Proceedings of the 1984
Winter Simulation Conference
S. Sheppard, U. Pooch, D. Pegden (eds.)

A SIMULATION OPTIMIZATION APPROACH TO OPTIMUM STORAGE AND RETRIEVAL POLICIES IN AN AUTOMATED WAREHOUSING SYSTEM

Farhad Azadivar
Department of Industrial and Systems Engineering
University of Illinois at Chicago
P.O. Box 4348, Chicago, IL 60680

ABSTRACT

In this paper a computerized automated warehousing system is modeled through digital computer simulation. The purpose of the study is to use simulation as a means of optimizing the decision parameters of the system. Those parameters define the pattern of assignment of inventoried items to storage racks. Previous work by other researchers have shown that assignment of items based on their turnover time (length of stay in the warehouse) reduces the average travel time of the stacking crane used for storage and retrieval. This study is aimed at finding the the opitmum way in which the storage racks can be divided and allocated to items with various turnover times. Since the result of evaluating the system for a given policy is only an error corrupted observation, a stochastic optimization search has been employed. In this search the simulation model is used as the means of evaluating the value of the objective function for the levels of the decision variables supplied by the search method. The results of the optimization by simulation have been compared with the optimums obtained for a mathematical model. It is shown that for the discrete rack systems better optimums can be obtained by simulating the real system than approximating it with a mathematical model.

INTRODUCTION

With the recent awareness about the need for automating manufacturing and production systems computerized automated warehousing systems is one of the areas that deserves a lot of attention. A computerized automated warehousing system consists of a set of storage spaces (often in the form of racks) and one or more stacking cranes which are controlled by means of a mini or a micro computer. For items arriving for storage, a space is found by the computer, the crane is automatically given the instruction for the store process and the storage location is recorded by the computer. This information is subsequently utilized when the retrieval request for the same item arrives. Store and retrieve requests wait in queues for the cranes to become available. The computer assigns the crane to a store or retrieve request through an optimum assignment rule.

The problems to study with respect to automated warehousing systems could be associated with the hardware or the software. Hardware problems deal with the design of mechanisms and control of the operation of the crane and rack system. The software portion deals with the optimum decision rules to be adopted for assigning the items to racks and cranes to store and retrieve requests. This study is aimed at the latter aspect.

Hausman W.H., Schwarz, L.B. and Graves, C.G., in their outstanding work reported in [1], [2] and [3], discussed many aspects of decision rules for assigning items to storage spaces with and without interleaving. Interleaving is the process of combining a store and a retrieve operation so that one of each is performed in just one round trip of the crane. They presented a model approximating the discrete rack system with a continuous one. This model assumes that rack is a continuous surface where an item can be stored at any arbitrary point. Based on analysis of this model they developed several optimum policies which were tested for discrete case through digital computer simulation.

This paper is an extention of their work in the sense that basically the same assumptions are made on the system and similar decision rules are sought. However, here the optimal policies are developed employing a simulation model of the system rather than its continuous approximation. In addition the system is considered as a stochastic one and is optimized through stochastic optimization using stochastic approximation method.

In what follows, first a short description of the continuous model analyzed in [1], [2] and [3] along with the major results obtained will be presented. Then the simulation model used in this study will be explained. Next the stochastic optimization procedure employed will be introduced and the results of its application to determine the optimum decision rules for this system will be presented.

CONTINUOUS MODEL OF A WAREHOUSING SYSTEM

Hausman et al [1] consider the storage rack as a continuous square like the one shown in Figure 1. Point O is the input/output point (I/O Point) at which all the requests for storage and retrieval are handled. The horizontal and vertical speed of the crane is assigned such that from I/O point the same amount of time is required for the crane to reach the farthest horizontal or vertical point. In other words, the storage rack is square in time. The time it takes to move from the I/O point to a point with coordinates (x_1,x_2) is Max $[\frac{x_1}{s_1},\frac{x_2}{s_2}]$ where s_1 and s_2 are the speeds of the crane in x_1 and x_2 directions respectively.

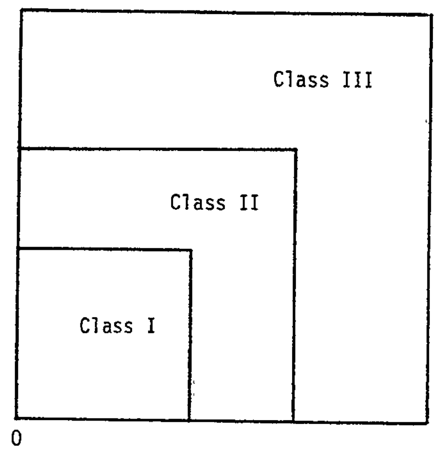

Figure 1: The Continuous Model of Storage Racks.

The turnover distribution is derived from the ABC rule for inventoried items. According to this rule, if items are ranked in a descending order based on their contribution to the overall demand often a small percent of items (α%) represent a large percent of the total volume of the demand (β%). Common values for α and β are α = 20% and β = 60%, 70%, 80% or 90%. For instance 20/80 rate (α=20%, β=80%) (which is used in this paper) indicates that 20% of items represent 80% of the total volume of the inventoried items.

In [3], under certain assumptions, the distribution of the turnover time (x) is derived as

$$F(x) = \left(\frac{zx}{\mu}\right)^{\frac{z}{1-z}} \quad \text{for} \quad 0 < x < \frac{\mu}{z} \qquad (1)$$

Where μ is the overall averge length of stay of items in the warehouse and $z=2s/(1+s)$. s is a factor which is determined by the type of ABC rule. For instance for 20/60, 20/70, 20/80 and 20/90 rules the values of s are .318, .222, .139 and .065 respectively.

Several decision rules regarding the assignment of items to storage racks and the crane to the store and retrieve requests have been examined. The simplest rule is the assignment of the arriving item for storage to the closest open location (COL rule) and first-in first-out (FIFO) rule for store and retrieve requests (without making a distinction between the store and retrieve requests). It has been shown that with a given stable rack capacity utilization and under steady state condition this rule is equivalent to the random assignment rule in which arriving items are assigned randomly to storage locations [1].

A more efficient assignment rule is the Turnover-Based assignment rule. The basic idea behind this rule is to assign items with shorter turnover times to the closer locations. Since these items are retrieved more frequently, the overall crane travel time could be reduced by implementing this assignment rule. This rule will require the allocation of each group of storage locations which are equidistance from the I/O point (in terms of time) to a group of items with a particular range of turnover time. Considering the random pattern of the arrival of items to the warehouse, in practice this assignment rule cannot be implemented. This is due to the fact that if such a policy is adopted many items will have to wait due to the unavailability of their corresponding locations while other rack locations are open waiting for their corresponding items.

A compromise between the random assignment and turnover-based assignment is the class-based assignment. In the class-based assignment rule, rather than allocating each equidistance group of rack locations to a narrow range of turnover times, the rack locations are divided into several classes with respect to the time distance from the I/O point. For instance in a two-class storage assignment system the locations are divided into two classes such that any location in class I is closer to the I/O point than any of those in class II. In the continuous case these classes can be shown by rectangles such as those shown in Figure 1. Similarly this can be defined for 3 or more class systems. The inventoried items are also divided into the same number of classes based on their turnover time. By storing items at locations in their corresponding classes the total crane time could be reduced compared to that of the random assignment while sufficient flexibility is provided for arriving items.

The assignment policy for the crane also affects the efficiency of the warehouse operations. The simplest policy is to have store and retrieve requests all wait in the same queue and have the crane serve this queue according to FIFO rule. That is, the crane takes the first request, say a store, starts from the I/O point, stores the item, returns to I/O point and then takes the next request if there is one.

An alternate crane utilization is with interleaving. In this case store and retrieve requests wait in separate queues. The crane selects a store and a retrieve request from their corresponding queues according to a given rule. Then from I/O point it travels to the store location and stores the item. From there it travels to the retrieve location, retrieves the requested item and then returns to the I/O point. Various rules can be provided for selection of the requests from queues. FIFO rule will select the first request from each queue. Another rule is to select the first request from one queue and then scan the other queue to find the request corresponding to the closest rack location with respect to the first request. The scan could include all members of the queue or just the first k members.

The studies conducted in [1] and [2] test several of the above policies with respect to the continuous model. Using deterministic models and under certain conditions they conclude that class-based storage assignment rule results much more efficient system output with respect to the crane time. In addition, they determine the optimal fractions of the storage spaces to include in each class for a two-class and three-class systems. They also show that interleaving cuts the crane time further more. In a subsequent paper [3] the above results are tested on a simulated model of the warehousing system of a discrete nature. Their conclusion is that the findings on the continuous model and under deterministic conditions is generally valid. However, considereing the highly stochastic nature of the system and the complexity associated with the discrete nature of the storage locations some discrepencies between the continuous deterministic model and more realistic ones are expected.

In this study a simulation model of the system is constructed such that it is not only able to test the effect of various assignment policies, it could also be used as a vehicle to optimize the system stochastically. The stochastic optimization will be used to determine the optimum boundaries of each class in the class-based storage assignment. For optimization purpose the principles of stochastic approximation are employed. These principles are built in a simulation optimization program which is capable of interacting with the simulation model and determining the optimum values of the decision variables through a recursive procedure.

FORMULATION OF THE PROBLEM

To formulate the problem as a simulation optimization one, the objective of the optimization and its relationship with the decision variables of the system have to be defined.

The Objective

Several objectives can be defined for this warehousing system. Perhaps the most significant objective is that of the efficient handling of the storage and retrieval requests. Efficiency could be defined in terms of shorter storage queues which translates into smaller counter or conveyor length for holding items arriving for storage. It could also be defined in terms of shorter retrieval queues which affects the delay of delivering retrieved items. Throughput (number of requests served during a given time) is another measure of efficiency of the operation.

One factor which directly or indirectly affects all above measures is the average service time required by the crane to serve a request. This is the average round trip time of the crane per a storage or a retrieval. For a given rate of arrival of the items for storage and a given distribution of length of stay of items on the storage racks, the length of the queues for the store and retrieve requests and the throughput are all functions of the service time of the crane. Due to these considerations the objective in this paper is to design a policy that minimizes the average round trip time/item for the crane.

Decision Variables

As indicated earlier the crane travel time is affected by two major types of assignment and priority policies. These are the pattern of assignment of storage spaces to items and the assignment of the crane to incoming requests. The storage spaces could be assigned randomly (COL Policy), which is the least favorable one, or be assigned according to classes. No particular decision variable is involved in the random assignment policy. Each arriving item is stored in the closest open location. Thus there is no ground for optimization.

For class based assignment, however, a decision must be made on the number of classes and the boundaries of each class. The variables involved in this case are of quantitative nature and can be incorporated into an optimization formulation. In this paper boundaries of classes are taken as decision variables. These are based on the length of stay (LOS) of the items in the warehouse. In terms of mathematical notations let x_i; i=1, 2,...k be the upper bound on the LOS of the items that are to be assigned to class i in a k-class system. That is, items with the LOS between 0 and x_1 are to be assigned to class 1, between x_1 and x_2 to class 2, and so on. Then the outcome of the system will be a function of $x_1, x_2, \ldots x_k$ and k.

The policies regarding the assignment of the crane to items affect the system in a different manner. The crane could be assigned to the combined queue of the store and retrieve requests to serve them according to FIFO rule. Alternatively in the case of the shortage of waiting space for items coming for storage the priority could be given to store requests. It is also possible to give priority to the retrieve requests in order to minmize the delay in service. There is also the possibility of adopting interleaving.

Since the crane assignment policy affects the system structurally rather than parametrically, the optimization in terms of decision variables has to be done separately for each individual crane assignment policy. In this paper two policies of with and without interleaving will be studied. In the no interleaving case the store and retrieve requests are all put in he same queue and are served based on FIFO policy. In the model with interleaving store and retrieve requests wait in separate queues. If there are items in both queues, the first item in the store queue is selected. Then the retrieve queue is scanned for a request most suitable to be combined with the store request. If only one queue contains requests, that queue will be served by FIFO policy.

The Objective Function

The objective function for this problem can be implicitly expressed as

$$Y = f(x_1, x_2, \ldots x_k, k)$$

where Y is the average round trip time for the crane, $x_1, x_2, \ldots x_k$ are the boundaries of classes in terms of LOS of items and k is the number of classes.

Obviously if f(.) could be defined as an analytical function of the decision variables the problem would have consisted of a classic minimization of a mixed integer problem. However, this is not the case and there are two major complexities involved. First, the relationship between the crane time and those of the decision variables is so complex that derivation of a mathematical function that represents the real system is not always possible. Of course some functions can be derived by approximation and making some assumptions. For instance Hausman, W.H. et al [1] have derived such a function for continuous rack space and some additional assumptions. However, as it will be shown later the results obtained under some of those assumptions may not be valid for some other situations.

In the absence of an analytical relationship between Y and the decision variables, the simulation is the best technique for evaluating the response of the system for any given levels of the decision variables. Being able to evaluate the response of the system for any given $x_1, x_2, \ldots x_k$ and k opens the possibility of using search methods to determine the optimal values of those variables. However, there is yet a major characteristic of the response obtained from simulation which creates a problem for optimization. This is the second complexity associated with such systems as is explained below.

This complexity is due to the stochastic nature of such a system. Since the arrival patterns and the length of stay in the warehouse for items are random, the result of evaluating the average crane time for a given policy is also random. That means the objective function is a stochastic function of the decision variables. In other words, for a particular policy the value of the objective function varies from one observation to the next. This fact affects the ability to optimize the system. The reason is that in order to apply a search method the values of the objective function at various points have to be compared with each other. However, since the value of the objective function is always accompanied with error, the results of comparisons of points cannot always be relied upon. This in turn may cause the search method to converge to a wrong point.

One remedy often used for stochastic objective functions is to make several observations at each point (several simulation runs with different random numbers) and use the average as a deterministic value. This may or may not alleviate the problem. It will not if the variation is too much, because the average of several observations would still contain too much error. In addition, if the simulation model is large it will cost too much to make several observations at each point.

Stochastic approximation is a method which, if applied properly, alleviates both of the above problems to some extent. Before describing the simulation model employed for this problem and presenting the results, a brief explanation of the stochastic approximation method and the algorithm used in this paper is given in the following section.

APPLICATION OF STOCHASTIC APPROXIMATION METHOD TO SIMULATION OPTIMIZATION

Let us show the response of interest from a simulation model by Z. Since this response is stochastic, Z will be a random variable. Obviously the response of the simulation model is a function of the structural behavior and the levels of the decision variables of the system. Thus, for a particular structure of a simulation model Z can be defined as

$$Z = f(x_1, x_2 \ldots x_m) = f(\underline{X}) \qquad (2)$$

where $x_1, x_2 \ldots x_m$ are the decision variables of the system and $\underline{X}$ is an m-dimensional vector whose components are $x_1, x_2 \ldots x_m$.

The fact that Z is a stochastic (and not a deterministic) function of $\underline{X}$ prompts two considerations in its optimization. First, for a given $\underline{X}$ the response of each evaluation of the system (each simulation run) Z will be different from the next (using a different random number stream). As was mentioned earlier, this creates a problem for application of a search method for optimization.

The second consideration involves defining the optimization for this case. Optimization is usually defined as the procedure of obtaining optimal values for components of $\underline{X}$ such that the response is either minimized or maximized. However, in this case, the response of the system for any given $\underline{X}$ is a random variable and minimization or maximization cannot be applied to random variables. Thus one has to pick a particular characteristic or a parameter of the random variable as the subject of optimization. This parameter could be the mean, variance or any other parameter of the random variable. Often optimization is aimed at the true value of the response which for most practical problems is the theoretical mean of the random variable representing the response. For instance in the present study we look for the optimum values of the boundaries for classes such that the mean of the crane time per item is minimized.

In order to search for the optimum mean through a search method, at each point the mean of the response has to be evaluated. When the means of evaluation is simulation, this means a large umber of replications or a long simulation run at each point. However, even these may not yield a good estimte of the mean.

Stochastic approximation method (SAM) is a mathematical technique which provides a reasonable solution to the above problem. Let

$$Y(\underline{X}) = E[(Z\underline{X})] \qquad (3)$$

be the regression function of Z and let $\underline{X}°$ be the value of vector $\underline{X}$ for which $Y(\underline{X})$ is optimum. Then

$$Y(\underline{X}^0) = E[Z|\underline{X}^0] = \mu_{Z|\underline{X}^0} \qquad (4)$$

will indicate that $\underline{X}°$ is the optimum value of $\underline{X}$ for which the mean of random variable $Z(\underline{X})$ is better than its mean at any other point.Thus $\underline{X}°$ is the solution for the stochastic optimization. However, finding $\underline{X}°$ depends on the availability of the theoretical regression function Y(X) which is never available. The most one could do is to estimate Y(X) through regression analysis. The estimation of $Y(\underline{X})$ is not feasible due to its extremely high cost if a reasonable accuracy is to be obtained. Stochastic approximation method is a technique by which the optimum value of $Y(\underline{X})$ can be approached using just one observation of Z at each point and without the need of estimating $Y(\underline{X})$.

According to one version of SAM if a_n and c_n are two series of numbers such that

$$\sum_{n=1}^{\infty} a_n = \infty$$

$$\sum_{n=1}^{\infty} a_n c_n < \infty \qquad (5)$$

$$\text{and} \quad \sum_{n=1}^{\infty} (a_n/c_n)^2 < \infty$$

then $\underline{X}_n$ in the recursive formula

$$\underline{X}_{n+1} = \underline{X}_n + (a_n/c_n)\,\underline{T}(\underline{X}_n, c_n) \qquad (6)$$

approches $\underline{X}°$ when $n \to \infty$. Vector $\underline{T}(\underline{X}, c_n)$ above is obtained using responses of the system where its i^{th} component is

$$t_i = Z(\underline{X}_n + c_n \underline{u}_i) - Z(\underline{X}_n) \qquad i = 1,2 \ldots m \qquad (7)$$

$\underline{u}_i$ is the unit vector of the i^{th} coordinate.

There are a few not so restricting conditions that have to exist for the above procedure to converge. Three main conditions are:

- $x_1, x_2 \ldots x_m$ must be continuous variables

- The variance of $Z(\underline{X})$ must be finite for all values of $\underline{X}$.

- $Y(\underline{X})$ must be a unimodal function.

The last condition is of course somewhat restrictive, but since this restriction is present for almost all search methods, the penalty associated with using SAM is not more than any other one. In fact, if the function is not unimodal, the optimum found would be a local rather than the global one.

The main restriction of SAM is the requirement of n to approach infinity. However, for some sacrifice in accuracy, this procedure has been modified in Azadivar [4] and Azadivar and Talavage [5] to approach to the optimum within a reasonable number of simulation runs. The algorithm developed is called SAMOPT. SAMOPT is an optimization package that can be interfaced with any simulation program. Based on the input data which consists of upper and lower boundaries on variables, the number of simulation runs allowed and the criterion for stopping, SAMOPT supplies a suitable starting point for the start of the search. Then applying a modified version of recursive formula (6) the search continues until the optimum is reached within the prespecified precision limits. The program is also capable of taking into account a set of linear constraints on the decision variables. Further description of SAMOPT and the results of tests performed on it can be found in [4] and [5].

THE SIMULATION MODEL

Assumptions

The simulation model constructed and analyzed in this paper employs basically the same assumptions made in [1], [2] and [3] with a few variations in some of them. The main assumptions are

- The demand for items is based on ABC rule with 20/80 ratio.
- The transfer of all pallets containing items from the conveyor to the stacking crane and vice versa takes place at a single point located at the bottom left corner of racks (I/O point).
- All rack locations are identical.
- The time to reach a particular location in the rack system is equal to the maximum of the horizontal or vertical movement time taken by the crane.
- Only one item is assigned to each pallet and crane can carry only one pallet at a time.
- In models without interleaving store and retrieve requests are put in the same queue and are served according to FIFO policy. In models with interleaving there are separate queues for store and retrieve requests. When both queues contain erquests, the first unit in the store queue is selected. Then all the requests in the retrieve queue are scanned to find the one belonging to the closest rack location to this store request.

Numerical Values

A warehousing system with the following numerical values for parameters is considered. The rack consists of 20 rows and 20 columns with an overall of 400 locations. Interarrival time for incoming items is exponentially distributed with an average of 1.5 time units. The average length of stay for items is 540 time units. This will result in a theoretical rack utilization factor of 90%. The theoretical rack utilization is obtained from

$$U_R = \lambda \mu/N \qquad (8)$$

where λ is the average umber of items arriving for storage per unit time (1/1.5 in this case), μ is the average length of stay (540 here) and N is the total number of available rack spaces (400).

The crane is assumed to move from one rack space to the next in .05 time unit in both horizontal and vertical directions. Thus, the time it takes to move from a location at row r_1 and column c_1 to a location at row r_2 and column c_2 is obtained from

$$t = .05\ \mathrm{Max}(|r_1 - r_2|, |c_1 - c_2|) \qquad (9)$$

The coordinates at the I/O point are both zero. Thus, the maximum one way travel time for the crane is one time unit (.05 x 20).

In class based storage space assignment the items are classified based on their length of stay. Let x_1, $x_2 \ldots x_k$ be the upper boundaries of classes 1 through k. Since the upper boundary of the last class coincides with maximum length of stay the number of variables to be determined is always k-1. That is, for a 2-class system there is only one variable to be specified. For 3-class system x_1 and x_2 have to be specified.

The assignment of the storage spaces to each class is based on the average rack utilization. The storage locations are assigned to each class such that each class will have approximately the same rack utilization as the entire system. This is done as follows. Let x_1 be the upper boundary of the length of stay for items in class 1. The total number of storage locations required to hold these items can be obtained from

$$M_1 = \lambda \int_0^{x_1} x f(x) dx \qquad (10)$$

where λ is the average arrival rate and f(x) is the density function for the length of stay of items. As stated earlier

$$F(x) = \left(\frac{zx}{\mu}\right)^{\frac{z}{1-z}} \qquad 0 < x < \frac{\mu}{z} \qquad (11)$$

where z = 2s/(s+1) and μ is the overall average length of stay [see (3)]. s is a factor which depends on the type of ABC curve for the demand. For our 20/80 system s=.139. From (11)

$$f(x) = \frac{z^2}{\mu(1-z)} \left(\frac{zx}{\mu}\right)^{\frac{2z-1}{1-z}} \qquad (12)$$

The total number of storage locations needed for all items is

$$M = \lambda \int_0^{\frac{\mu}{z}} x f(x) dx = \lambda\mu \qquad (13)$$

The fraction of the storage spaces to allocate to the first class is then given by M_1/M. Then the number of storage locations to allocate to class 1 out of a total of N rack locations is

$$N1 = \frac{M_1}{M} N = \frac{\lambda \int_0^{x_1} x f(x) dx}{\lambda\mu} N$$

or

$$N_1 = \frac{N}{\mu} \int_0^{x_1} x f(x) dx = N\left(\frac{zx_1}{\mu}\right)^{\frac{1}{1-z}} \qquad (14)$$

This would require the allocation of the N_1 closest locations to the first class. Since the rack is square in time, these locations are in a square whose one corner is I/O point and has sides consisting of $1/\sqrt{N_1}$ rows and the same number of columns.

Similarly for class 2 the number will be

$$N_2 = \frac{N}{\mu} \int_{x_1}^{x_2} x f(x) dx = N[(\frac{zx_2}{\mu})^{\frac{1}{1-z}} - (\frac{zx_1}{\mu})^{\frac{1}{1-z}}] \qquad (15)$$

and the locations will consist of those between a square of size $\sqrt{N_1+N_2}$ and another of size $\sqrt{N_1}$. Since the storage spaces are discrete, the results are rounded to the closest integer. If there are more classes, the corresponding rack spaces can be allocated by extending the above procedure.

ANALYSIS AND OPTIMIZATION OF THE SIMULATION MODEL

Four different structures for the simulation model are analyzed. These are 2-class system without interleaving, 2-class system with interleaving, 3-class system without interleaving and 3-class system with interleaving.

In 2-class systems the variable to be optimized is the upper boundary of the length of stay of the items to be stored in the first class. Note that the time (rather than the number of rack locations) was selected as the decision variable in order to comply with the continuous variable requerements of SAM. For 3-class systems 2 variables of x_1 and x_2 corresponding to the upper bounds on the length of stay for items to be included in class 1 and 2 respectively, were considered. In addition the constraint $x_2 > x_1$ is added to assure the feasibility of the solutions.

For 2-class systems the upper and lower boundaries of $\frac{\mu}{z}$ and 0 were specified on the variables. 70 simulation runs were allowed. (SAMOPT algorithm requires this information.) The same range was supplied for x_1 and x_2 for 3-class systems, but the number of allowed runs were increased to 100.

The system was simulated by SLAM simulation language [6]. For each run, after allowing the system to reach its steady state the statistics were cleared. Then the statistics were collected for 2000 time units. This interval corresponds to arrival of about 2000/1.5=1,333 items. Since the items that are stored must eventually be retrieved, in steady state, this time interval corresponds to about 2,667 observations of the crane time. In practice the number was slightly less because some of the items waiting for storage in the store queue, were rertieved before being stored.

The interface between SAMOPT and the simulation model takes place as follows. Based on initial information SAMOPT performs a design of experiment calling the simulation program for evaluation of various points. Through this a suitable starting point is obtained. This starting point is used as the first value in the stochastic approximation recursive formula (n=1). Then whenever the value of Z is required SAMOPT supplies the values of the decision variables and calls the simulation program. This continues until the allowable runs are exhausted or the desired precision limit is reached. The flow chart for this procedure is shown in Figure 2.

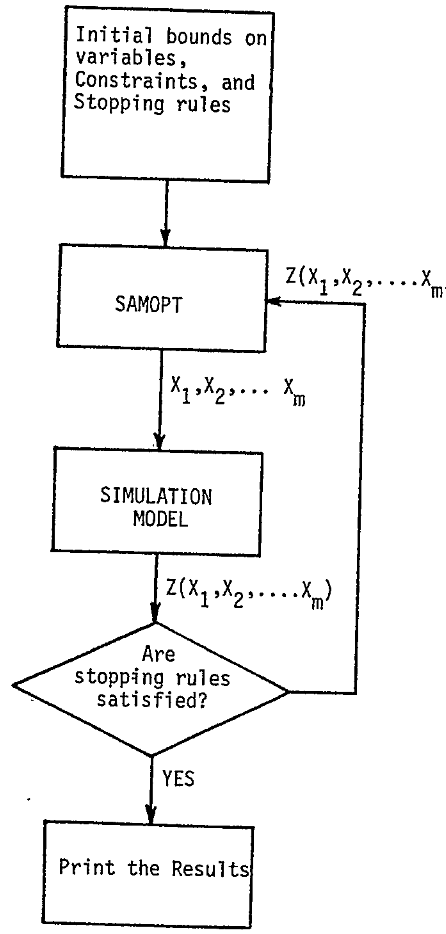

Figure 2: Flow Chart of the Interaction between SAMOPT and The Simulation Model.

Results of Optimization

The results of optimization are shown in Table 1. In this table the response of the discrete model for the optimum values of decision variables obtained from the continuous model are also shown. The x values for continuous model are obtained by using the ratios given in [2] in conjunction with the density function of the length of stay. The table shows the optimum values of the decision variables and the average time for crane in each case. The average crane time is the average of 50 runs each for 2000 time units. Also given in the table is the estimated standard deviation of these averages. The standard deviation is used for testing the difference in the mean response of the system for alternative values of the decision variables.

	Responses Using Optimums Found By SAMOPT				Responses Using Optimums Found By Contiuous Model			
	X_1	X_2	$\bar{Z}$ *	$s_{\bar{Z}}$ **	X_1	X_2	$\bar{Z}$ *	$s_{\bar{Z}}$ **
Random Assignment Without Interleaving			1.2964	0.0033			1.2964	0.0033
Random Assignment With Interleaving			0.8943	0.0018			0.8943	0.0018
2-Class Assignment Without Interleaving	478.50		0.9235	0.0036	331.65		0.9531	0.0045
2-Class Assignment With Interleaving	440.80		0.7421	0.0015	331.65		0.7445	0.0017
3-Class Assignment Without Interleaving	124.50	591.00	0.7595	0.0028	44.07	661.58	0.7874	0.0025
3-Class Assignment With Interleaving	179.60	1320.00	0.7034	0.0015	44.07	661.58	0.6981	0.0019

* $\bar{Z}$ is the average of 50 replications.

** $s_{\bar{Z}}$ is the estimated standard deviation of the $\bar{Z}$ based on the same 50 runs.

Table 1: Results of Optimization

The table generally agrees with the results obtained in [1], [2] and [3]. The random assignment without interleaving is the worst case, while the 3-class system with interleaving provides the lowest crane time. However, the results of the optimum found for class boundaries are somewhat different. This difference is especially significant in 2-class and 3-class systems without interleaving. As it is seen, the results obtained by SAMOPT are significantly better than those obtained by optimizing the continuous model. A test of significance at 95% level indicates this difference.

The difference between the results for models with interleaving is not significant. This was, in a sense, expected, because in systems with interleaving the boundaries should not necessarily be similar to those without interleaving. In other words, square shaped classes are not necessarily the optimum arrangement for systems with interleaving. As a result the response of the system is less sensitive to the boundaries specified in terms of squares. In fact, test runs indicated the flatness of the response surface of such models with respect to a considerable portion of the feasible region.

CONCLUSION

This study shows the potentials of optimizing real world systems through simulating them and using the simulation model for search purposes. Mathematical models provide a useful general insight into the problems, and are more suitable for application of optimization routines. However, as it is shown here, when systems become stochastic and more complex better results could be obtained by applying optimization to responses obtained from the simulation model of the system. With regard to the warehousing model, in addition to the cases considered in this paper, there are a lot of other situations that could be modeled and optimized. For instance, possibilities of other shaped division of rack spaces (other than squares) would be of interest, especially when interleaving is adopted. The dimensions of these shapes would be suitable dicision variables for optimization. Also other structural changes, such as carrying more than one pallet at a time, can be tested and corresponding variables optimized through simulation optimization. Some of these possibilities are currently under study by the author.

REFERENCES

1. Hausman, W.H., Schwarz, L.B. and Graves, S.C., "Optimal Storage Assignment in Automatic Warehousing Systems," Management Science, Vol. 22, No. 5, pp.629-638, February 1976.

2. Graves, S.C., Hausman, W.H. and Schwarz, L.B., "Storage-Retrieval Interleaving in Automatic Warehousing Systems, " Management Science, Vol. 23, No. 9, pp.935-947, May 1977.

3. Schwarz, L.B., Graves, S.C., and Hausman, W.H., "Scheduling Policies for Automatic Warehousing Systems: Simulation Results," AIIE Transactions, Vol. 10, No. 3, pp.260-270, September 1978.

4. Azadivar, F. "Optimization of Stochastic Systems Through Simulation using Stochastic Approximation Method", Ph.D. Thesis, Purdue University, December 1979.

5. Azadivar, F. and Talavage J.J., "Optimization of

Stochastic Simulation Models", Mathematics and Computers In Simulation, Vol. 22, No. 3, pp. 231-241, September, 1980.

6. Prisker, A.B., and Pegden, C.D., Introduction to Simulation and SLAM, John Wiley, 1979.