Available online at www.sciencedirect.com

ScienceDirect

Computers & Industrial Engineering 54 (2008) 783–799

computers & industrial engineering

www.elsevier.com/locate/caie

# A particle swarm optimization algorithm for the multiple-level warehouse layout design problem

Semih Önüt [a,*], Umut R. Tuzkaya [a], Bilgehan Doğaç [b]

[a] Department of Industrial Engineering, Mechanical Faculty, Yildiz Technical University, 34349 Istanbul, Turkey
[b] Lojitek Logistics Technologies and Consultancy Company, Istanbul, Turkey

Received 25 January 2007; received in revised form 24 September 2007; accepted 26 October 2007
Available online 4 November 2007

**Abstract**

Warehouse operation and management is one of the essential parts of manufacturing and service operations. The warehouse layout problem is a key to warehouse operations. Generally, warehouse layout design models attempt to optimize different objectives such as the orientation of storage racks, the allocation of space among competing uses, the number of cranes, the overall configuration of the facility, etc. The warehousing strategies can be classified as distribution-type, production-type and contract-type warehouse strategies. In this study, a distribution-type warehouse considered that various type products are collected from different suppliers for storing in the warehouse for a determined period and for delivery to different customers. The aim of the study is to design a multiple-level warehouse shelf configuration which minimizes the annual carrying costs. The turnover rates of the products are classified and they are considered while putting/picking them to/from shelves regarding the distances between the shelves and docks. Since proposed mathematical model was shown to be NP-hard, a particle swarm optimization algorithm (PSO) as a novel heuristic was developed for determining the optimal layout.


*Keywords:* Warehouse layout; Order-picking; Class-based storage; Particle swarm optimization

## 1. Introduction

A supply chain can be considered as a network of individual entities that are collectively responsible for procurement, manufacturing, warehousing and transportation activities. Performance of any entity in the network depends on the performance of others. The efficiency and effectiveness in any supply network in turn is highly determined by the operation of the nodes in such a network, like the warehouses (Jayaraman & Ross, 2003). Warehouses provide an important connection among suppliers, manufacturers, distributors and customers in the supply network. Warehousing involves all movement of goods within warehouses or distribution centers consisting of receiving, storage, order picking, accumulation, sorting and shipping (Van den Berg, 1999). Planning, design and control of warehousing systems are complex issues. It includes a large number

* Corresponding author. Tel.: +90 212 259 70 70x2823; fax: +90 212 258 59 28.
*E-mail address:* onut@yildiz.edu.tr (S. Önüt).

0360-8352/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2007.10.012

of interrelated decisions from a functional description through a technical specification, to equipment selection and layout determination. Warehouse operation and management is also one of the essential parts of a supply network. The warehouse layout design problem is a key component of warehousing operations.

Essentially warehousing strategies are classified as distribution-type, production-type and contract-type warehouses (Van den Berg & Zijm, 1999). Distribution-type warehouses collect different type products from different suppliers for delivery to different customers. Production-type warehouse is used for storage of the products having different kinds of characteristics in a production facility. Contract-type warehouse executes warehousing activities for different customers. In this study, a distribution-type warehouse is considered as the suitable warehousing strategy.

Warehousing processes may be divided into different phases. These are mainly receiving, storage, order picking and shipping phases (Rouwenhorst et al., 2000; Van den Berg & Zijm, 1999). In the receiving phase as the first process, goods are delivered generally by trucks, unloaded at the receiving area, checked and prepared for transportation to the storage area. The goods are transported to a storage location in the storage area as the storage phase. A storage area can be divided into the following areas: reserve storage area and forward area. In the reserve area goods are stored till they are required for shipment to the customer. In the forward area goods are stored and prepared for the retrieving operations carried out by the order pickers. As the third phase, order picking refers to the retrieval of goods from their storage locations whenever a good is requested. These operations executed by the order pickers. At the shipping area orders are checked, sorted, packed and loaded in trucks as the last phase. A cross-docking process can also be used as a warehousing process. Cross-docking refers to the process in which goods, cartons or pallet loads are taken directly from the receiving trucks to the shipping trucks. It provides rapid consolidation and processing (Chen, Guo, Lim, & Rodrigues, 2006; Heragu, Du, Mantel, & Schuur, 2005).

There are a number of studies related with the design and integration of the different warehousing processes. Also different design and integration models have been developed in the literature including mathematical models and heuristic procedures. A genetic algorithm with a special crossover operator was proposed by Poulos, Rigatos, Tzafestas, and Koukos (2001), who find the Pareto-optimal solutions in the warehouse replenishment problem. Lee and Elsayed (2005) considered the storage sizing problem for generic warehouses under a dedicated storage policy. Since the problem was formulated as a nonlinear optimization model, an iterative search procedure was developed to solve the model optimally. There are also some studies classified as the single- or multiple-level warehouse layout design models mentioned below. Lai, Xue, and Zhang (2002) considered a paper reel layout problem formulating by integer programming, which is shown to be NP-hard. Following a natural decomposition of the problem, they proposed a two-stage heuristic procedure including an optimal method and a simulated annealing heuristic. Zhang, Xue, and Lai (2002) investigated a multiple-level warehouse layout problem considering multiple storage areas in different levels of a warehouse. An integer programming model was proposed and due to the similarity of the Lai et al. (2002)'s NP-hard problem, a class of genetic algorithm-based heuristic was developed. Zhang and Lai (2006) considered a multiple-level warehouse layout problem again and proposed a class of new heuristics by combining a genetic algorithm and path linking strategy to solve the problem. Heragu et al. (2005) presented a mathematical model and a heuristic algorithm that jointly determine product allocation to the functional areas in the warehouse.

As mentioned before, order picking refers to the retrieval of goods from their storage locations on the basis of customer orders and one of the most important activities in a warehouse. This process is also the most laborious and time-consuming process of all warehousing processes. Hence modeling a warehouse with the order picking locations and considering all order pickers in the same model is very useful to determine good order picker routes and to reduce travel times simultaneously. Additionally batching several orders in a single order picking route can be taken into account to reduce travel times. Various methods have been evaluated for order picking processes in the literature (Chen & Wu, 2005; De Koster, Van der Poort, & Wolters, 1999; Hsu, Chen, & Chen, 2005). De Koster and Van der Poort (1998) studied the problem of finding efficient order picking routes for both conventional and modern warehouses. They used extended polynomial algorithm to find order picking routes with a minimal length and then compared the performance of this new algorithm with the well-known S-shape heuristic solution. In another work (Chew & Tang, 1999) a travel time model with general item location assignment in a warehouse system was developed to analyze order batching and storage allocation strategies in an order picking system. Roodbergen and De Koster (2001a) presented a dynamic programming-based routing algorithm to determine the shortest order picking tours for a warehouse where aisle changing is possible at the front, the rear and in the middle of the warehouse. The same authors also described a number of extended heuristics to evaluate order picking routes in a warehouse with two or more cross aisles and they presented a new routing heuristic, called the combined heuristic, based on dynamic programming again. To analyze the performance of the heuristics, a branch-and-bound algorithm that generates shortest order picking routes was used (Roodbergen & De Koster, 2001b). Kim, Graves, Heragu, and Onge (2002) solved an industrial warehouse order picking problem using an intelligent agent-based model. In the problem, goods are stored at multiple locations and the pick location of goods can be selected dynamically. Same authors presented a hybrid intelligent agent-based scheduling and control system architecture for the same problem mentioned above in another survey. They also developed a mathematical model and a genetic algorithm-based heuristic for the resource assignment problem (Kim, Heragu, Graves, & Onge, 2003). Jewkes, Lee, and Vickson (2004) considered a multiple-picker order picking line which stores $nk$ types of products in $n$ bins, each with $k$ shelves and determined optimal policy for the problem of product location, picker home base location and allocating products to each picker for minimizing expected order cycle time using a dynamic programming algorithm. Another survey (Hsieh & Tsai, 2006) presented the effects on the order picking system performance for factors, such as quantity and layout type of cross aisles in a warehouse system, storage assignment policy, picking route, average picking density inside an aisle, order combination type, etc. using a simulation and analysis tool.

The performance and efficiency of an order picking system depend on the demand pattern of the items, the layout of the warehouse, storage process, batching method and routing method (Petersen, 1999). Storage process can be performed by different storage policies. The most used and preferred policies can be given as randomized storage policy, dedicated storage policy and class-based storage policy. The randomized storage policy is performed by the allocation of the storage location based on the available space at the time of the storage job. Storage decision is left to the operator in another word. A dedicated storage policy determines a particular predetermined location for each product to be stored. A class-based storage policy is a common and shared policy between randomized and dedicated policies. It divides goods into classes based on some criteria and each class is assigned a block of storage locations. This policy can be called as ABC zoning (Hausman, Schwarz, & Graves, 1976; Lai et al., 2002; Larson, March, & Kusiak, 1997; Rosenblatt & Eynan, 1989; Rouwenhorst et al., 2000).

As seen above, the literature on warehouse design and operations is highly extensive. Much of the research in this area has been motivated by the need to improve the efficiency of order picking operation and to estimate travel distance or time. For these reasons order picking process, storage strategy, batching method and routing method have received considerable attention in the warehousing literature. However warehouse layout design, especially in multiple-level, has received less attention. There are also a limited number of studies concerned with the determination of the number of door in the literature. Additionally a multiple-level warehouse layout design problem regarding the receiving and shipping processes, order picking process, storage strategy and optimal number of door has not taken into account by any author. This is the most powerful motivation to consider this problem. Furthermore, a large number of mathematical models and heuristic procedures have been proposed for warehouse layout design problems and different warehousing processes.

Warehouse design problem can also be considered as facility layout problem. Especially, the problem type that assigns the facilities to locations by considering capacity limitations, priorities of facilities, etc. is difficult to solve. There is no algorithm that solves this kind of problems in polynomial time even small ones (Kusiak & Heragu, 1987). Therefore these problems, like assigning the goods to shelves in warehouses, belong to NP-hard problems class. The constituted model in this study has similarities with Zhang and Lai (2006) in terms of the difficulty of solving multi-level warehouse problem and NP-hardness. For detailed information Garey and Jhonson (1979) can be seen.

In this paper, we present a mathematical model for determining a multi-level warehouse layout. We also focus on locating the items in the warehouse (storage policy), picking process from the pick location (order picking) and specifying the optimal number of dock. The employed storage policy is the class-based storage strategy. It is widely used in practice because it often leads to a substantial reduction in order pick travel distance. Essentially a distribution type warehouse considered that various type products are collected from different suppliers for storing in the warehouse for a determined period and for delivery to different customers.

The turnover rates of the products are classified and they are also considered while putting/picking them to/from shelves regarding the distances between the shelves and docks. Since proposed mathematical model was shown to be NP-hard because of the above mentioned reasons and dealt with nonlinear components in the objective function and constraints, a particle swarm optimization (PSO) algorithm as a novel heuristic was developed for determining the optimal layout. The rest of the paper arranged as follows: Section 2 formulates the addressed multiple-level warehouse layout problem. In Section 3, PSO algorithm is introduced by giving the original PSO algorithm and improvements related with dealing the constrained problems. Section 4 presents the proposed PSO algorithm for tackling the problem. The computational results and some scenario analysis are given in Sections 5 and 6, respectively. Finally, the paper is concluded in Section 7.

## 2. The formulation of multiple-level warehouse layout design model

In this study, a warehouse layout problem that tries to minimize material handling cost was considered. Bassan, Roll, and Rosenblatt (1980) were examined the same problem by offering a comparison method for two alternative shelf arrangement for a rectangular warehouse. They were considered homogeneous items that have the equal probabilities for picking and putting away in a two-dimensional warehouse. Here, we extended their study for heterogeneous items stocked multiple-level warehouse.

The yearly throughput of the warehouse is classified into three groups including A, B and C, according to their turnovers. Considering the turnover rates and stocking periods of the items, probabilities of (picking or putting) the orders belonging in A, B and C classes are $P_A$, $P_B$ and $P_C$, respectively. The main reason of this classification is to locate the items in the warehouse basing on the closeness to the dock. The total capacity and yearly throughput of the warehouse, total storage spaces for each item class and the lengths of the aisles and shelves are entered to the model as the pre-determined parameters. The notation of the parameters and variables are shown in Table 1. Also, Fig. 1 is given to visualize some of the dimensions given in Table 1.

By using given details, a mathematical model providing the optimal number of storage spaces along a shelf and the optimum number of shelves is constituted. In other words, the three dimensions of the warehouse namely, length ($u$), width ($v$) and height ($h$) are obtained by the model solution. The objective function of the model is constituted from the average travel distances in three dimensions and the unit material handling cost. Before giving the objective function and constraint formulations, some points should be clarified. The dock of the warehouse is located at the center of the horizontal wall. In the other word, the distance between the dock and the left vertical wall is $u/2$. Also, the probability of carrying an item to right or left side of the dock is equal. Therefore, the average travel distance in the horizontal axis is given in Eq. (1).

Table 1
Nomenclatures

| | |
|---|---|
| $w$ | Width of the double shelf |
| $L$ | Length of a storage space (pallet) |
| $m$ | Number of the total storage spaces along a shelf |
| $m_i$ | Number of the storage spaces allocated to class $i$ items |
| $h$ | Number of the storage levels in the height directions |
| $n$ | Number of the double shelves |
| $K$ | Total warehouse capacity in the storage spaces |
| $a$ | Width of an aisle |
| $u$ | Length of the whole warehouse |
| $v$ | Width of the whole warehouse |
| $d$ | Yearly throughput of the warehouse, in storage units |
| $P_i$ | Probability of an order belonging to class $i$ items |
| $N_i$ | Total number of the storage spaces of $i$ type items |
| $C_h$ | Material handling cost of moving an item in unit length |
| $T_v$ | Average travel distance in vertical axis |
| $T_u$ | Average travel distance in horizontal axis |
| $T_h$ | Average travel distance in height axis |

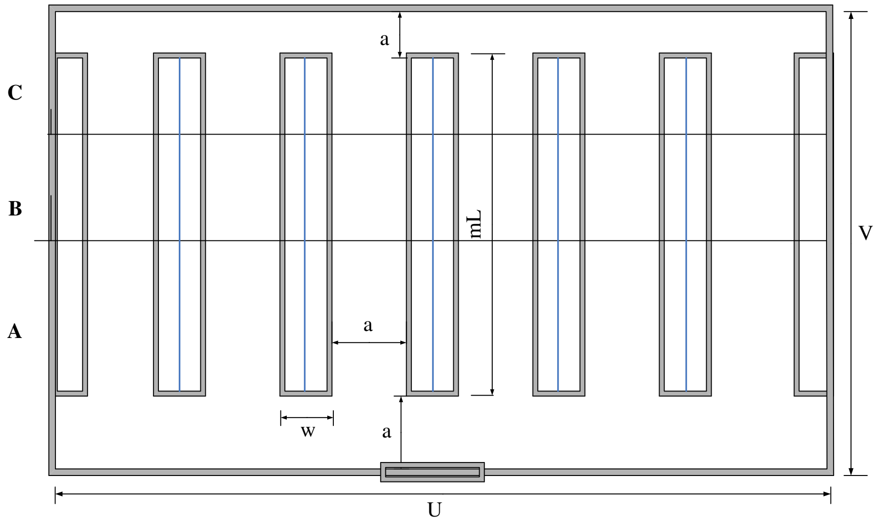


Fig. 1. A cross-section of the warehouse denoting the parameters.

$$T_u = \frac{u}{4} \tag{1}$$

The average vertical travel distances depend on the probability of the order that belonging the A, B and C classes. In such a case, the formulation of the average travel distance in the vertical axis can be given in the following equation:

$$T_v = a + P_a\left(\frac{m_aL}{2}\right) + P_b\left(m_aL + \frac{m_bL}{2}\right) + P_c\left(m_aL + m_bL + \frac{m_cL}{2}\right) \tag{2}$$

After some simplifications, Eq. (2) can be expressed as follows:

$$T_v = \left[a + \frac{m}{K}L\left[(N_A + N_B)\left(1 - \frac{P_a}{2}\right) + \frac{P_c}{2}(N_B + N_C) - \frac{N_B}{2}\right]\right. \tag{3}$$

The average travel distance in the last axis height can be given in the following equation:

$$T_h = \frac{h}{2} \tag{4}$$

Multiplying the total travel distances in all dimensions by the unit material handling cost and by the yearly throughput of the warehouse, the total cost minimizing objective function will constitute as shown in Eq. (5)

$$C = 4dC_h\left[a + \frac{m}{K}L\left[(N_\mathrm{A} + N_\mathrm{B})\left(1 - \frac{P_a}{2}\right) + \frac{P_c}{2}(N_\mathrm{B} + N_\mathrm{C}) - \frac{N_\mathrm{B}}{2}\right] + \frac{u}{4} + \frac{h}{2}\right] \tag{5}$$

The part of the Eq. (5), which is given in the parenthesis, represents a simplified version of the summation of the average travel distances in all dimensions. The carrier starts the carrying process of an item at the dock. It carries the item to the shelf and return to the dock. It means that the distance is passed two times. At the end of the storage time, the same item is picked back to the dock and this means two more passing the same distance. This situation is represented by adding a leading coefficient 4 in Eq. (5).

The constraints of the model are given in Eq. (6). The first constraint of the model satisfies the yearly demand and the others provide designing a warehouse in three dimensions. The details of the calculation of the total moving distance for picking and putting away the items through a year can be found in Bassan et al. (1980), but for only one type product and a two-dimensional layout.

$$
\begin{aligned}
&K - 2mnh = 0 \\
&m \geqslant 1 \\
&n \geqslant 1 \\
&h \geqslant 1 \\
&u = n(w + a)
\end{aligned}
\tag{6}
$$

Here, the decision variables are $m$, $n$ and $h$. $n$ shows the required number of the shelves in the warehouse ($u = n(w + a)$). $m$ shows the number of the storage spaces in a shelf ($v = 2a + mL$). $h$ shows the number of storage levels in the height directions. As a result, the model tries to determine the optimal number, length and height of the shelves. The assignments of the items to the shelves are realized by considering the order probabilities of the item classes and the quantities of the separated parts of the shelves to the item class.

## 3. Particle swarm optimization

Particle swarm optimization (PSO) is a stochastic optimization technique and also a population based search algorithm first proposed by Eberhart and Kennedy (1995) and Kennedy and Eberhart (1995), inspired by social behavior of bird flocking or fish schooling. PSO is a meta-heuristic approach used for solving hard global optimization problems. Commonly used meta-heuristics can be briefly summarized as ant colony optimization, genetic algorithms, artificial neural networks, evolutionary algorithms and simulated annealing (Eussuf, Lansey, & Pasha, 2006). PSO is one of the modern meta-heuristic algorithms under the evolutionary algorithms. Evolutionary algorithms, like genetic algorithm, and evolutionary programming strategies are search algorithms based on the simulated evolutionary process of natural selection, variation and genetics. PSO has been defined as an evolutionary computation algorithm and has typical features of both genetic algorithms and evolution programming strategies. Evolutionary algorithms can also provide a near global solution (Arumugam & Rao, 2005; Kennedy & Eberhart, 1995). PSO combines local search and global search and ensures high efficiency. It has a more global searching ability at the beginning of the run and a local search near the end of the run. Therefore, while solving problems with more local optima, there are more possibilities for the PSO to explore local optima at the end of the run (Naka, Genji, Yura, & Fukuyama, 2003; Shi & Eberhart, 1999). PSO has some common characteristics with genetic algorithm, such as starting with a group of randomly generated population, having fitness values to evaluate the population, updating the population and searching for the optimum with random techniques. But PSO does not have genetic operators. Particles update themselves and they also have own memory (Haq, Sivakumar, Saravanan, & Karthikeyan, 2006). PSO algorithm has been applied to a wide range of engineering problems in the literature. Recently some attentions have been focused on hybrid applications and some comparisons with the different heuristics, especially in electrical, mechanical and industrial engineering. Huang and Mohan (2005) proposed a simple hybrid boundary condition that could be used to obtain a robust and consistent PSO performance for high dimensional optimization problems regardless of where the global optimum is located in the search space. In another study, a solution model for the unit commitment problem was obtained using fuzzy logic to address uncertainties in the problem (Victoire & Jeyakumar, 2006). In order to schedule the generating units based on the fuzzy logic decisions, hybrid tabu search, PSO and sequential quadratic programming was used. Jeyakumar, Jayabarathi, and Raghunathan (2006) described an adaptation of the PSO algorithm to solve various types of economic dispatch problems in power systems and solved these problems using both the PSO method and the classical evolutionary programming approach. The following examples are related with the manufacturing systems specifically. Arumugam and Rao (2005) presented several novel approaches of PSO algorithm with new particle speed equations and inertia weights to solve the optimal control problems of a class of hybrid manufacturing systems. Xia and Wu (2005) proposed a hierarchical solution approach to solve multi objective flexible job-shop scheduling problem by using PSO algorithm and simulated annealing algorithm. Another work focused on the different scheduling mechanisms which are designed to generate optimum scheduling. These nontraditional optimization procedures, such as genetic algorithm, simulated annealing, memetic algorithm and PSO algorithm were then implemented for solving the scheduling optimization problem of flexible manufacturing systems and results were compared by Jerald, Asokan, Prabaharan, and Sravanan (2005). An activity network based multi-objective partner selection model in a supply chain was developed by Zhao, Yu, and Chen (2005). They proposed a hybrid heuristic algorithm based on PSO and simulated annealing to solve this multi-objective problem. Although the PSO algorithm has been applied to the different industrial areas and the different engineering problems, only a few applications are known in the supply chain management, especially in the layout design, network design or warehouse design in a supply network. There is only one study encountered in the literature recently. In this work, Paul, Asokan, and Prabhakar (2006) proposed the PSO algorithm to derive better solutions for unequal-area facility layouts having inner walls and passages. Ant colony optimization and PSO are the most popular swarm inspired methods in the computational intelligence area. While ant colony optimization inspires by the behaviors of ants, PSO originates as a simulation of simplified social system which graphically simulates the composition of a flock of bird or a school of fish.

### 3.1. Original PSO algorithm

In the implementation of the PSO, the population is referred to as a swarm and each individual as a particle. It is initialized with a random particles group and then searches the solution space for optima by updating generations. The general PSO algorithm is represented step by step in Fig. 2.

In PSO, each particle included by social structure keeps in mind its best position and uses this as a factor affecting its speed. *A* particle gains speed toward its individual best position considering with how far away from that point. It also shows the same behavior for the global best position. In other words, while it is scanning the surface, it is affected by the global best position and adjusts its own speed. In the situation of that it is far from the global best position, there will be a higher change in its speed and direction. Individuals (particles) of a swarm show inclination to change their movements by using the information below

- Position of the $i$th particle in $k$th iteration $x_i^k$ ($k = 0, \ldots iter_{\max}$ and $i = 1, \ldots, N$).
- Speed of the particle $i$ in iteration $k$ $V_i^k$,
- Best position of the particle $i$ (local best) ($\mathrm{Pbest_i}$),
- Best position of the particle group (global best) (gbest).

**Initialization (for $k = 0$)**

For $i = 1$ to $N$

Assign particles randomly in solution space ( $x_i^k$ )

Generate initial solutions S( $x_i^k$ )

Assign $\mathrm{Pbest_i}$ = initial solutions S( $x_i^k$ )

Assign $\mathrm{Gbest_i}$ = the obtained best solution among all particles

Generate initial velocities randomly ( $V_i^k$ )

Add velocities to the corresponding particles ( $x_i^{k+1}$ )

**Improve the solution (for $k = 1$ to $iter_{max}$)**

Determine the inertia weight ( $w_k$ )

For $i = 1$ to $N$

Update velocities ( $V_i^k$ )

Modify the current positions ( $x_i^{k+1}$ )

Update the $\mathrm{Pbest_i}$

Update the $\mathrm{Gbest_i}$

**Finalize the algorithm ($k = iter_{max}$)**

Assign the $\mathrm{Gbest_i}$ = Ubest and stop.

Fig. 2. Algorithmic schema for general PSO algorithm.

Each individual's speed changes according to the formula in Eq. (7);

$$V_i^{k+1} = X(w * V_i^k + C_1 * (\text{Rnd}) * (\text{pBest}_i - x_i^k) + C_2 * (\text{Rnd}) * (\text{gBest}_i - x_i^k)) \tag{7}$$

$V_i^k$ $i$th individual's speed on $k$th iteration
$x_i^k$ $i$th individual's position on $k$th iteration
$w$ inertia function
$C_i$ inertia factor
Rnd random number
$\text{Pbest}_i$ individual's best position
Gbest global best position
$X$ constriction factor

Inertia value of the equation changes on the each iteration. This change is based on the logic of decreasing from the value determined to minimum value according to inertia function. The objective is to converge the created speed by diminishing on the further iterations; hence more similar results can be obtained.

Inertia function is obtained as follows:

$$w = w_{\max} - \left(\frac{w_{\max} - w_{\min}}{\text{iter}_{\max}}\right) * k \tag{8}$$

$w_{\max}$ first inertia force
$w_{\min}$ minimum inertia force
$iter_{\max}$ maximum iteration number

The values of $C_i$ inertia factor and $w_{\max}$ and $w_{\min}$ inertia forces are investigated by Shi and Eberhart (1998a, 1998b). It is found that these values should not be changed from a problem to another. They fixed the values of these parameters as; $C_i = 2$, $w_{\max} = 0.9$ and $w_{\min} = 0.4$. Hence, in this study we also used these fixed values.

Positions of the particles change by speeds as shown in Eq. (9)

$$x_i^{k+1} = x_i^k + vx_i^{k+1} \tag{9}$$

Same procedure is reiterated for each dimension.

As it can be seen above, the advantages of the PSO are easiness to implement and having few parameters to adjust. However, there are some difficulties related with applying PSO on constricted models even it has been successfully applied in many areas, such as function optimization, artificial neural network training, fuzzy system control, and other areas (Eberhart & Shi, 1998).

### 3.2. *Difficulties of using PSO on constricted models and improvements*

PSO determines minimum or maximum value of any function easily between specific bounds. However, it has some difficulties when it is necessary to ensure some constriction equations. According to PSO, the difference between a constricted model and a function is based on a high contradiction probability of any chosen point and constriction equation.

Another effect of this problem is no points ensuring the constrictions on the first iteration, hence no global best can be found. If a global best can be found on the first iteration, then algorithm can reach to better results around this point. But if global and local bests can not be determined on the first iteration, then PSO is useless.

Considering all this cases, keeping sensibility low on the first iteration and redounding step by step of the procedure will grow up the efficiency of PSO. On the first iteration, the best integer will be chosen, then other points will be investigated on decimal degree and this process will reach to the desirable sensibility about the last iterations. Therefore the probability of finding the best points on the first iteration arises and no need to redound particle number too much so no need to extend solving time.

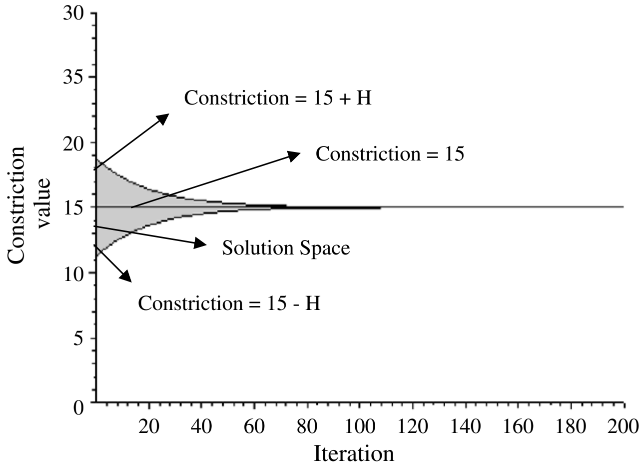


Fig. 3. Funnel effect.

When solution set is small, enlarging the solution set to cover some of the neighbor points can be a useful method to use. However, this wideness must shrink while the iteration number grows up, hence deviation from the constrictions can be diminished.

Stretching the constrictions including the points on a specific approximation to the solution set is useful to overcome this problem. But this elasticity should not be permanent; it should disappear to the maximum iteration so that we can reach to the real points ensuring the constrictions. This effect is shown in Fig. 3. On the each iteration, original constriction value is stretched by a certain quantity. This elasticity changes according to the direction of the constriction too. Equation constrictions are positively and negatively stretched to reach an interval. If it is bigger than constriction, then the value is pulled to higher value, else the value is pulled to a lower value.

$+\backslash-$ **H** value, shown on the graphic, is calculated as in Eq. (10)

$$H = \text{Constriction} * M_{\text{ep}} * \exp(-\text{Iter}/M_{\text{iter}} * 0.1) \tag{10}$$

where in the each iteration

$M_{\text{ep}}$ maximum flexion portion

$M_{\text{iter}}$ maximum iteration number

Hence the constriction will be stretched by $M_{\text{ep}}$ on the first iteration.

To adapt standard PSO to this technique, some changes should be done into the criteria ensuring best point choice. According to this, if any point is the best, then it will continue on the solution set. However, if the point remains out of the solution set, even it is still the best one, the first point caught in the solution set will be chosen instead of that point. This will prevent missing a better point in the solution set.

The pseudo code of the developed PSO algorithm embedding the required improvements to deal with the shortcomings of the standard PSO algorithm is detailed below.

**Initialization**

Solve the objective function for the worst value (by using general PSO algorithm)

Assign $\text{Gbest}_{\text{i}}$ = The worst solution of general PSO algorithm

**(for *k* = 0)**

For $i = 1$ to $N$

Assign particles randomly in solution space

If the sensitivity is increasing step by step, round the solution to integer value

Generate initial solution $S(x_i^k)$

Generate initial speed randomly $(V_i^k)$

If the sensitivity is increasing step by step, round the speed to integer value
AssignPbest$_i$ $= S(x_i^k)$
Keep Gbest$_i$ = The worst solution of general PSO algorithm

**Improve the solution (*k = 1 to iter$_{max}$*)**
Determine the inertia weight

$$\left[w_k = w_{\max} - \left(\frac{w_{\max} - w_{\min}}{iter_{\max}}\right) * k\right]$$

For $i = 1$ to $N$
Update speed $(V_i^k)$ as
$V_i^{k+1} = X(w * V_i^k + C_1 * (Rnd) * (Pbest_i - x_i^k) + C_2 * (Rnd) * (Gbest_i - x_i^k))$
Modify the current positions $(x_i^k)$ as
$[x_i^{k+1} = x_i^k + v_i^{k+1}]$
If $x_i^{k+1}$ is still in the boundaries
Accept the current position
Else
Reduce or increase to the boundaries
Endif
Calculate the objective function value $S(x_i^{k+1})$
If the funnel effect is activated;
Repeat for each constraint
$H$ = Constraint value * $M_{ep}$ * exp (-*e no*/$M_{iter}$ * 0,1)
if the constraint is an equality (=) constraints
Separate into (<=) and (>=) constraints
Endif
Add $H$ value to the right hand side values of (<) and (<=) constraints
Subtract $H$ value from the right hand side values of (>) and (>=) constraints
Else
Consider the current right hand side values for the constraints
Endif
Control whether the constraints are ensured
If Gbest$_i$ position ensures constraints then
Assign tempBoolean = True
Else
Assign tempBoolean = False
End if
If the constraints are ensured
If$S(x_i^{k+1}) >$ Pbest$_i$
Assign Pbest$_i$ $= S(x_i^{k+1})$
Else
Assign Pbest$_i$ $= S(x_i^k)$
Endif
Else
Assign Pbest$_i$ $= S(x_i^k)$
Endif
If the Pbest$_i$ > = Gbest$_i$ OR if tempBoolean = False
Gbest$_i$ = Pbest$_i$

**Finalize the algorithm (k=iter$_{max}$)**
Assign the Gbest$_i$ = Ubest and stop.

In order to solve an optimization problem with this improved PSO algorithm, solution space should be constricted and the numbers of particles, iterations, max/min inertia forces and factors have to be defined at the beginning. In addition, considering the funnel effect activation, maximum flexion portion should be determined.

## 4. Proposed PSO algorithm for multiple-level warehouse layout design problem

The developed warehouse design model is applied to a distribution-type warehouse, which is considered on the construction phase, and the solution is obtained by using improved PSO algorithm. The warehouse is planned to serve for six product groups, i.e., personal cleaning products, house and kitchen cleaning products, food products, chemical raw materials, electronics, and ceramic objects of which include 10, 14, 6, 2, 11 and 7 item types, respectively. Items belonging to different product groups are stored for different periods in the shelves of the warehouse.

A classification phase that separates the storage products belonging to the different groups is realized before applying the model to warehouse design problem. The aim is to determine which product should be put in which class and the required size for each class, i.e., A, B, or C. Here, the closeness to the warehouse door decreases from class A products to class B and C products. Three criteria, namely, turnover rate (circulation speed of products on shelves), fragility, and weights are chosen to rate the products by using the analytic hierarchy process (AHP) methodology. The AHP is a multi-criteria decision-making method that uses hierarchic or network structures to represent a decision-making problem and develops priorities for the alternatives based on the decision maker's judgments throughout the system (Saaty, 1980). The reasons for including the AHP in our study are especially the qualitative data and weighting requirement of the criteria because the qualitative factors are often complicated and sometimes conflicting. The calculations of the product ratings are realized by Expert Choice Package Program. First, the criteria are compared with each other and the priorities for turnover rate, fragility and weights are obtained as 0.669, 0.088 and 0.243, respectively. Then, the products are rated for each criterion by verbal judgments such as, very high, high, normal, low and very low. At the end of this evaluation process, the priorities for all items of each product groups are determined. Considering with this data, the items of product groups are allocated to the A, B, and C classes as 14, 24 and 12 items respectively. And the ordering probabilities of each class are determined as 0.6, 0.3 and 0.1 mainly considering the turnovers. The throughput of the warehouse is 120,000 palletized products in a year and the whole capacity of the warehouse is 6000 pallets. The total storage capacity of the warehouse is divided to the classes into the sizes of 3000 ($N_A$), 2000 ($N_B$), and 1000 ($N_C$) pallets, considering the ordering probabilities and storage periods.

The second data set is related with the dimensions in the storage area. *A* back to back shelve warehouse design is considered and the width of 2.2 m, length of 0.9 m and height of 1.0 m for a storage space in a double shelf is determined. The width of an aisle between shelves is 2.0 m and also the width of a dock is 4 m which is determined considering the aisles' width. The width of a dock can be unconsidered in the situation of using one dock. But if the number of docks increases, it should be added to the formulation as a parameter. The only one cost factor, material handling cost, is calculated by taking the cost of workers, forklift usage, fuel consumption, and depreciations rates into account and $1.13 * 10^{-3}$ \$/m is obtained.

## 5. Computational results and discussions

Above-mentioned data are inserted to the model formulation as the parameters. Then, the model is simplified to make it convenient to solve with the PSO algorithm. The PSO algorithm, which uses funnel affect to be able to embedding the equality constraints to the objective function, is coded by using Visual Basic package program. Sixty particles are used in each iteration and the maximum number of iterations is determined as 200. Fig. 4 shows the model inputs and obtained results screen of the program. The solution process is ended in 22 s and at 62nd iteration. Due to the integer values are tried to be obtained, the decimal sensitivity of the algorithm is constant and zero.

The variables X1, X2 and X3 shown in Fig. 2 denote the *m, n* and *h* variables of the model, respectively. The obtained result is a warehouse which includes 25 shelves, 8 storage spaces in each shelf, and height of 15 stor-

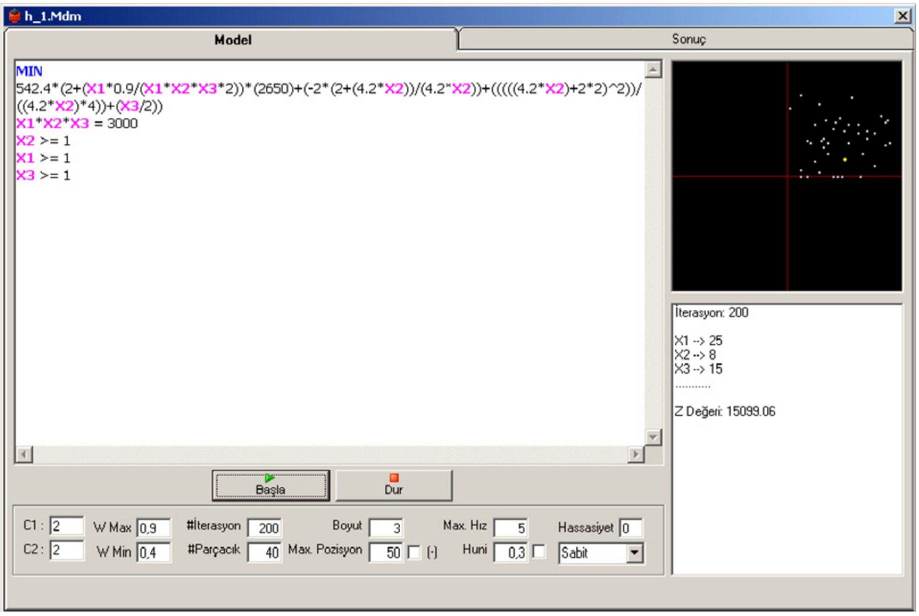


Fig. 4. Model inputs and obtained results screen.

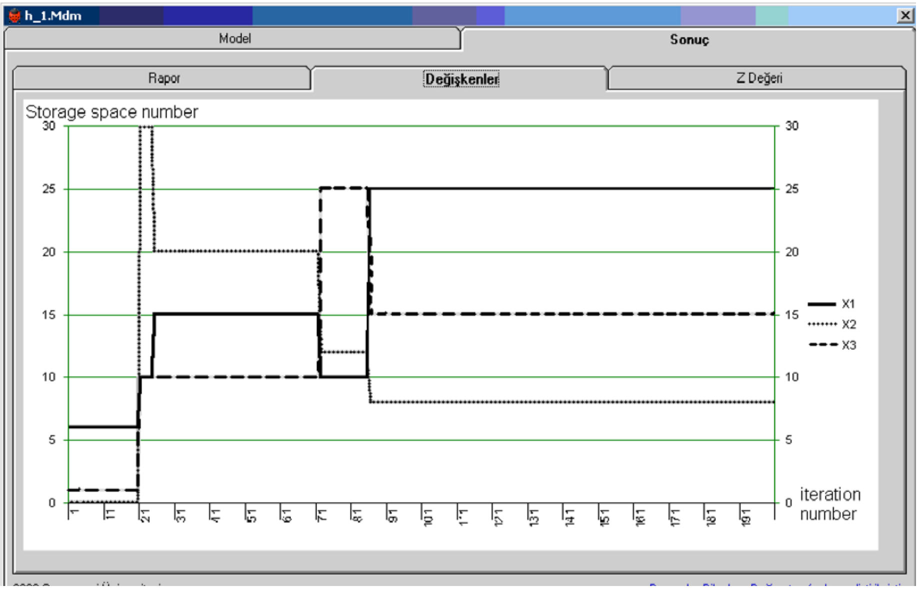


Fig. 5. Improving of the model variables in the iterations.

age spaces. The changes in the variable values and objective function value, while iterations are increasing, are shown in Figs. 5 and 6.

The objective function value can also be denoted as the total distance of picking and putting away of the palletized products for one year as 13,362,000.00 m and the cost of $15,099.06. This version of the model does not consider any physical constraint, but it can provided by putting some limitations to the dimensions and by increasing the dock numbers for getting more realistic results or solutions for specific situations.

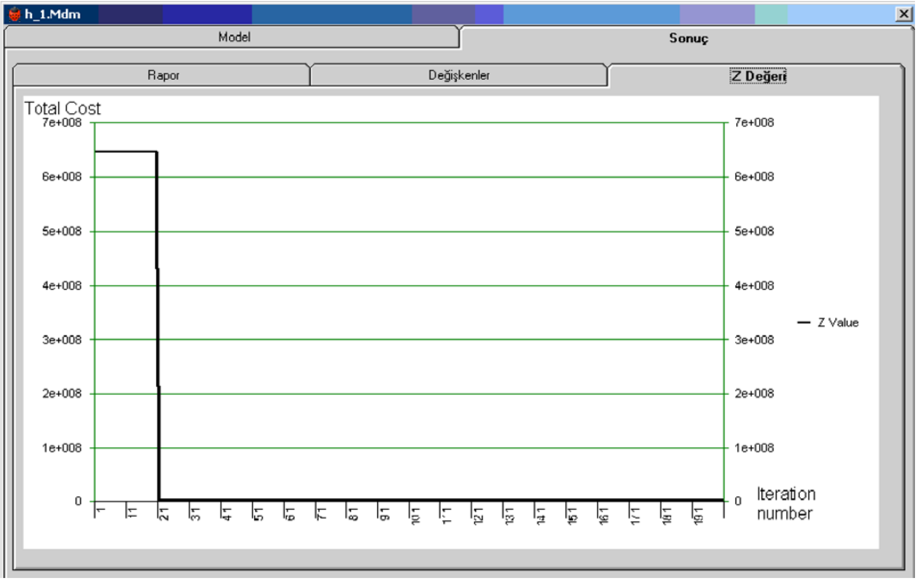


Fig. 6. Improving of the objective function value in the iterations.

## 6. Scenario analysis

There are some different situations that have minor or major affects on the results of the multi-level warehouse design model solution. Firstly, all three dimensions of the travel distances in the warehouse are given in equal weights in the cost minimization objective function. However, the height dimension may be handled differently as the speed along this dimension is unlikely to be close to the speed along the other two dimensions. Moreover as the maximum height to be reached increases, the investment cost of the storage and retrieval equipment needed will likely increase at a nonlinear rate. Due to the speed differentiations, traveling time will increase exponentially while the height of the shelves increasing. By using the technical data of a few material handling equipments, it is decided to change the average travel distance $T_h$, which may also be thought as travel time, to a nonlinear function (Eq. (11))

$$T_h = \mathrm{e}^{\sqrt{\frac{h}{2}}} \tag{11}$$

Another alteration in the model is to assume that there is not only one dock in the horizontal wall of the warehouse. In order to avoid the possibility of waiting goods in the dock queue and to increase the service quality and throughput capacity, having multiple docks on the horizontal wall for entrance and exit may be a necessity. To embed this situation to the model, it should be assume that the warehouse has $r$ docks with a width of $2a$. The $l_k$ is the distance between left wall of the warehouse and the middle of the $k$th dock (Eq. (12))

$$l_k = \frac{k(u+2a)}{r+1} - a \tag{12}$$

The possibility of moving to left or right from the $k$th dock for picking/putting a good from A, B or C classes is $\frac{l_k}{u}$ and $\frac{u-l_k}{u}$, respectively. And the average travel distances to left and right directions are $\frac{l_k}{2}$ and $\frac{u-l_k}{2}$, respectively. Thus, the average traveling distance ($t_k$) to left or right directions can be expressed as follows.

$$t_k = \frac{l_k}{u}\frac{l_k}{2} + \frac{(u-l_k)}{u}\frac{(u-l_k)}{2} \tag{13}$$

When $l_k$ and the probabilities of picking and putting a good from different classes is embedded to the model formulation and the required calculations are completed, the obtained average distance traveled in the horizontal axis of the warehouse for all the docks will be as shown in Eq. (14)

$$T_u = -\frac{a(a+u)}{2u} + \frac{(u+2a)^2}{ur(r+1)^2}\sum_{k=1}^{r} k^2 \tag{14}$$

After these improvements in the model, the objective function that represent the total annual handling cost ($C_{\text{ver1}}$) for the shelf configuration will be derived as shown in Eq. (15)

$$C_{\text{ver1}} = 4dC_h\left[a + \frac{m}{K}L\left[(N_A+N_B)\left(1-\frac{P_a}{2}\right) + \frac{P_c}{2}(N_B+N_C) - \frac{N_B}{2}\right] - \frac{a(a+u)}{u} + \frac{(u+2a)^2}{ur(r+1)^2}\sum_{k=1}^{r} k^2 + \frac{h}{2}\right] \tag{15}$$

Also, Eq. (15) is differentiated by applying the changes on height dimension that is given in Eq. (11). The last version of the objective function ($C_{\text{ver2}}$) has obtained as shown in Eq. (16)

$$C_{\text{ver2}} = 4dC_h\left[a + \frac{m}{K}L\left[(N_{\text{A}}+N_{\text{B}})\left(1-\frac{P_a}{2}\right) + \frac{P_c}{2}(N_{\text{B}}+N_{\text{C}}) - \frac{N_{\text{B}}}{2}\right] - \frac{a(a+u)}{u} + \frac{(u+2a)^2}{ur(r+1)^2}\sum_{k=1}^{r} k^2 + \mathrm{e}^{\sqrt{\frac{h}{2}}}\right] \tag{16}$$

This improved nonlinear objective functions with previously given nonlinear capacity constraints are solved by using PSO algorithm for one to nine docks. As a result, 18 different scenarios are obtained for multiple-level warehouse design problem (Fig. 7).

The results, which are given in Table 2 show that, while the dock numbers are increasing, the annual handling cost increases independently from $T_h$ function. The main reason of this situation is that the good arrivals to any dock have the equal probabilities. For example, a good can arrive to ninth dock to be placed in the first double-shelf which has the longest horizontal path. Due to the above mentioned possibilities, the total handling cost increases. Also, the change on the obtained values of the variables, which represents the dimensions of the warehouse, is an expected situation while the numbers of docks increase. Another main point is the constriction of the height dimension by using a coefficient or a function that does not permit to high level increase which is infeasible for a warehouse layout. As it is seen, the height dimension $h$ is decreasing from 15 or 20 to 10 for all dock numbers. Due to the exponential increase of $h$ function, there is no necessity to put an upper bound for the height dimension. It already prevents to exceed 10 shelves for the height dimension.

There is one more main point related with the solution time of the PSO algorithm. All the scenarios are also solved by LINGO Release 9.0 which uses branch and bound algorithm (B and B) for pure integer nonlinear problems. LINGO is able to find only local optimum results with longer solution times in this problem. When the solution performance of PSO algorithm is compared with B and B, it is superior for most of the situations in terms of the solution times and number of iterations to reach to the best solution especially for scenarios that uses exponential height dimensions (Table 3). Furthermore, for all situations, PSO finds less than or equal cost objective values. As a result, even it is hard to compare an algorithm that gives optimal result and another algorithm that gives approximate result, it is proven that PSO is superior in terms of obtained objective values and solution times.

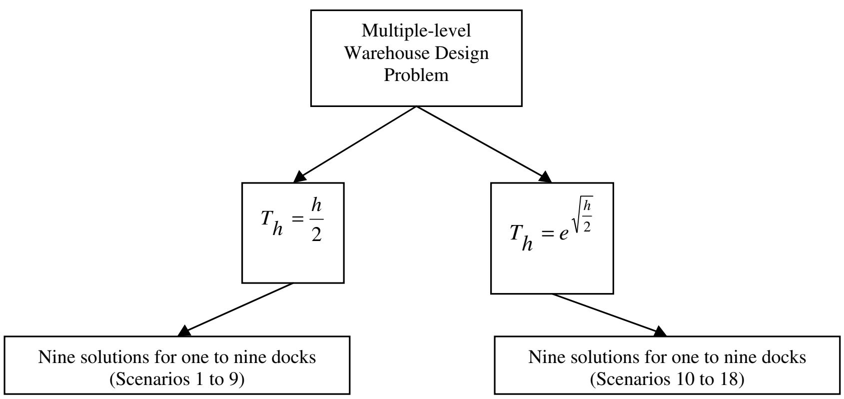


Fig. 7. Scenarios applied to the main problem.

Table 2
The obtained results for 18 scenarios

| | | Number of docks | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| $T_h = \frac{h}{2}$ | Objective function | 15099.06 | 15733.01 | 16049.98 | 16233.62 | 16335.58 | 16408.4 | 16525.44 | 16578.27 | 16981.86 |
| | X1 (m) | 25 | 25 | 25 | 25 | 25 | 25 | 25 | 25 | 20 |
| | X2 (n) | 8 | 8 | 8 | 6 | 6 | 6 | 8 | 8 | 10 |
| | X3 (h) | 15 | 15 | 15 | 20 | 20 | 20 | 15 | 15 | 15 |
| $T_h = e^{\sqrt{\frac{h}{2}}}$ | Objective function | 18323.07 | 19082.14 | 19461.68 | 19689.4 | 19841.21 | 19949.65 | 20030.98 | 20094.24 | 20144.84 |
| | X1 (m) | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 |
| | X2 (n) | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| | X3 (h) | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |

Table 3
Comparison of the solution times of PSO algorithm versus branch and bound algorithm

| Number of docks | Linear height dimension | | | | | Exponential height dimension | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Iteration number | | Solution time (s) | | | Iteration number | | Solution time (s) | | |
| | PSO | B and B | PSO | B and B | % Change | PSO | B and B | PSO | B and B | % Change |
| 1 | 62 | 6152 | 23 | 21 | −0.05 | 27 | 1514 | 22 | 22 | −0.05 |
| 2 | 39 | 9683 | 22 | 34 | 0.35 | 119 | 489 | 22 | 9 | −1.44 |
| 3 | 67 | 8327 | 22 | 27 | 0.19 | 103 | 5396 | 22 | 48 | 0.54 |
| 4 | 147 | 9934 | 22 | 34 | 0.35 | 127 | 9469 | 22 | 104 | 0.79 |
| 5 | 150 | 6507 | 22 | 21 | −0.05 | 89 | 10273 | 22 | 126 | 0.83 |
| 6 | 106 | 3565 | 22 | 10 | −1.2 | 121 | 7872 | 22 | 77 | 0.71 |
| 7 | 98 | 14467 | 20 | 54 | 0.59 | 61 | 5280 | 22 | 112 | 0.82 |
| 8 | 107 | 1786 | 22 | 7 | −2.14 | 103 | 4563 | 22 | 97 | 0.77 |
| 9 | 144 | 9406 | 22 | 30 | 0.27 | 93 | 5723 | 22 | 87 | 0.75 |

However, if the problem should be solved by continuous variables in the situation of having possibility to use decimal data, the improved PSO algorithm has some shortcomings related with the solution time. In that condition, the solution space is getting larger and the required particle numbers and iteration numbers are increasing. For example, solving the problem by using continuous variables for two docks and exponential height dimension, the required particle numbers increase from 60 to 100 and required iteration numbers from 119 to 343. As a result of these changes, solution time increases from 22 to 127 s and the deviation in the constraints is occurred as 0.0099.

## 7. Conclusions

This study aims to model the problem of designing a multiple-level warehouse considering the handling costs in three dimensions. One of the contribution of our proposed model is to enhance the two-dimensional warehouse design (Bassan et al., 1980) to the multiple-level warehouse design considering a class-based storage strategy which includes three palletized product types, namely, A, B and C. But, the main difficulty of solving this kind of designing problems is to struggle with nonlinearity in the variables and the constraints for finding an optimal solution. To overcome this difficulty, we used a novel algorithm PSO which is able to find near optimal results in a short time. To adapt the standard PSO algorithm to the constrained problems, which can be applied to only unconstrained problems, it is modified by funnel effect and penalty methods.

Then, the dock numbers on the same wall of the warehouse is increased one by one from one to nine docks to see the changes on the total handling cost and the change on the configuration of the warehouse in three dimensions. It is seen that the total handling cost increases by the increase of the dock numbers even the configuration is changed or not. This is another contribution to multiple-warehouse layout design problem. However, some future directions can be added to the existing study. For example, there is a trade off between total material handling cost and vehicle waiting cost for delivering the goods at the dock while the dock number increases. Therefore, to determine the optimal dock number of this kind of warehouses, both the material handling and vehicle waiting costs should be considered together. Vehicle waiting costs can be calculated by simulation techniques and embedded to the warehouse design problem in a future work.

## References


Arumugam, M. S., & Rao, M. V. C. (2005). On the optimal control of single-stage hybrid manufacturing systems via novel and different variants of particle swarm optimization algorithm. *Discrete Dynamics in Nature and Society, 3*, 257–279.

Bassan, Y., Roll, Y., & Rosenblatt, M. J. (1980). Internal layout design of a warehouse. *AIIE Transactions, 12*(4), 317–322.

Chen, M. C., & Wu, H. P. (2005). An association-based clustering approach to order batching considering customer demand patterns. *Omega, 33*, 333–343.

Chen, P., Guo, Y., Lim, A., & Rodrigues, B. (2006). Multiple crossdocks with inventory and time windows. *Computers and Operations Research, 33*, 43–63.

Chew, E. P., & Tang, L. C. (1999). Travel time analysis for general item location assignment in a rectangular warehouse. *European Journal of Operational Research, 112*, 582–597.

De Koster, M. B. M., Van der Poort, E. S., & Wolters, M. (1999). Efficient order batching methods in warehouses. *International Journal of Production Research, 37*(7), 1479–1504.

De Koster, R., & Van der Poort, E. S. (1998). Routing order pickers in a warehouse: a comparison between optimal and heuristic solution. *IIE Transactions, 30*, 469–480.

Eberhart, R. C., Kennedy, J. (1995). A new optimizer using particle swarm theory. In Proceedings of 6th Symposium Micro Machine and Human Science, Nagoya, pp. 39–43.

Eberhart, R. C., Shi, Y. (1998). Comparison between genetic algorithms and particle swarm optimization. In Proceedings of 7th ICEC, pp. 611–616.

Eussuf, M., Lansey, K., & Pasha, F. (2006). Shuffled frog-leaping algorithm: A memetic meta-heuristic for discrete optimization. *Engineering Optimization, 38*(2), 129–154.

Garey, M. R., & Jhonson, D. S. (1979). *Computers and intractability: a guide to the theory of NP-completeness*. Newyork: W.H. Freeman.

Haq, A. N., Sivakumar, K., Saravanan, R., & Karthikeyan, K. (2006). Particle swarm optimization algorithm for optimal machining allocation of clutch assembly. *International Journal of Advanced Manufacturing Technology, 27*, 865–869.

Hausman, W. H., Schwarz, L. B., & Graves, S. C. (1976). Optimal storage assignment in automatic warehousing systems. *Management Science, 22*(6), 629–638.

Heragu, S. S., Du, L., Mantel, R. J., & Schuur, P. C. (2005). Mathematical model for warehouse design and product allocation. *International Journal of Production Research, 43*(2), 327–338.

Hsieh, L. F., & Tsai, L. (2006). The optimum design a warehouse system on order picking efficiency. *International Journal of Manufacturing Technology, 28*, 626–637.

Hsu, C. M., Chen, K. Y., & Chen, M. C. (2005). Batching orders in warehouses by minimizing travel distance with genetic algorithms. *Computers in Industry, 56*, 169–178.

Huang, T., & Mohan, A. S. (2005). A hybrid boundary condition for robust particle swarm optimization. *IEEE Antennas and Wireless Propagation Letters, 4*, 112–117.

Jayaraman, V., & Ross, A. (2003). A simulated annealing methodology to distribution network design and management. *European Journal of Operational Research, 144*, 629–645.

Jerald, J., Asokan, P., Prabaharan, G., & Sravanan, R. (2005). Scheduling optimization of flexible manufacturing systems using particle swarm optimization algorithm. *International Journal of Advanced Manufacturing Technology, 25*, 964–971.

Jewkes, E., Lee, C., & Vickson, R. (2004). Product location, allocation and server home base location for an order picking line with multiple servers. *Computers and Operations Research, 31*, 623–636.

Jeyakumar, D. N., Jayabarathi, T., & Raghunathan, T. (2006). Particle swarm optimization for various types of economic dispatch problems. *Electrical Power and Energy Systems, 28*, 36–42.

Kennedy, J., & Eberhart, R. C. (1995). Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks, 4*, 1942–1948.

Kim, B., Graves, R. J., Heragu, S. S., & Onge, A. St. (2002). Intelligent agent modeling of an industrial warehousing problem. *IIE Transactions, 34*, 601–612.

Kim, B., Heragu, S. S., Graves, R. J., & Onge, A. St. (2003). A hybrid scheduling and control system architecture for warehouse management. *IIE Transactions on Robotics and Automation, 19*(6), 991–1001.

Kusiak, A., & Heragu, S. S. (1987). The facility layout problem. *European Journal of Operation Research, 29*, 229–251.

Lai, K. K., Xue, J., & Zhang, G. (2002). Layout design for a paper reel warehouse: A two-stage heuristic approach. *International Journal of Production Economics, 75*, 231–243.

Larson, T. N., March, H., & Kusiak, A. (1997). A heuristic approach to warehouse layout with class-based storage. *IIE Transactions, 29*, 337–348.

Lee, M. K., & Elsayed, E. A. (2005). Optimization of warehouse storage capacity under a dedicated storage policy. *International Journal of Production Research, 43*(9), 1785–1805.

Naka, S., Genji, T., Yura, T., & Fukuyama, Y. (2003). A hybrid particle swarm optimization for distribution state estimation. *IEEE Transactions on Power Systems, 18*(1), 60–68.

Paul, R. C., Asokan, P., & Prabhakar, V. I. (2006). A solution to the facility layout problem having passages and inner structure walls using particle swarm optimization. *International Journal of Advanced Manufacturing Technology, 29*, 766–771.

Petersen, C. G. (1999). The impact of routing and storage policies on warehouse efficiency. *International Journal of Operations and Production Management, 19*(10), 1053–1064.

Poulos, P. N., Rigatos, G. G., Tzafestas, S. G., & Koukos, A. K. (2001). A Pareto-optimal genetic algorithm for warehouse multi-objective optimization. *Engineering Applications of Artificial Intelligence, 14*, 737–749.

Roodbergen, K. J., & De Koster, R. (2001a). Routing order pickers in a warehouse with a middle aisle. *European Journal of Operational Research, 133*, 32–43.

Roodbergen, K. J., & De Koster, R. (2001b). Routing methods for warehouses with multiple cross aisles. *International Journal of Production Research, 39*(9), 1865–1883.

Rosenblatt, M. J., & Eynan, A. (1989). Deriving the optimal boundaries for class-based automatic storage/retrieval systems. *Management Science, 35*(12), 1519–1524.

Rouwenhorst, B., Reuter, B., Stockrahm, V., Van Houtum, G. J., Mantel, R. J., & Zijm, W. H. M. (2000). Warehouse design and control: Framework and literature review. *European Journal of Operational Research, 122*, 515–533.

Saaty, T. L. (1980). *The analytic hierarchy process*. Pittsburgh, PA: RWS Publications.

Shi, Y., & Eberhart, R. C. (1998a). A modified particle swarm optimizer. In *Proceedings of the IEEE international conference on evolutionary computation* (pp. 69–73). Piscataway, NJ: IEEE Press.

Shi, Y., & Eberhart, R. C. (1998b). Parameter selection in particle swarm optimization. In *Evolutionary Programming, VII: Proc. EP98* (pp. 591–600). New York: Springer-Verlag.

Shi, Y., & Eberhart, R. C. (1999). Empirical study of particle swarm optimization. In *Proceedings of the congress on evolutionary computation* (pp. 1945–1950). New Jersey: IEEE Service Center.

Van den Berg, J. P. (1999). A literature survey on planning and control of warehouse systems. *IIE Transactions, 31*, 751–762.

Van den Berg, J. P., & Zijm, W. H. M. (1999). Models for warehouse management: Classification and examples. *International Journal of Production Economics, 59*, 519–528.

Victoire, T. A. A., & Jeyakumar, A. E. (2006). A tabu search based hybrid optimization approach for a fuzzy modeled unit commitment problem. *Electric Power Systems Research, 76*, 413–425.

Xia, W., & Wu, Z. (2005). An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems. *Computers and Industrial Engineering, 48*, 409–425.

Zhang, G. Q., & Lai, K. K. (2006). Combining path relinking and genetic algorithms for the multiple-level warehouse layout problem. *European Journal of Operational Research, 169*(2), 413–425.

Zhang, G. Q., Xue, J., & Lai, K. K. (2002). A class of genetic algorithms for multiple-level warehouse layout problems. *International Journal of Production Research, 40*(3), 731–744.

Zhao, F. Q., Zhang Q. Y., Yu, D. M., Chen, X. H. Yang, Y. H. (2005). A hybrid algorithm based on PSO and simulated annealing and its applications for partner selection in virtual enterprises. Advances in Intelligent Computing, PT 1, Proceedings Lecture Notes in Computer, Science 3644, pp. 380–389.