is

$$\alpha_f + \varphi_k \quad < \quad 1.$$

■

If the above condition is met, then we can compute ρ using the following iterative algorithm:

Algorithm to compute ρ:

i) Set n = 0. Start with an initial value of ρ^n, (say, $\rho^n = \alpha_f$).

ii) Compute $q^n_{i_f}$, i=1,...,M, using equation (4.16).

iii) Using these $q^n_{i_f}$ values, compute ρ from equation (4.4).

iv) Compute $\rho^{n+1} = \rho^n + \Delta(\rho - \rho^n)$, where Δ is a sufficiently small step size. Set n=n+1.

v) Iterate over steps ii) through iv) until two successive values of ρ^n are reasonably close.

Note, from equations (4.4) and (4.16), that in the above algorithm, we obtain ρ as the solution to the following fixed point problem:

$$1 - \rho \quad = \prod_i \Big\{ \frac{\lambda_i}{\Lambda_i \chi} (\lambda_T - \lambda_i) (\rho - \alpha_f - \varphi_i) + \frac{\lambda_i}{\lambda_T} (1 - \rho) \Big\}. \qquad (4.18)$$

Proposition 4.4: For the fixed point problem defined by equation (4.18), the above algorithm returns a unique solution for ρ if $\alpha_f + \varphi_k < 1$. This value of ρ lies between $\alpha_f + \varphi_k$ and 1.

■

A proof of Proposition 4.4 is provided in Appendix II.

Using Lemma 4.1, the cycle time is now obtained as follows. Since

$$q_{1_f} = \frac{v_1 - \lambda_1 C_1}{\Lambda_1 C_1} = \frac{1 - \lambda_1 C_1}{\Lambda_1 C_1},$$

collecting terms involving C_1 and using Proposition 4.2 to express q_{1_f}, we obtain :

$$C_1 = \frac{\dfrac{\chi}{\lambda_1 (\lambda_T - \lambda_1)}}{\rho - \alpha_f - \varphi_1 + \dfrac{\chi}{\lambda_T - \lambda_1}\left[1 + (1-\rho)\dfrac{\Lambda_1}{\lambda_T}\right]}. \qquad (4.19)$$

The terms v_i can now be easily computed using Lemma 4.1. Once C_1 and the v_i values are computed, then we can determine C_i, $i = 2,\ldots,M$, using the relationship $v_i C_i = v_1 C_1$. Note that the C_i and v_i values are provided only as additional information; that is, we do not need to explicitly compute their values in order to test the system for stability.

5. Extension to Multiple Devices

One possible way to extend the single device model to multiple-device systems is to use a "single faster device." That is, in order to model a system with K devices, we assume that there is a single device which travels K times faster (Kleinrock [8]). Using a proportionally faster single server to represent multiple servers has also been proposed by Dukhovnyy [4] who studied urban transportation systems using this approximation.

The single faster device is modeled simply by dividing all the σ_{ij} and τ_{ij} values by K. We realize that this approach is not likely to yield satisfactory results for expected waiting times. However, as shown in section 6, such an approximation seems to yield reasonably accurate results for device statistics and cycle times.

6. Model Evaluation and Numerical Experiments

In this section we will present numerical examples to study two fundamental issues. First, we will evaluate the performance of the MOD FCFS dispatching rule relative to those rules studied earlier in the AGV context. Second, we will evaluate the performance of the analytical model for single–device and multiple–device systems and compare the results obtained from it with those obtained from the three analytical models described in section 2. In making the above comparisons we will also use results obtained from a simulation model. Three different layouts, namely, L1, L2, and L3 were developed for the experiment. Each layout was analyzed with various data sets designated by DAT1, DAT2, etc. A fourth layout, which is identical to the one presented by

Egbelu [6], was also included in the study. The above layouts and the flow and travel time data associated with each are shown in Appendix III.

Consider first the performance of MOD FCFS. The results of a simulation experiment are shown in Table 1 where α_f, α_e, and α_I, denote the fraction of time that a device is traveling loaded, traveling empty, and waiting in an idle state, respectively. (All the simulation runs are based on 5,000 loaded trips per device per replication, and 10 replications.) For given flow data and travel time data it is straightforward to compute the α_f value which is shown as the third column in Table 1. The four dispatching rules of interest are listed across the Table as: FCFS, MFCFS, STTF, and MOD FCFS. For each rule, we present the values of α_e and α_I as well as the overall mean waiting time (in minutes) for any move request in the system.

The results shown in Table 1 indicate that FCFS and MFCFS do not yield satisfactory results. This is primarily due to the fact that neither rule attempts to reduce empty device travel. On the other hand, STTF and MOD FCFS yield comparable results. Recall that, under STTF, certain stations may not receive empty devices for long periods of time. This accounts for the unstable system encountered for layout 2 with a high throughput level under STTF. Shop locking issues aside, past results have established that STTF is an efficient rule. Since MOD FCFS yields results that are very close to those obtained with STTF, and since shop locking is less likely to occur with MOD FCFS (this may require further investigation), we conclude that MOD FCFS is a viable dispatching rule.

Consider next, the performance of our analytical model relative to other analytical models and simulation. Comparing the results presented under the two columns labeled MOD FCFS and our analytical model in Table 2, we observe that our model slightly (but consistently) overestimates the minimum number of devices required. On the other hand, comparing the results presented under STTF and the other three analytical models, we observe that model 2 shows the overall best performance although it slightly (but consistently) underestimates the minimum number of devices required under STTF. Note that determining the minimum number of devices required via simulation is not straightforward because this number, to some extent, depends on how one defines a "stable" system. Table 2 is based on three alternative measures for "stability", labeled A, B, and C. Such definitions of stability mostly reflect what is "acceptable performance" from a practical viewpoint rather than a theoretical viewpoint.

It is interesting to note that model 2 also yields fairly good estimates on the number of devices required under the MOD FCFS rule. However, studying the number of devices required alone can be misleading. A more accurate measure of model performance is the fraction of time that a device travels empty, that is, α_e. (Recall that, with no blocking or congestion delays, computing α_f is straightforward and that $\alpha_I = 1.0 - \alpha_f - \alpha_e$.) Hence, not considering blocking, the accuracy of a model depends on how well it can estimate α_e for a given dispatching rule.

Table 1. Performance of the four empty vehicle dispatching rules.
(All values are obtained from simulation, 10 replications)

Layout and throughput level	No. of devices	α_f/dev	FCFS			MFCFS			STTF			MOD FCFS		
			α_e/dev	α_l/dev	Overall mean waiting time	α_e/dev	α_l/dev	Overall mean waiting time	α_e/dev	α_l/dev	Overall mean waiting time	α_e/dev	α_l/dev	Overall mean waiting time
L1.DAT1	1	0.189	0.139	0.672	3.433	0.139	0.672	3.425	0.135	0.675	3.261	0.137	0.674	3.318
L1.DAT2	1	0.323	0.223	0.454	5.833	0.222	0.455	5.758	0.208	0.469	4.942	0.213	0.464	5.222
L1.DAT3	1	0.433	0.301	0.266	11.123	0.296	0.271	10.349	0.258	0.309	6.971	0.270	0.297	7.788
L1.DAT4	1	0.650		unstable[1]			unstable		0.271	0.078	16.867	0.290	0.060	21.044
L1.DAT5	1	0.814		unstable			unstable		0.180	0.005	76.790	0.182	0.004	61.220
L2.DAT1	4	0.244	0.166	0.590	1.389	0.166	0.590	1.390	0.164	0.591	1.372	0.165	0.591	1.334
L2.DAT2	4	0.327	0.222	0.451	1.561	0.222	0.451	1.555	0.216	0.457	1.484	0.218	0.458	1.459
L2.DAT3	4	0.492	0.336	0.172	3.341	0.336	0.172	3.366	0.289	0.219	1.998	0.306	0.202	2.254
L2.DAT4	4	0.706		unstable			unstable		0.264	0.030	5.011	0.279	0.015	6.853
L2.DAT5	4	0.827		unstable			unstable			unstable		0.173	0.000	20.130
L3.DAT1	8	0.220	0.193	0.587	1.410	0.193	0.587	1.410	0.192	0.588	1.405	0.193	0.587	1.336
L3.DAT2	8	0.315	0.276	0.409	1.481	0.277	0.408	1.484	0.271	0.414	1.442	0.274	0.410	1.365
L3.DAT3	8	0.443	0.389	0.168	2.176	0.390	0.167	2.177	0.347	0.210	1.567	0.368	0.189	1.670
L3.DAT4	8	0.748		unstable			unstable		0.247	0.005	3.437	0.252	0.000	7.138
L3.DAT5	8	0.818		unstable			unstable		0.182	0.000	13.600	0.182	0.000	10.790
Layout 4	9	0.531		unstable			unstable		0.458	0.011	7.454	0.457	0.012	6.155

(1) System is assumed to be unstable if the maximum number of jobs in any output buffer reaches 500.

Table 2. Minimum number of devices required to meet the given throughput.

Layout and throughput level	Simulation						The MOD FCFS analytic model	Model 1 (Beisteiner)	Model 2 (Beisteiner)	Model 3 (Egbelu)
	STTF			MOD FCFS						
	A	B	C	A	B	C				
L1.DAT3	1	1	1	1	1	1	1	1	1	1
L1.DAT4	1	1	1	1	1	1	1	2	1	2
L1.DAT5	1	2	2	1	1	2	2	2	1	2
L2.DAT3	3	3	3	3	3	3	3	4	3	4
L2.DAT4	4	4	4	4	4	4	5	5	4	5
L2.DAT5	5	5	5	4	4	5	5	6	4	6
L3.DAT3	5	5	5	4	5	5	6	6	4	7
L3.DAT4	8	8	8	7	7	8	9	10	7	12
L3.DAT5	8	9	9	8	8	9	10	11	7	13
Layout 4	8	9	9	8	9	9	10	9	9	13

A : System is assumed to be stable if Max q_i is less than 500, where q_i is the length of the output buffer i.

B : System is assumed to be stable if Max q_i is less than 500 and Max $E[q_i]$ is less than 10.

C : System is assumed to be stable if Max q_i is less than 500 and Max $E[q_i]$ is less than 5.

Table 3. Detailed evaluation of Model 2 and the MOD FCFS analytical model.

Layout and throughput level	Model 2					Simulation (MOD FCFS)				The MOD FCFS analyt. model
	Estimated devices	Required devices	α_f/dev	α_e/dev	α_l/dev	No. of devices	α_f/dev	α_e/dev	α_l/dev	α_e/dev
L1.DAT3	0.477	1	0.443	0.033	0.524	1	0.433	0.270	0.297	0.287
L1.DAT4	0.715	1	0.665	0.050	0.285	1	0.650	0.290	0.060	0.316
L1.DAT5	0.893	1	0.831	0.062	0.107	1	0.814	0.182	0.004	Unstable
						2	0.407	0.260	0.333	0.273
L2.DAT3	2.148	3	0.674	0.042	0.284	3	0.658	0.305	0.037	0.323
						4	0.492	0.306	0.202	0.320
L2.DAT4	3.069	4	0.722	0.045	0.233	4	0.706	0.279	0.015	Unstable
						5	0.564	0.324	0.112	0.345
L2.DAT5	3.580	4	0.843	0.052	0.105	4	0.827	0.173	0.000	Unstable
						5	0.659	0.307	0.034	0.323
						6	0.549	0.323	0.128	0.341
L3.DAT3	**3.789**	4	0.915	0.032	0.053	4	0.894	0.105	0.001	Unstable
						5	0.714	0.285	0.001	Unstable
						6	0.594	0.381	0.025	0.389
						7	0.514	0.396	0.090	0.415
L3.DAT4	**6.316**	7	0.872	0.030	0.098	7	0.855	0.145	0.000	Unstable
						8	0.748	0.252	0.000	Unstable
						10	0.598	0.380	0.022	0.389
						12	0.498	0.392	0.110	0.409
L3.DAT5	6.890	7	0.951	0.033	0.016	8	0.818	0.182	0.000	Unstable
						9	0.726	0.273	0.001	Unstable
						11	0.594	0.381	0.025	0.394
						13	0.503	0.393	0.104	0.410
Layout 4	8.003	9	0.530	0.359	0.111	9	0.531	0.457	0.012	Unstable
						10	0.478	0.487	0.035	0.524
						11	0.434	0.500	0.066	0.556

According to the results obtained from the simulation model, model 2 also generates reasonably good estimates on the number of devices required. However, as shown in Table 3, model 2 severely underestimates the value of α_e (resulting in a severe overestimation of α_I). That is, model 2 mostly generates misleading results in terms of device statistics. On the other hand, as shown in Table 3, our model generates reasonably good estimates for α_e for all the layout and throughput level combinations tested.

We further evaluated the performance of our model by comparing the analytical cycle times obtained for each station with those obtained from simulation. The results for layout 1 are shown in Table 4 which indicates that the model performs reasonably well from a cycle time viewpoint as well. Note that the fifth column in Table 4 (i.e., simulation results obtained for a single faster device) enables us to evaluate the impact of approximating several devices by a single faster device. The results indicate that such an approximation seems to generate satisfactory results.

Obviously, the larger the number of devices in the actual system, the more error we would expect from the above approximation. However, as shown in Table A10 in Appendix III (where we present cycle time results for layout 2 and layout 3), even with 10 devices, a single faster device yields reasonably accurate results.

7. Conclusions

In this study, we derived a necessary and sufficient condition for stability in trip–based material handling systems. To our knowledge the model we present is the first analytical model which explicitly incorporates a dispatching rule in the context of trip–based material handling systems. Unlike existing models, our model also gives the cycle time for each station. Moreover, one could run our model with different number of devices and obtain device statistics and station cycle times quite rapidly. This would not be possible with existing models since they only give a single estimate on the number of devices required. Furthermore, their performance is unpredictable since their accuracy mostly depends on the layout and (other than FCFS), they do not explicitly capture the empty device dispatching rule.

We also empirically observed that a single fast device serves as a useful approximation to model multiple-device trip-based material handling systems as far as the expected device utilization is concerned. Lastly, the dispatching rule we used in the study, namely, MOD FCFS, is simple, efficient, and was not analyzed before. This rule was empirically shown to be nearly as good as the Shortest–Travel–Time–First (STTF) rule. It is instructive to note that, assuming the pick-up point immediately follows the deposit point of each station, as the throughput requirement is increased, the MOD FCFS rule will approach the STTF rule since a device is very likely to find a load at a pick–up point after it delivers a load at the corresponding deposit point.

Table 4. Cycle times obtained from the simulation model and the MOD FCFS analytical model. (Layout 1)

Layout and throughput level	No. of devices	Station No.	Multiple devices simulation	Single faster device simulation	MOD FCFS analytical model
L1.DAT3	1	1	21.146 ± 0.240	Same as Multiple device case	20.822
		2	76.538 ± 1.715		76.672
		3	18.519 ± 0.240		18.058
		4	26.714 ± 0.624		26.300
		5	25.541 ± 0.426		25.154
		6	23.852 ± 0.570		23.649
		7	75.163 ± 3.490		75.603
L1.DAT4	1	1	16.451 ± 0.141	Same as Multiple device case	16.594
		2	51.000 ± 1.157		51.115
		3	15.016 ± 0.166		14.103
		4	21.086 ± 0.296		20.455
		5	20.251 ± 0.241		19.572
		6	19.077 ± 0.251		18.411
		7	54.911 ± 1.990		58.502
L1.DAT5	2	1	10.897 ± 0.138	11.037 ± 0.140	10.980
		2	40.789 ± 0.921	40.788 ± 0.916	40.892
		3	9.500 ± 0.124	9.665 ± 0.149	9.524
		4	13.806 ± 0.348	13.998 ± 0.340	13.891
		5	13.240 ± 0.211	13.357 ± 0.228	13.284
		6	12.337 ± 0.290	12.477 ± 0.284	12.486
		7	39.494 ± 1.803	39.662 ± 1.860	40.005

References:

1. Beisteiner, F., "Strategies for the Employment of Vehicles in an Automated Guided Transportation System," *Proc. of the 2nd Intl. Conf. on Automated Guided Vehicle Systems*, Stuttgart, W. Germany, June 7–9, 1983.

2. Chow, W.M., "Design for Line Flexibility," *IIE Transns.*, Vol. 18 No. 1, pp. 95–108, 1986.

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

4. Dreyer, J., "12 Good Reasons for Computer Terminals on Industrial Trucks !," *Modern Materials Handling*, Vol. 35, No. 12, pp. 42–45, 1980.

5. Dukhovnyy, I.M., "An Approximate Model of Motion of Urban Passenger Transportation over Annular Routes," *Engineering Cybernetics*, Vol. 17, No. 1, pp. 161–162, 1979.

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

7. Egbelu, P.J. and J.M.A. Tanchoco, "Characterization of Automatic Guided Vehicle Dispatching Rules," *Intl. Journal of Production Research*, Vol. 22, No. 3, pp. 359–374.

8. Kleinrock, L, *Queueing Systems, Vol II - Computer Applications*, John Wiley, 1986.

9. Koff, G.A., "Automatic Guided Vehicle Systems: Applications, Controls, and Planning," *Matl. Flow*, Vol. 4, pp. 3–16, 1987.

10. Leung, L.C., S.K. Khator and D.L. Kimbler, "Assignment of AGVS with Different Vehicle Types," *Matl. Flow*, Vol. 4, pp. 65–72, 1987.

11. Maxwell, W.L. and J.A. Muckstadt, "Design of Automatic Guided Vehicle Systems," *IIE Transns.*, Vol. 14, No. 2, pp. 114–124, 1982.

12. Solberg, J. J., "The Optimal Planning of Computerized Manufacturing Systems: CAN–Q User's Guide," School of Indl. Engineering, Purdue University, West Lafayette, IN, 1980.

13. Tanchoco, J.M.A., P.J. Egbelu and F. Taghaboni, "Determination of the Total Number of Vehicles in an AGV–Based Material Transport System," *Matl. Flow*, Vol.4, pp. 33–51, 1987.

14. Tompkins, J.A. and J.A. White, *Facilities Planning*, John Wiley, 1984.

15. Toro-Ramos, Z.R. and L.F. McGinnis, "Performance Modeling for a Single Material Handling Device with Random Service Requests - Part I: Pure Blocking," *Proc. of the 1990 Material Handling Research Colloquium*, pp. 271-283, Hebron, Kentucky, 1990.

16. Toro-Ramos, Z.R. and L.F. McGinnis, "Performance Modeling for a Single Material Handling Device with Random Service Requests - Part II: Blocking with Recourse," *Proc. of the 1990 Material Handling Research Colloquium*, pp. 285-297, Hebron, Kentucky, 1990.

17. Wysk, R.A., P.J. Egbelu, C. Zhou and B.K. Ghosh, "Use of Spread Analysis for Evaluating AGV Systems," *Matl. Flow*, Vol. 4, pp. 53–64, 1987.

18. 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.

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

20. "Cpmputers on Trucks – the Future is Now !" *Modern Materials Handling*, Vol. 36, No. 2, pp. 54-57, 1981.

21. "Now, Real–Time Control of In–Process Inventory with a Lift Truck System," *Modern Materials Handling*, Vol. 36, No. 6, pp. 66–69, 1981.

22. "RF Terminals Key to Faster Response," *Modern Materials Handling*, Vol. 44, No. 11, pp. 146, 1990.

Appendix I: Proof of Proposition 4.2

Proposition 4.2:

$$q_{i_f} \quad = \quad \frac{\lambda_i}{\Lambda_i \chi}(\lambda_T - \lambda_i)\,(\rho - \alpha_f - \frac{\hat{\chi}}{\lambda_T} + \frac{\Lambda_i\,\chi}{\lambda_i\,\lambda_T}) \;+\; \frac{\lambda_i}{\lambda_T}(1-\rho). \qquad (I.1)$$

Proof: From equations (4.8) and (4.9.b),

$$v_i \;=\; \sum_j \Lambda_j C_1\,(\rho-1+q_{j_f})\frac{\lambda_i}{\lambda_T - \lambda_j} + (1-\rho)\lambda_i C_1 \;+ \Lambda_i C_1 \;-\; \Lambda_i C_1\,(\rho-1+q_{i_f})\frac{\lambda_i}{\lambda_T - \lambda_i}. \qquad (I.2)$$

From equations (4.11) and (I.2), we obtain

$$\Lambda_i q_{i_f} \quad = \quad \sum_j \Lambda_j\,(\rho-1+q_{j_f})\frac{\lambda_i}{\lambda_T - \lambda_j} + (1-\rho)\lambda_i \;+ (\Lambda_i \;-\; \lambda_i) \;-\; \Lambda_i\,(\rho-1+q_{i_f})\frac{\lambda_i}{\lambda_T - \lambda_i},$$

and so this gives

$$\frac{\Lambda_i q_{i_f}}{\lambda_T - \lambda_i}\lambda_T \quad = \quad \lambda_i(1-\rho)[1+\frac{\Lambda_i}{\lambda_T - \lambda_i} - \sum_j \frac{\Lambda_j}{\lambda_T - \lambda_j}] + (\Lambda_i - \lambda_i) + \lambda_i \sum_j \frac{\Lambda_j\, q_{j_f}}{\lambda_T - \lambda_j}. \qquad (I.3)$$

From equations (4.8), (4.9), (4.10), (4.12), (4.13) and (4.14.a), noting that p_{ii} and σ_{ii} are both equal to zero, we obtain:

$$\begin{aligned} C_1 \quad &= \quad \sum_i (v_{i_e} + v_{i_f}\,\bar{q}_{i_f}) \sum_j p_{ij}\tau_{ij} + \sum_i \sum_j v_{i_f}\sigma_{ij}\,\Big[(\rho-1+q_{i_f})\frac{\lambda_j}{\lambda_T - \lambda_i} + (1-\rho)\frac{\lambda_j}{\lambda_T}\Big] + (1-\rho)C_1 \\ &= \quad \alpha_f C_1 + (1-\rho)C_1\Big[\sum_i \sum_j \Lambda_i \sigma_{ij}\,(\frac{\lambda_j}{\lambda_T} - \frac{\lambda_j}{\lambda_T - \lambda_i}) + 1\Big] + \sum_i \sum_j \frac{\Lambda_i C_1 q_{i_f}}{\lambda_T - \lambda_i}\lambda_j \sigma_{ij}. \\ &= \quad \alpha_f C_1 + (1-\rho)C_1\Big[1 - \sum_i \frac{\Lambda_i\,\lambda_i\,\theta_i}{\lambda_T\,(\lambda_T - \lambda_i)}\Big] + \sum_i \frac{\Lambda_i q_{i_f}}{\lambda_T - \lambda_i}\theta_i C_1, \end{aligned}$$

and so, cancelling out the term C_1 on both sides of the above expression, we get:

$$\sum_i \frac{\Lambda_i q_{i_f}}{\lambda_T - \lambda_i}\theta_i \quad = \quad (1-\rho)\sum_i \frac{\Lambda_i\,\lambda_i\,\theta_i}{\lambda_T\,(\lambda_T - \lambda_i)} \;+ \rho \;-\; \alpha f. \qquad (I.4)$$

Multiplying equation (I.3) by θ_i, and summing over all i, we get:

$$\lambda_T \sum_i \frac{\Lambda_i q_{i_f}}{\lambda_T - \lambda_i}\theta_i = (1-\rho)[\chi + \sum_i \frac{\lambda_i \Lambda_i}{\lambda_T - \lambda_i}\theta_i - \chi \sum_j \frac{\Lambda_j}{\lambda_T - \lambda_j}] + \hat{\chi} - \chi + \chi \sum_i \frac{\Lambda_j q_{j_f}}{\lambda_T - \lambda_j}. \qquad (I.5)$$

Comparing equations (I.4) and (I.5), we obtain

$$\sum_i \frac{\Lambda_j q_{j_f}}{\lambda_T - \lambda_i} \quad = \quad \lambda_T \frac{\rho - \alpha_f}{\chi} + (1-\rho)\sum_j \frac{\Lambda_j}{\lambda_T - \lambda_j} + \rho \;-\; \frac{\hat{\chi}}{\chi}. \qquad (I.6)$$

From equations (I.3) and (I.6), after some elementary algebra, we obtain the desired result. ■

Appendix II: Proof of Proposition 4.4

Proposition 4.4: For the fixed point problem defined by

$$1-\rho \;=\; \prod_i \left\{\frac{\lambda_i}{\Lambda_i \chi}(\lambda_T-\lambda_i)(\rho-\alpha_f-\varphi_i)+\frac{\lambda_i}{\lambda_T}(1-\rho)\right\}, \qquad \text{(II.1)}$$

there exists a unique solution for ρ, if $\alpha_f + \varphi_k < 1$, where k is the index of the station with the maximum value of φ_i. This value of ρ lies between $\alpha_f + \varphi_k$ and 1.

Proof: Let $f(\rho) = 1-\rho$; and $g(\rho) = \prod_i \left\{\frac{\lambda_i}{\Lambda_i \chi}(\lambda_T-\lambda_i)(\rho-\alpha_f-\varphi_i)+\frac{\lambda_i}{\lambda_T}(1-\rho)\right\}$.

We are seeking a fixed point solution to the equation:

$$f(\rho) \;=\; g(\rho);\ \alpha_f+\varphi_k \le \rho < 1. \qquad \text{(II.2)}$$

Note that at $\rho_0 = \alpha_f + \varphi_k$, $f(\rho_0) > g(\rho_0)$. To see this, observe that

$$g(\rho_0) \;=\; \frac{\lambda_k}{\lambda_T}(1-\rho_0)\Big[\prod_{i\neq k}\left\{\frac{\lambda_i}{\Lambda_i \chi}(\lambda_T-\lambda_i)(\rho_0-\alpha_f-\varphi_i)+\frac{\lambda_i}{\lambda_T}(1-\rho_0)\right\}\Big]. \qquad \text{(II.3)}$$

Since $f(\rho_0) = 1-\rho_0$, and $\lambda_k/\lambda_T < 1$, we observe that in order to show $f(\rho_0) > g(\rho_0)$, it is enough to show that $\prod_{i\neq k} h_k(i) < 1$, where

$$h_k(i) \;=\; \frac{\lambda_i}{\Lambda_i \chi}(\lambda_T-\lambda_i)(\varphi_k-\varphi_i)+\frac{\lambda_i}{\lambda_T}(1-\rho_0) \;=\; \frac{\lambda_i}{\Lambda_i \chi}(\lambda_T-\lambda_i)\Big(\frac{\Lambda_i}{\lambda_i}-\frac{\Lambda_k}{\lambda_k}\Big)\frac{\chi}{\lambda_T}+\frac{\lambda_i}{\lambda_T}(1-\rho_0)$$

$$=\; 1-\frac{\lambda_i}{\lambda_T}+\frac{\lambda_i}{\lambda_T}(1-\rho_0)-\frac{\Lambda_k}{\lambda_k}\frac{\lambda_i}{\Lambda_i}\frac{\lambda_T-\lambda_i}{\lambda_T} \;<\; 1.$$

Note, also, that as $\rho \rightarrow 1$, $g(\rho) > f(\rho)$. To see this, observe that

$$g(\rho) \;>\; \prod_i \left\{\frac{\lambda_i}{\Lambda_i \chi}(\lambda_T-\lambda_i)(\rho-\alpha_f-\varphi_i)\right\}.$$

Hence, $g(\rho)|_{\rho\to 1} > 0$. However, $f(\rho)|_{\rho\to 1} = 0$. That is, $g(\rho)|_{\rho\to 1} > f(\rho)|_{\rho\to 1}$.

It is thus clear that there is at least one value of ρ, in the region between $\alpha_f + \varphi_k$ and 1, where $f(\rho) = g(\rho)$. This demonstrates the existence of a fixed point. To show that this point is unique, we first rewrite $g(\rho)$ as

$$g(\rho) \;=\; \prod_i \beta_i\big(\eta_i(1-\rho)+(\rho-\alpha_f-\varphi_i)\big),$$

where ,

$$\beta_i \;=\; \frac{\lambda_i}{\Lambda_i \chi}(\lambda_T-\lambda_i); \quad \text{and} \quad \eta_i \;=\; \frac{\Lambda_i \chi}{\lambda_T(\lambda_T-\lambda_i)}. \qquad \text{(II.4)}$$

We now differentiate $f(\rho)$ and $g(\rho)$ with respect to ρ. This gives $f'(\rho) = -1$, and

$$g'(\rho) \quad = \quad \sum_{j}\Big[\prod_{i\neq j}\beta_i\big(\eta_i(1-\rho)+(\rho-\alpha_f-\varphi_i)\big)\Big]\,\beta_j(-\eta_j+1)$$

$$= \quad \sum_{j}\Big[\prod_{i}\beta_i\big(\eta_i(1-\rho)+(\rho-\alpha_f-\varphi_i)\big)\Big]\,\frac{\beta_j(1-\eta_j)}{\beta_j(\eta_j(1-\rho)+\rho-\alpha_f-\varphi_j)}$$

$$= \quad g(\rho)\sum_{j}\frac{1-\eta_j}{\eta_j(1-\rho)+\rho-\alpha_f-\varphi_j} \quad = \quad g(\rho)(a(\rho)+b(\rho)),$$

where

$$a(\rho) \quad = \quad \sum_{j:\eta_j\leq 1}\frac{1-\eta_j}{\eta_j(1-\rho)+\rho-\alpha_f-\varphi_j} \quad = \quad \sum_{j:\eta_j\leq 1}\frac{1-\eta_j}{\rho(1-\eta_j)+\eta_j-\alpha_f-\varphi_j}$$

and

$$b(\rho) \quad = \quad \sum_{j:\eta_j>1}\frac{1-\eta_j}{\eta_j(1-\rho)+\rho-\alpha_f-\varphi_j} \quad = \quad \sum_{j:\eta_j>1}\frac{(1-\eta_j)}{(1-\rho)(\eta_j-1)+1-\alpha_f-\varphi_j}.$$

We now prove, by contradiction, that there is only one solution for ρ, in the region $[\alpha_f+\varphi_k,1)$. Suppose, on the contrary, that there are multiple solutions in this region. Since $f(\rho) > g(\rho)$ at $\rho=\alpha_f+\varphi_k$, and $f(\rho)|_{\rho\to 1} < g(\rho)|_{\rho\to 1}$, this implies that the functions cross exactly an odd number of times in the region $[\alpha_f+\varphi_k,1)$. Let these points be designated $\rho_1,\ldots,\rho_{2n+1}$, with $\rho_1 <\ldots< \rho_{2n+1}$, for some $n > 0$, and consider the first three points, ρ_1, ρ_2 and ρ_3.

Note that we must have $g'(\rho_1) > f'(\rho_1)$, $g'(\rho_2) < f'(\rho_2)$, and $g'(\rho_3) > f'(\rho_3)$. Since $f'(\rho_i) = -1$ for all i, in particular this implies that $g'(\rho_2) < g'(\rho_3)$. Clearly, at these points of intersection, $g(\rho)=f(\rho)$, and so we also have:

$$g'(\rho_i) \quad = \quad (1-\rho_i)\,(a(\rho_i)+b(\rho_i)), \quad i = 1,\ldots,3.$$

Observe that as ρ_i increases in value, the terms $a(\rho_i)$, $b(\rho_i)$ and $(1-\rho_i)$ all decrease in value. Hence we have $g'(\rho_1) > g'(\rho_2) > g'(\rho_3)$. This immediately contradicts the earlier claim that $g'(\rho_2) < g'(\rho_3)$. ■

Appendix III: Input data for the numerical experiments

The numerical experiment is based on three layouts labeled L1, L2, and L3. For all three layouts, it is assumed that the device travels at a speed of 15 distance units per minute (except for L3 where it travels at 30 distance units per minute) and that the load or unload time is equal to 1/3 minutes. The first layout, namely, L1, is shown in Figure A1 where stations 1 and 2 are the I/O stations. (Note that no jobs are received through station 2.) The routing matrix and the distance matrix are presented in Table A1 and Table A2. The interarrival time for jobs received through station 1 is shown in Table A3 for low, medium, and high throughput levels. Lastly, for L1, it is assumed that the processing times are exponentially distributed and the travel times (loaded or empty) are constant.

Similar data for L2 (shown in Figure A2) are presented in Table A4, Table A5, and Table A6. For L2, the processing times are assumed to be exponentially distributed while the empty travel times are assumed to be uniformly distributed between 0.8z and 1.2z, where z reflects the mean empty travel time obtained by dividing each entry in Table A5 by the device speed. The loaded travel time is obtained by simply adding the (constant) load and unload times to uniformly distributed empty travel times.

Lastly, the data associated with L3 (shown in Figure A3) are presented in Table A7, Table A8, and Table A9. For L3, the processing times are exponentially distributed and the travel times (loaded or empty) are constant.

A fourth layout we included in the study is presented by Egbelu [6]. The reader may refer to his paper to view the layout, the distance/travel time matrix and other data. We used the same data except for the routing matrix: the last station for product types 4 and 5 is station 1 in our experiment, rather than station 8. This minor change in the routing matrix was made since our model, in its current form, does not allow I/O stations to operate as input–only stations. Note that $\bar{q}_{i_f}$ is not defined for these stations because the device never delivers a load at an input–only station. However, the model can be modified to accommodate such stations with relative ease.

Also, in the fourth layout, the pick–up and deposit points of each station are separated by a non–negligible distance. Nevertheless, following the MOD FCFS rule, we forced the device to always first inspect the corresponding pick–up point after it deposits a load at a deposit point. (Adjusting our model to capture the additional distance traveled is straightforward.) Although this may create some unnecessary empty device travel from a deposit point to the corresponding pick–up point, simulation results showed that it has a small impact on the throughput capacity of the system, especially since the device is likely to find a load waiting at the pick–up point as one increases the throughput level.

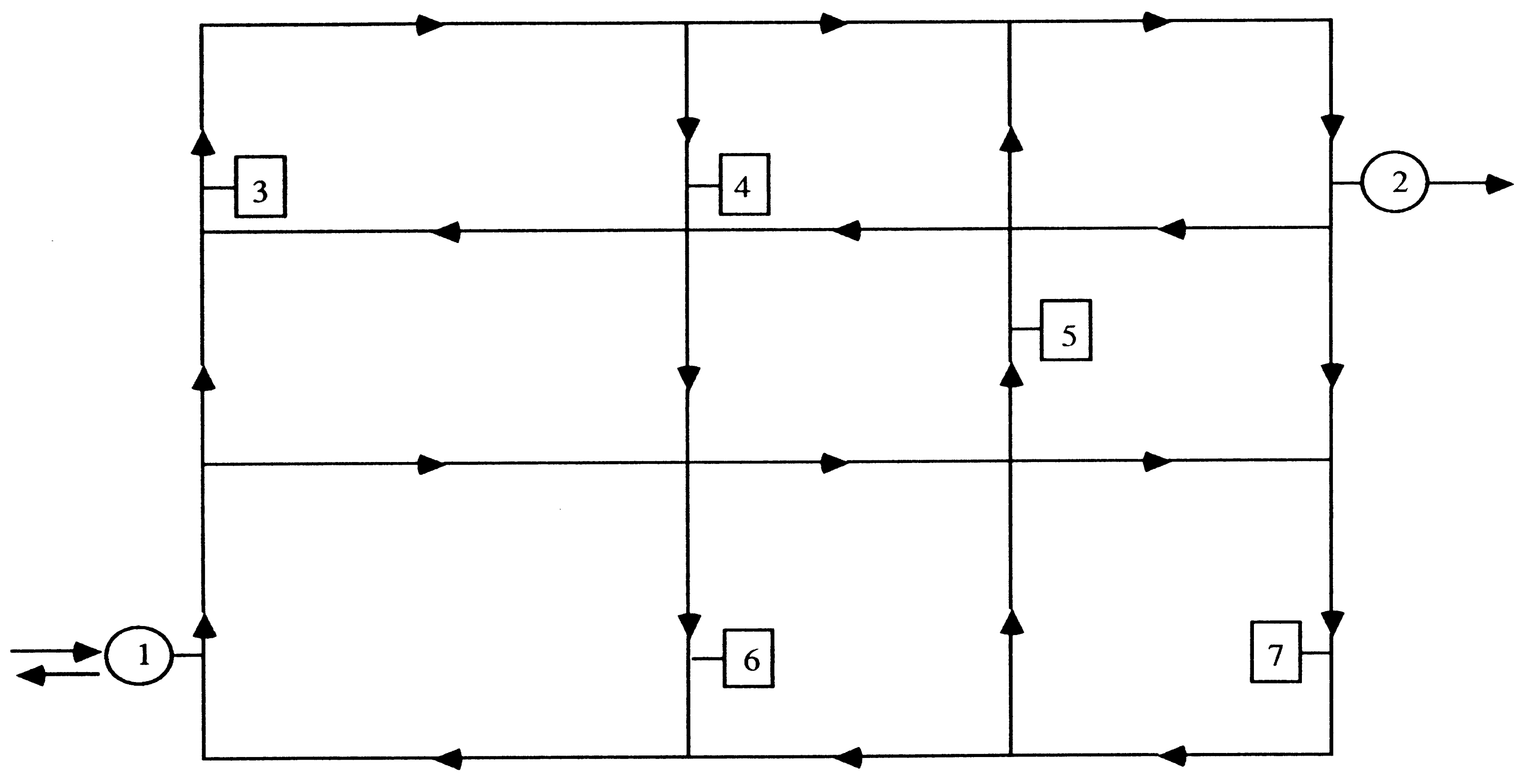

Figure A1. Layout 1 with 2 I/O and 5 processor stations.

○ : I/O station

□ : Processor station

Table A1. Routing matrix of jobs in Layout 1.

Station No.	1	2	3	4	5	6	7
1	0.0	0.0	0.5	0.5	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.5	0.0	0.0	0.0	0.5	0.0	0.0
4	0.0	0.0	0.0	0.0	0.3	0.7	0.0
5	0.0	0.5	0.1	0.0	0.0	0.4	0.0
6	0.2	0.0	0.5	0.0	0.0	0.0	0.3
7	0.0	0.3	0.1	0.6	0.0	0.0	0.0

Table A2. Vehicle travel distance between stations in Layout 1, distance units.

Station No.	1	2	3	4	5	6	7
1	0	62	16	42	36	28	48
2	58	0	38	64	44	36	16
3	64	46	0	26	50	42	62
4	38	50	18	0	24	16	36
5	50	26	30	56	0	28	42
6	22	84	38	64	58	0	70
7	42	54	58	84	28	56	0

Table A3. Mean interarrival time of jobs at the input station in Layout 1, mins/job.

Throughput level Input station	DAT1	DAT2	DAT3	DAT4	DAT5
1	60	40	30	20	16

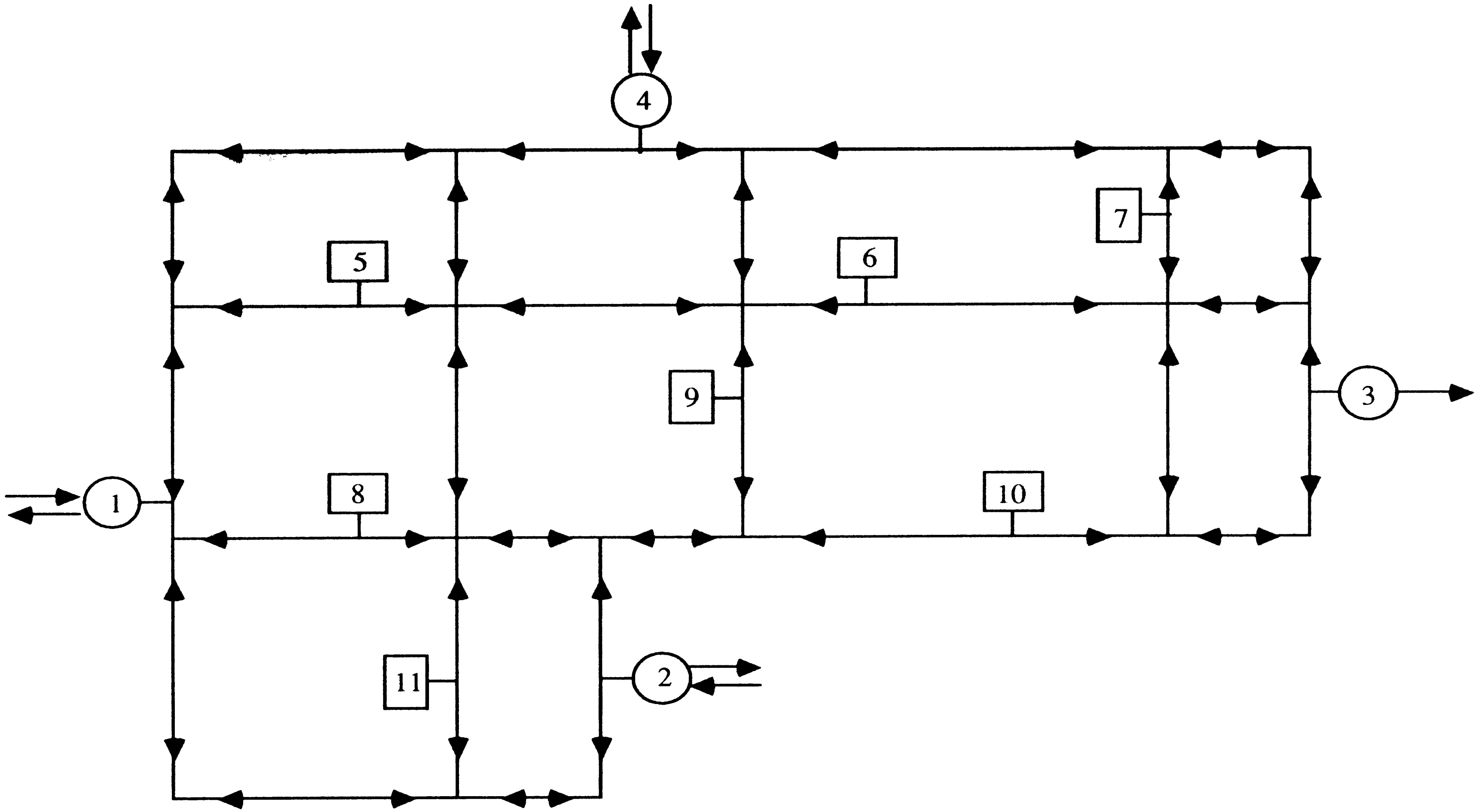

Figure A2. Layout 2 with 4 I/O and 7 processor stations.

○ : I/O station

□ : Processor station

Table A4. Routing matrix of jobs in Layout 2.

Station No.	1	2	3	4	5	6	7	8	9	10	11
1	0.0	0.0	0.0	0.0	0.2	0.2	0.0	0.2	0.2	0.0	0.2
2	0.0	0.0	0.0	0.0	0.2	0.1	0.1	0.2	0.1	0.1	0.2
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.2	0.4	0.0	0.1	0.1	0.1	0.1
5	0.1	0.1	0.1	0.2	0.0	0.1	0.0	0.1	0.1	0.1	0.1
6	0.0	0.0	0.1	0.1	0.1	0.0	0.2	0.1	0.2	0.2	0.0
7	0.0	0.1	0.2	0.1	0.0	0.2	0.0	0.1	0.2	0.0	0.1
8	0.2	0.0	0.0	0.0	0.2	0.1	0.1	0.0	0.2	0.1	0.1
9	0.0	0.2	0.1	0.0	0.1	0.2	0.1	0.2	0.0	0.1	0.0
10	0.0	0.2	0.2	0.1	0.0	0.1	0.0	0.1	0.1	0.0	0.2
11	0.2	0.0	0.1	0.1	0.2	0.0	0.1	0.0	0.1	0.2	0.0

Table A5. Vehicle travel distance between stations in Layout 2, distance units.

Station No.	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 A6. Mean interarrival time of jobs at the input stations in Layout 2, mins/job.

Throughput level Input station	DAT1	DAT2	DAT3	DAT4	DAT5
1	14	10	7	4.9	4.2
2	28	20	14	9.8	8.4
4	42	40	21	14.7	12.6

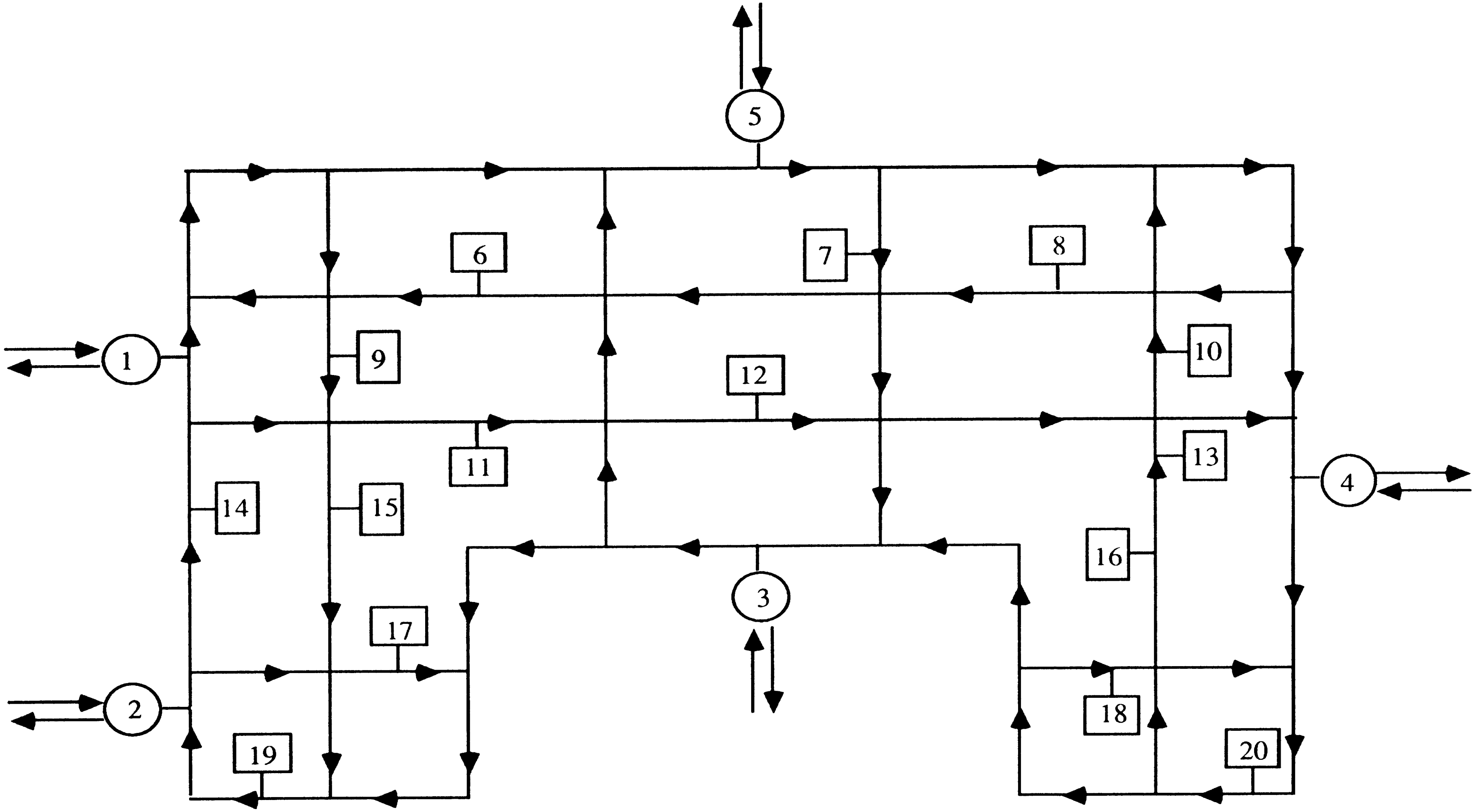

Figure A3. Layout 3 with 5 I/O and 15 processor stations.

○ : I/O station

□ : Processor station

Table A7. Routing matrix of jobs in Layout 3.

Station No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.2	0.0	0.0	0.2	0.2	0.0	0.2	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.2	0.2	0.0	0.2	0.0	0.2	0.0
3	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.2	0.2	0.2	0.0	0.0	0.0	0.0	0.0	0.2	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.2	0.0	0.0	0.2	0.0	0.0	0.2	0.0	0.2	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.2	0.2	0.2	0.2	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.15	0.0	0.2	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.2	0.05	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0
7	0.0	0.0	0.2	0.1	0.0	0.1	0.0	0.2	0.0	0.1	0.0	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.1
8	0.0	0.0	0.0	0.1	0.2	0.0	0.1	0.0	0.0	0.0	0.1	0.0	0.1	0.0	0.0	0.1	0.0	0.1	0.0	0.2
9	0.2	0.05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.1	0.15	0.0	0.1	0.0	0.1	0.0	0.1	0.0
10	0.0	0.0	0.1	0.2	0.0	0.0	0.1	0.1	0.0	0.0	0.0	0.1	0.1	0.0	0.0	0.1	0.0	0.1	0.0	0.1
11	0.0	0.1	0.1	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.2	0.1	0.0	0.0	0.0	0.2	0.0
12	0.0	0.0	0.15	0.15	0.0	0.05	0.0	0.1	0.0	0.1	0.0	0.0	0.1	0.05	0.0	0.1	0.0	0.1	0.0	0.1
13	0.0	0.0	0.0	0.05	0.2	0.0	0.1	0.1	0.1	0.1	0.0	0.0	0.0	0.05	0.0	0.0	0.0	0.2	0.0	0.1
14	0.2	0.0	0.0	0.0	0.1	0.1	0.1	0.0	0.1	0.0	0.1	0.1	0.0	0.0	0.0	0.0	0.1	0.0	0.1	0.0
15	0.1	0.05	0.05	0.0	0.0	0.1	0.1	0.0	0.1	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.2	0.0	0.1	0.0
16	0.0	0.0	0.0	0.2	0.1	0.0	0.0	0.1	0.0	0.1	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.2	0.0	0.2
17	0.1	0.1	0.0	0.0	0.0	0.05	0.0	0.0	0.0	0.0	0.15	0.0	0.0	0.2	0.2	0.0	0.0	0.0	0.2	0.0
18	0.0	0.0	0.2	0.1	0.0	0.0	0.1	0.0	0.0	0.1	0.0	0.0	0.1	0.0	0.0	0.1	0.0	0.0	0.0	0.3
19	0.2	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.25	0.1	0.1	0.0	0.05	0.1	0.0	0.1	0.0	0.0	0.0
20	0.0	0.0	0.0	0.14	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.2	0.06	0.0	0.2	0.0	0.2	0.0	0.0

Table A8. Vehicle travel distance between stations in Layout 3, distance units.

station No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	46	52	60	28	42	36	61	20	55	28	38	91	54	26	87	35	91	40	74
2	14	0	46	54	42	36	50	55	34	49	22	32	85	8	20	81	9	85	14	68
3	48	34	0	38	26	20	34	39	28	33	36	16	69	42	34	65	43	69	28	52
4	90	76	42	0	68	62	76	41	70	35	78	58	31	84	76	27	85	31	70	14
5	72	58	24	32	0	24	8	33	32	27	40	40	63	66	38	59	47	63	52	46
6	48	34	40	48	36	0	44	49	8	43	16	26	79	42	14	75	23	79	28	62
7	64	50	16	24	22	16	0	25	24	19	32	32	55	58	30	51	39	55	44	38
8	69	55	21	29	27	21	35	0	29	24	37	37	60	63	35	56	44	60	49	43
9	40	26	32	40	28	22	36	41	0	35	8	18	71	34	6	67	15	71	20	54
10	75	61	27	25	33	27	41	6	35	0	43	43	56	69	41	52	50	56	55	39
11	62	48	24	32	20	14	28	33	22	27	0	10	63	56	28	59	37	63	42	46
12	62	48	14	22	40	34	48	23	42	17	50	0	53	56	48	49	57	53	42	36
13	79	65	31	9	37	31	45	10	39	4	47	47	0	73	45	36	54	40	59	23
14	6	32	38	46	34	28	42	47	26	41	14	24	77	0	12	73	21	77	26	60
15	34	20	66	74	62	56	70	75	54	69	42	52	105	28	0	101	9	105	14	88
16	83	69	35	13	41	35	49	14	43	8	51	51	4	77	49	0	58	44	63	27
17	35	21	67	75	63	57	71	76	55	70	43	53	106	29	41	102	0	106	15	89
18	89	75	41	19	47	41	55	20	49	14	57	57	10	83	55	6	64	0	69	13
19	20	6	52	60	48	42	56	61	40	55	28	38	91	14	26	87	15	91	0	74
20	76	62	28	26	54	48	62	27	56	21	64	44	17	70	62	13	71	17	56	0

Table A9. Mean interarrival time of jobs at the input stations in Layout 3, mins/job.

Throughput level Input station	DAT1	DAT2	DAT3	DAT4	DAT5
1	10	7	5	3	2.75
2	20	14	10	6	5.5
3	30	21	15	9	8.25
4	40	28	20	12	11.0
5	50	35	25	15	13.75

Table A10. Cycle times obtained from simulation and the MOD FCFS analytical model.
(Layout 2)

Layout and throughput level	No. of devices	Station No.	Multiple devices simulation	Single faster device Simulation	MOD FCFS analytical model
L2.DAT3	4	1	5.347 ± 0.058	5.417 ± 0.095	5.496
		2	7.814 ± 0.114	8.059 ± 0.263	7.865
		3	12.441 ± 0.203	12.675 ± 0.385	12.469
		4	9.457 ± 0.146	9.811 ± 0.326	9.558
		5	4.485 ± 0.063	4.591 ± 0.146	4.500
		6	4.266 ± 0.052	4.490 ± 0.176	4.265
		7	7.958 ± 0.138	7.949 ± 0.236	7.915
		8	4.717 ± 0.073	4.847 ± 0.219	4.708
		9	4.099 ± 0.064	4.306 ± 0.107	4.091
		10	5.785 ± 0.084	5.914 ± 0.144	5.738
		11	5.555 ± 0.110	5.634 ± 0.117	5.573
L2.DAT4	5	1	3.880 ± 0.048	3.944 ± 0.074	4.007
		2	5.664 ± 0.068	5.822 ± 0.153	5.683
		3	8.732 ± 0.116	8.871 ± 0.274	8.728
		4	6.829 ± 0.082	7.096 ± 0.180	6.867
		5	3.312 ± 0.033	3.391 ± 0.113	3.254
		6	3.156 ± 0.042	3.324 ± 0.116	3.085
		7	5.760 ± 0.089	5.778 ± 0.160	5.711
		8	3.504 ± 0.056	3.582 ± 0.141	3.403
		9	3.046 ± 0.045	3.205 ± 0.081	2.959
		10	4.218 ± 0.062	4.352 ± 0.116	4.144
		11	4.062 ± 0.080	4.139 ± 0.086	4.025
L2.DAT5	5	1	3.590 ± 0.044	3.667 ± 0.060	3.952
		2	5.328 ± 0.082	5.467 ± 0.176	5.416
		3	7.482 ± 0.099	7.603 ± 0.244	7.481
		4	6.350 ± 0.092	6.514 ± 0.166	6.419
		5	3.214 ± 0.052	3.271 ± 0.078	3.081
		6	3.072 ± 0.046	3.201 ± 0.113	2.921
		7	5.430 ± 0.072	5.485 ± 0.141	5.414
		8	3.390 ± 0.069	3.421 ± 0.122	3.223
		9	2.966 ± 0.037	3.081 ± 0.077	2.802
		10	4.050 ± 0.059	4.113 ± 0.094	3.927
		11	3.892 ± 0.080	3.951 ± 0.116	3.814

Table A10 (Contd.). Cycle times obtained from simulation and the MOD FCFS analytical model. (Layout 3)

Layout and throughput level	No. of devices	Station No.	Multiple devices simulation	Single faster device Simulation	MOD FCFS analytical model
L3.DAT3	8	1	3.291 ± 0.035	3.377 ± 0.079	3.329
		2	6.714 ± 0.101	6.872 ± 0.399	6.782
		3	5.843 ± 0.058	5.893 ± 0.266	5.925
		4	6.441 ± 0.103	6.563 ± 0.289	6.518
		5	9.213 ± 0.124	9.527 ± 0.556	9.389
		6	5.878 ± 0.135	5.835 ± 0.357	5.935
		7	6.203 ± 0.129	6.593 ± 0.361	6.262
		8	5.358 ± 0.133	5.623 ± 0.366	5.410
		9	5.640 ± 0.098	5.886 ± 0.279	5.772
		10	4.121 ± 0.062	4.231 ± 0.194	4.172
		11	3.123 ± 0.031	3.200 ± 0.101	3.143
		12	4.610 ± 0.090	4.668 ± 0.179	4.665
		13	4.065 ± 0.100	4.287 ± 0.239	4.100
		14	3.119 ± 0.041	3.242 ± 0.069	3.124
		15	4.158 ± 0.055	4.397 ± 0.222	4.202
		16	6.807 ± 0.146	6.993 ± 0.353	6.803
		17	4.160 ± 0.071	4.265 ± 0.180	4.199
		18	4.878 ± 0.083	5.077 ± 0.312	4.902
		19	3.794 ± 0.050	3.881 ± 0.208	3.777
		20	4.635 ± 0.115	4.698 ± 0.264	4.694
L3.DAT4	9	1	2.504 ± 0.034	2.502 ± 0.047	2.679
		2	4.735 ± 0.072	4.752 ± 0.237	5.563
		3	4.202 ± 0.070	4.182 ± 0.161	4.234
		4	4.503 ± 0.082	4.481 ± 0.145	4.512
		5	6.339 ± 0.065	6.379 ± 0.250	6.654
		6	4.301 ± 0.061	4.191 ± 0.182	4.499
		7	4.541 ± 0.100	4.686 ± 0.193	4.747
		8	3.986 ± 0.090	4.043 ± 0.214	4.100
		9	4.171 ± 0.073	4.222 ± 0.162	4.376
		10	3.151 ± 0.082	3.130 ± 0.082	3.159
		11	2.481 ± 0.036	2.458 ± 0.079	2.378
		12	3.480 ± 0.058	3.470 ± 0.138	3.534
		13	3.131 ± 0.078	3.190 ± 0.136	3.105
		14	2.481 ± 0.045	2.480 ± 0.072	2.363
		15	3.176 ± 0.046	3.218 ± 0.135	3.182
		16	4.926 ± 0.118	4.942 ± 0.212	5.159
		17	3.179 ± 0.044	3.153 ± 0.130	3.180
		18	3.670 ± 0.060	3.709 ± 0.181	3.715
		19	2.940 ± 0.039	2.884 ± 0.116	2.860
		20	3.496 ± 0.089	3.455 ± 0.176	3.556
L3.DAT5	10	1	2.261 ± 0.034	2.235 ± 0.043	2.394
		2	4.288 ± 0.053	4.292 ± 0.256	4.959
		3	3.794 ± 0.071	3.746 ± 0.158	3.826
		4	4.083 ± 0.073	4.048 ± 0.147	4.089
		5	5.744 ± 0.071	5.770 ± 0.246	6.016
		6	3.874 ± 0.053	3.794 ± 0.175	4.043
		7	4.081 ± 0.074	4.240 ± 0.141	4.266
		8	3.603 ± 0.071	3.653 ± 0.211	3.685
		9	3.766 ± 0.068	3.785 ± 0.133	3.932
		10	2.845 ± 0.069	2.829 ± 0.098	2.840
		11	2.229 ± 0.036	2.211 ± 0.072	2.138
		12	3.136 ± 0.055	3.101 ± 0.113	3.176
		13	2.825 ± 0.066	2.865 ± 0.126	2.791
		14	2.232 ± 0.035	2.230 ± 0.058	2.125
		15	2.857 ± 0.041	2.907 ± 0.100	2.860
		16	4.446 ± 0.096	4.437 ± 0.203	4.635
		17	2.863 ± 0.033	2.839 ± 0.103	2.858
		18	3.311 ± 0.051	3.314 ± 0.157	3.338
		19	2.644 ± 0.034	2.569 ± 0.100	2.571
		20	3.143 ± 0.074	3.082 ± 0.153	3.196