*İ. Ü. İşletme Fakültesi İşletme İktisadı Enstitüsü Yönetim Dergisi*
Yıl : 25 Sayı : 77 Aralık 2014

# *ÇOK SEVİYELİ DEPO YERLEŞİM DÜZENLEMESİ İÇİN PARÇACIK SÜRÜ OPTİMİZASYON ALGORİTMASI TABANLI TASARIM METODOLOJİSİ*

**S. Alp BARAY**
Istanbul University Faculty of Engineering,
Industrial Engineering Department

**Emre ÇAKMAK**
Okan University Faculty of Economics and Business Administration,
International Logistics Department

## ÖZET

Farklı z-eksenli depolama alanlarına sahip çok seviyeli depo yerleşim düzenlemesi problemi araştırılmıştır. Bu çalışmada, fiziksel kısıtlar altında farklı depolama alanlarına z-ekseni boyunca farklı grupların yerleştirildiği çok seviyeli depo yerleşim düzenlemesi tasarım metodolojisi geliştirilmiştir. Önerilen matematiksel model NP-zordur. Parçacık Sürü Optimizasyon (PSO)'da çoğunlukla kullanılan sınırlandırma koşulları, parçacıkların olabildiğince kabul edilebilir çözüm uzayı içerisinde tutmaktadır. Buna ek olarak parçacıkları kabul edilebilir çözüm uzayında kalmasını için iki yeni sınırlandırma koşulu önerilmiştir. Ayrıca, parçacıkların kabul edilebilir çözüm uzayında uygun olmayan çözümleri araştırmasıyla ortaya çıkan zaman kaybı probleminin üstesinden gelebilmek için parçacıkların başlangıç değerleri için önerilen atama algoritması kullanılmıştır.
**Anahtar Kelimeler:** Depo Yerleşim Düzenlemesi, Sınırlandırma Koşulları, Sınıf-bazlı Depolama, Parçacık Sürü Optimizasyon

# *DESIGN METHODOLOGY FOR A MULTIPLE-LEVEL WAREHOUSE LAYOUT BASED ON PARTICLE SWARM OPTIMIZATION ALGORITHM*

## ABSTRACT

The multi-level warehouse layout problem with different z-axis storage spaces is investigated. In this study, a design methodology for a multiple-level warehouse layout (MLWL) is developed in order to minimize the total material handling costs by considering different number of the storage areas allocated to different groups along the z-axis under physical constraints. The proposed mathematical model is NP-hard. Boundary conditions are often used in particle swarm optimization (PSO) in order to keep the particles as much as possible in the allowable solution spaces. Moreover, two new boundary conditions are proposed for keeping the particles in the allowable solution spaces. Besides, a proposed assignment algorithm for particles' initial values is used to overcome the problem of the time lost while particles are searching inappropriate solutions in allowable solution spaces.
**Keywords:** Warehouse layout, Boundary Conditions, Class-based storage, Particle swarm optimization

## INTRODUCTION

Recently, with the rapid diversification and variation of customer needs, companies have been challenged to keep pace with these changes. These changes not only affect the demand forecasts, but also make the decision-making process difficult for managers. On the other hand, the reflections of these changes both have an impact on the company, and affect the entities of the entire supply chain. It is obvious that the success of the supply chain depends on the success of each entity of the supply chain. Hence, it is important for companies to have good communication with not only all their suppliers, but also their customers. This communication can be achieved by supply chain management.

According to the principles of supply chain management, all types of raw materials and products have to be delivered to the customers within short delivery times; and these delivery times have to be reached by minimum inventories. Minimum inventories can be provided by well-managed logistics operations, especially distribution and warehousing. In this context, warehousing is one of the main elements in the supply chain management (Lambert, Stock and Ellram, 1998, s: 268). A company's success or its failure depends on well-managed warehousing (Baker and Canessa, 2009, s: 425–436). Furthermore, a crucial element of any supply chain is a warehouse that has links between producers, distributors and customers.

Warehousing consists of different processes which are receiving, storage, order picking and shipping processes (Gu, Goetschalckx and Mcginnis, 2007, s: 1–21; Van den Berg and Zijm, 1999, s: 519–528). The first process in warehousing is the receiving process; where incoming goods and raw materials are unloaded from the carriers. Both quantity and quality of incoming goods are controlled in the receiving process. The receiving process also involves repacking, relabeling and physical movement. The second process is storage, which is the main function of warehousing. In this process, all goods are stocked and kept under appropriate conditions until the customer makes an order. The next one is order picking process; where all customer orders are collected, clustered, and order picker lists are prepared. In order picking process, all goods are picked from storage locations. Some additional operations can also be done such as repacking, accumulation and sortation. The last process in warehousing is the shipping process; where necessary documents are prepared; all orders are controlled, counted and loaded outgoing trucks.

The warehouse layout design plays an important role on warehousing processes (Yang and Sun, 2004, s: 751–756; Onut, Tuzkaya and Dogac, 2008, s: 783–799). A well-designed warehouse layout can reduce material handling and storage costs. Rouwenhorst, Reuter, Stockrahm, Van Houtum, Mantel and Zijm (2000) categorized the warehouse design and design problems into three decision levels such as strategic level, tactical level and operational level. The strategic level concerns decisions such as specifying the quantity of warehouses and their locations, the number of storage areas, storage systems and handling systems. This level also interested in material flow determination and the choice of the warehouse management information system. Tactical level contains determination of workforce to manage the entire systems, assignment of materials to the storage areas, management of order picking and replenishment principles, capacity planning, etc. In addition, the operational level involves determination of order picker routes and batch size, assignment of docks and appointment of short-term workforce.

The warehouse design involves complex and interrelated decisions such as technical characteristics of handling equipment, equipment selection and layout determination (Heragu Du, Mantel and Schuur, 2005, s: 327–338). The design also requires the simultaneous considerations of all these decisions. When designing a warehouse, there are several types of storage systems and handling equipment alternatives. Moreover, there are complex tasks with many tradeoffs between conflicting objectives within these alternatives.

Warehouse designers need to determine the best storage approach, to select material handling equipments and systems, to design the proper warehouse layout (Ashayeri and Gelders, 1985, s: 285–294). Warehouse layout design aims to plan the storage area for storing different types of materials into one warehouse. This helps to minimize the total material handling and storage costs. The warehouse layout design consists of three plans such as warehouse layout plan, warehouse material handling plan and warehouse operations plan. These plans are:

- Warehouse layout plan: storage area plan, aisle plan, shelf types and sizes, dock plan.
- Warehouse material handling plan: material handling equipment plan and personnel plan.
- Warehouse operations plan: placement/picking policies and assignment policies.

Several studies have been conducted about the warehouse design and warehouse operations. These studies are helped to increase the performance of order picking operations and to estimate the total or average order picker tour length or order picker tour time. Different warehouse design models and integration of storage systems have been developed in the literature. For instance, Rosenblatt and Roll (1984) presented a search procedure to find the optimal solution for warehouse layout and warehouse assignment policy problems. This search procedure helped to determine the warehouse capacity and storage areas by using an analytical method which considers the warehouse initial investment costs, stock-out costs and storage policy related costs. Park and Webster (1989) developed an optimization procedure for the design of three dimensional palletized storage systems. In their procedure, warehouse alternatives were developed by changing storage policies, material handling equipments, control procedures, and the movement of material equipment in an aisle. Then, the optimal storage system was found from these alternatives based on system costs, average time of material equipment tours and area requirement. A heuristic procedure developed by Larson, March and Kusiak (1997) presented a warehouse layout with a class-based storage policy. This procedure included three phases which were determination of aisle plan and storage size, allocation of the materials to storage zones and allocation of floor space. The procedure also determined storage capacity, row depth for floor stacking, directions of material flow, storage policy and allocation of products. Caron, Marchet and Perego (2000) considered to find the aisle layout in a warehouse with the largest gap method under the COI-based storage assignment and random storage assignment. In their study, three different layout types with different number of aisles were presented and compared by using the simulation approach. Roodbergen (2001) proposed a non-linear objective function for determining the aisle layout for warehouses with random storage policy in order to minimize the average tour distance. Le-Duc and Koster (2005) introduced a heuristic procedure for storage area optimization problem in order to determine 2- block warehouse layout with a class-based storage policy. In this study, proposed probabilistic model was used to estimate the average travel distance of picker and was validated by using simulation. Roodbergen and Vis (2006) presented an analytical model for warehouse layout optimization. This model determined warehouse layout by calculating the estimation of the average travel distances in the picking area. The model estimates the average travel distances for two different picking tour policies: S-shaped policy and the largest gap policy. A model was developed by Huertas, Ramírez and Salazar (2007) for estimation and evaluation of the operational costs of alternative layouts for large warehouses or distribution centres. Gu, Goetschalckx and McGinnis (2010) made a comprehensive literature review on warehouse design, performance evaluation, practical case studies, and computational support tools. Cakmak, Gunay, Aybakan and Tanyas (2012) proposed an analytical model for a flow type warehouse design and a U- type warehouse design for determining rack configuration in order to minimize the total picker distance with considering number of docks.

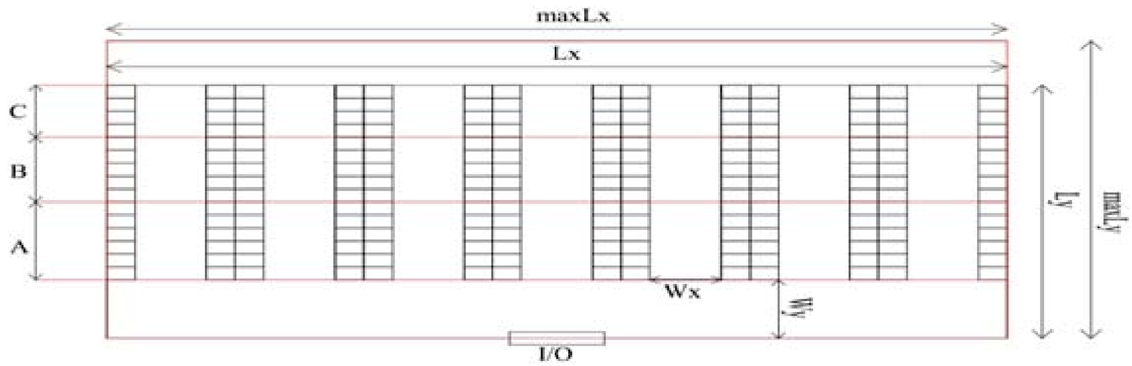
Furthermore, warehouse design models have been developed with simulation and heuristic methods. Queirolo, Tonelli, Schenone, Nan and Zunino (2002) proposed a warehouse layout optimization model in order to minimize distance and the travel time by using genetic and simulation approaches. The simulation played a basic role inside the genetic approach (GA) research process in order to evaluate the fitness. A hybrid (genetic and simulative) approach was possible to analyze multiple scenarios and to share the obtained results. Moreover, the simulation model was developed by using the ProModel software tool to analyze the storage capacity and rack efficiency of different types of warehouses which have various volumes, by Macro and Salmi (2002). This simulation model could be modified to simulate rack systems configurations. This simulation could also answer to several decisions such as equipment/resource limitations, storage method efficiency and order picking method performance. Lai, Xue and Zhang (2002) developed a heuristic procedure for a paper reel warehouse layout optimization problem in order to minimize total transportation costs. They proposed a two-stage iterative solution procedure which were an optimal cell assignment procedure and simulated annealing based heuristic. In the second stage of solution procedure, they modified simulated annealing algorithm in order to solve this combinatorial optimization problem. Zhang, Xue and Lai (2002) introduced a genetic algorithm based heuristic for the product allocation in warehouse which has multiple storage areas in different levels of a warehouse. This GA-based heuristic helps to minimize the horizontal and vertical costs. Zhang and Lai (2006) proposed several hybrid heuristics in order to determine warehouse layout for multi level warehouses. These hybrid heuristics are developed by combining a genetic algorithm and path linking strategy and tested on small and large size problems. Zhang and Lai (2010) considered the MLWLP under adjacency constraints formulating by integer programming. The NP-hard problem is solved by using the greedy Tabu Search and dynamic neighbourhood techniques. In another study by Onut et al .(2008), a warehouse layout plan was developed with particle swarm optimization for multi level distribution warehouses with class-based storage policy. Heragu et al. (2005) formulated a mathematical model for assigning items to functional storage areas and determining functional area size by considering the holding cost and stock-out cost. This mathematical model was solved by using branch-and-bound algorithm, heuristic algorithm and simulated annealing algorithm. Yang and Sun (2004, s: 751–756) proposed a mathematical model and hybrid heuristic based on fuzzy random simulation for a warehouse layout problem.

In this paper, a multi-level warehouse layout mathematical model, which have similarities with Onut et al.’s (2008) MLWLP from the point of the difficulty level of multi-level warehouse layout problem and NP-hardness, is developed. Unlike with Onut et al.’s (2008) MLWLP, different number of the storage areas allocated to different groups along the z-axis under physical constraints are considered in this study.

This paper aims to propose a design methodology for multi-level warehouse layout in order to minimize the total material handling costs. In order to reach this aim, a mathematical model which has different numbers of storage spaces allocated to different product groups along the z-axis is developed. Moreover, boundary conditions, two of them are newly presented in this study, for particle swarm optimization (PSO) are used to solve the proposed model and an assignment algorithm for initial value is developed for constricting solving space. The paper is also focused on the distance from storage location to I/O port (order picking process) and the allocation of items into the warehouse (class based storage policy).

The paper is organized as follows: Section 2 formulates the problem description of multiple-level warehouse layout. In Section 3, first, PSO algorithm is proposed by giving the original PSO algorithm and advancements in terms of solving the constrained problems. Next, the proposed assignment algorithm for initial values of particles is given. In Section 4, examples for the MLWLP are presented and numerical results are reported. Then, the performances of different boundary conditions are compared in Section 5. Finally, Section 6 concludes the research.

**FIGURE 1: Notations of the Warehouse**




## 1. THE FORMULATION OF MULTIPLE-LEVEL WAREHOUSE LAYOUT DESIGN MODEL

This study aims to examine the MLWLP for minimizing the total material handling cost. The same problem was also investigated by Onut et al. (2008) by offering a method for rack configuration for a rectangular warehouse for heterogeneous items stocked in a multiple-level warehouse. This study is extended version of Onut et al's (2008) model , by allocating different numbers of storage spaces for each product groups along the z-axis. The physical constraints are also embedded to the model. Moreover, the formulation of Onut et al.'s (2008) study is generalized for unlimited numbers of groups in this study.

After categorized the items into groups, probabilities of (picking or putting) the orders belonging in all classes can be calculated. This classification is used to locate groups into storage locations. Groups which have higher probability can be placed in easily accessible storage locations.

**TABLE 1: Nomenclatures**

| | |
|---|---|
| D | Yearly throughput of the warehouse |
| $C_h$ | Handling Cost |
| I | Number of categorized group |
| $N_i$ | Number of the storage areas allocated to group i items |
| $P_i$ | Probability of an order belonging to group i items |
| $L_{Xmax}$ | Maximum length in horizontal axis |
| $L_{Ymax}$ | Maximum length in vertical axis |
| $L_{Zmax}$ | Maximum length in height axis |
| $T_x$ | Average travel distance in horizontal axis |
| $T_y$ | Average travel distance in vertical axis |
| $T_z$ | Average travel distance in height axis |
| $T_{yi}$ | Average travel distance in vertical axis for group i items along the y-axis |
| $T_{zi}$ | Average travel distance in height axis for group i items along the z-axis |
| $a_x$ | Occupation of one pallet along the x-axis |
| $a_y$ | Occupation of one pallet along the y-axis |
| $a_{zi}$ | Occupation of one pallet along the z-axis belonging to group i items |
| $w_x$ | Width of the sub-aisle |
| $w_y$ | Width of the main aisle |
| $n_x$ | Number of shelves along x-axis |
| $n_{yi}$ | Number of the storage areas allocated to group i items along the y-axis |
| $n_{zi}$ | Number of the storage areas allocated to group i items along the z-axis |
| $n_{Xmax}$ | Maximum number of shelves along the x-axis |
| $n_{Yi\text{-}max}$ | Maximum number of the storage areas along the y-axis for group i items |
| $n_{Zi\text{-}max}$ | Maximum number of the storage areas along the z-axis for group i items |
| $n_{Xmin}$ | Minimum number of shelves along the x-axis |
| $n_{Ymin}$ | Minimum number of the storage areas along the y-axis |
| $n_{Zmin}$ | Minimum number of the storage areas along the z-axis |

The notation of the parameters and variables are shown in Table 1. Then, Fig. 1 is given to visualize some of the dimensions given in Table 1. In order to make the Figure 1 understandable, the groups are limited to three items groups; but group numbers can be increased.

By using details in Table 1, the proposed mathematical model formulates (1) the appropriate number of storage locations along a shelf, (2) the appropriate number of locations for each groups and (3) the appropriate number of shelves. In other words, warehouse dimensions: length (*Lx*), width (*Ly*) and height (*Lz*) are obtained by the model solution. The decision variables, $n_x$, $n_{Yi}$ and $n_{Zi}$, which are storage locations for group i items in all dimensions, are also determined by model. The total decisions variables are (*2i +1*) twice of number of groups and plus one. The following equations are shown the length (*Lx*), width (*Ly*) and height (*Lz*) of warehouse dimensions' calculations (Eq.1, Eq.2 and Eq.3).

$$L_X = \left(a_x + {}^{w_x}/_2\right) n_X \quad (1)$$

$$L_Y = \sum_{i=1}^{n} a_y n_{Yi} + w_y \quad (2)$$

$$L_{Zi} = (n_{Zi}\, a_{zi}) \quad (3)$$

The first average distance is the average distance in x-axis (horizontal axis). The locations of docks affect the average distances. If there is only one dock, the dock of the warehouse is placed at the middle of the horizontal wall. The distance between the dock and the right vertical wall is length (*Lx/2*) for one dock in horizontal wall. After given the location of dock, the probability of carrying a group to right-side or left-side of the dock is equal. But, if there is a need for more than one dock in the horizontal wall in order to increase the service quality and throughput capacity, a new dock can be added to the model. This new dock can also help to avoid the possibility of trucks (which are) waiting to take service. It is assumed that the warehouse has *d* docks with *a* width of *2a*. The equation (Eq.4) formulates the distance ($l_k$) between left wall of the warehouse and the middle of the *k*th dock. The possibility of carrying right side and left side are $l_k /Lx$ and $(Lx – l_k )/Lx$, respectively. The average travel distance to right side and left side are $l_k /2$ and *(Lx - lk)/2* (Eq. 5).

$$l_k = \frac{k\,(L_x + 2a)}{r+1} - a \quad (4)$$

$$T_X = \frac{l_k}{L_x}\,\frac{l_k}{2} + \frac{(L_x - l_k)}{L_x}\,\frac{(L_x - l_k)}{2} \quad (5)$$

When Eq.1 and Eq.4 are embedded to the model formulation and the required calculations are completed, the average distance traveled in the horizontal axis of the warehouse for all the docks will be as shown in Eq. (6) (Onut et al ., 2008, s: 783–799).

$$T_X = -\frac{w_x\left(w_x + n_x\,(a_x + w_x/2)\right)}{2\,n_x\,(a_x + w_x/2)} + \frac{(n_x(a_x + w_x/2) + 2w_x)}{n_x(a_x + w_x/2)\,(r+1)^2} \sum_{d=1}^{D} d^2 \quad (6)$$

Unlike Onut et. al (2008), the following equations for the average vertical distance and the average distance in the height axis are proposed. The average vertical travel distances depends on the probability of the order that belonging the group i items. The formulation of the average travel distance in the vertical axis can be expressed following equation (7):

$$T_Y = w_Y + P_n \left(\sum_{i=1}^{n-1} a_y n_{Yi} + (\, n_{Yn}\, a_Y/2)\right) \quad (7)$$

The average travel distance in the height axis can be calculated in the following equation (8). The capacity of each column in the shelves will be different because of the different height of the pallets. In other words, the numbers of occupations along the z-axis for each group are not the same value.

$$T_Z = \sum_{i=1}^{r} a_{Zi}\, P_i\, z_i/2 \qquad (8)$$

To sum up all average travel distances, then to multiply the total travel distances by the unit material handling cost and by the yearly throughput of the warehouse, the total cost minimizing objective function will be as shown in Eq. (9)

$$C = D\, C_h \Big( w_Y + P_n \Big( \sum_{i=1}^{n-1} a_y n_{Yi} + (n_{Yn}\, a_Y/2) \Big) - \frac{w_x \left( w_x + n_x\,(a_x + w_x/2) \right)}{2\, n_x\,(a_x + w_x/2)} + \frac{(n_x(a_x + w_x/2) + 2w_x)}{n_x(a_x + w_x/2)(r+1)^2} \sum_{d=1}^{D} d^2 + \sum_{i=1}^{r} a_{Zi}\, P_i\, z_i/2 \Big) \qquad (9)$$

The constraints of the model are given in Eq. (10). The first line of model is the constraint of the number of the storage areas allocated to each group i items. *n* is equal to the number of the groups. The second one is the constraint of maximum distance of horizontal axis. This value, which is the total of each variables for allocating to group i items along the y-axis is smaller than the length of warehouse. The third constraint is the vertical constraint which limits the number of shelves along x-axis. The last one is the height constraint which restricts the maximum height value of each group.

$$\begin{aligned} & n_{Yi}\, n_{Zi}\, n_X \geq N_i && i = 1,2,..,n \\ & \sum_{i=1}^{i} n_{Yi}\; a_y < L_{Ymax} && i = 1,2,..,n \\ & a_x\, n_x < L_{Xmax} && \\ & a_{zi}\, n_{zi} < L_{Zmax} && i = 1,2,..,n \end{aligned} \qquad (10)$$

## 2. THE DESIGN METHODOLOGY OF MLWLP

### 2.1. Particle Swarm Optimization

One of the successful optimization algorithms is particle swarm optimization which is a population based search algorithm first proposed by Eberhart and Kennedy (1995) derived from the social behavior such as flocks of birds and schools of fish. Although PSO has many similarities with evolutionary techniques, the standard PSO does not use evolution operators such as crossover and mutation. PSO is easy to implement and it has few control parameters (Ting, Wu and Chou, 2014, s: 1543–1550).

PSO is a swarm intelligent algorithm in which the swarm consists of particles; each particle has its own position and velocity. The best solution is searched by collaborating between every individual (Zhu, 2009, s: 1231–1236). Each particle determines its movement through the search space by combining its best position and the best position their neighbors according to the fitness value (Poli, Kennedy and Blackwell, 2007, s: 33-57). In other words, the success of an individual particle in the swarm is affected not only by its own ‘experience’ but also the 'experience of neighboring particles. It is initialized with a random

D-dimensional particles group and then searches the solution space for the best position by updating velocity and positions (Sun, Zeng, Chu and Roddick, 2011, s: 124-128). The notations of the parameters for particle swarm optimization are given in Table 2. Also, Fig. 2 is given to explain particle swarm optimization algorithms' steps.

**TABLE 2: Nomenclatures for General PSO Algorithm**

| | |
|---|---|
| $v_i^k$ | Velocity of the *i*th particle in *k*th iteration |
| $x_i^k$ | Position of the *i*th particle in *k*th iteration |
| $p_i^k$ | Best position of the particle *i* in *k*th iteration |
| $g_i^k$ | Best position of the particle group in *k*th iteration |
| $S\ (x_i^k)$ | Solution of the *i*th particle in *k*th iteration |
| $c_1 ve\ c_2$ | Learning factors |
| $r_1 ve\ r_2$ | Random number from 0 to 1 |
| $w$ | nonnegative inertia factor |

Each individual's velocity changes and individual's position changes are updated by using Eq. (11). In fig.2, while improving solutions for all particles, these two equations are used for updating individual's velocity and individual's positions.

$$v_i^{k+1} = (w_k v_i^k + c_1 r_1 (p_i^k - x_i^k) + c_2 r_2 (g_i^k - x_i^k))$$
$$x_i^{k+1} = v_i^{k+1} + x_i^k \qquad (11)$$

**FIGURE 2: Algorithmic Schema for General PSO Algorithm**

Initialization (for k = 0)
  For i = 1 to N
    Generate particles randomly in solution space ($x_i^k$)
    Calculate initial solutions ($S$ ($x_i^k$))
    Assign $p_i^k$ = initial position ($x_i^k$)
    Assign $p_i^k$ = best position among the all particles
    Generate initial velocities randomly ($v_i^k$)
Improve the solution (for k = 1 to $iter_{max}$ )
  For i = 1 to N
    Update velocities ($v_i^{k+1}$)
    Modify the current positions ($x_i^{k+1}$)
  Calculate initial solutions ($S$ ($x_i^{k+1}$) )
  Update the best position of the *i*th particle ($p_i^{k+1}$)
  Update the best position of the particle group ($g_i^{k+1}$)
Finalize the algorithm
  (k = $iter_{max}$ )
    Assign the best solution = ($g_i^k$) and stop.

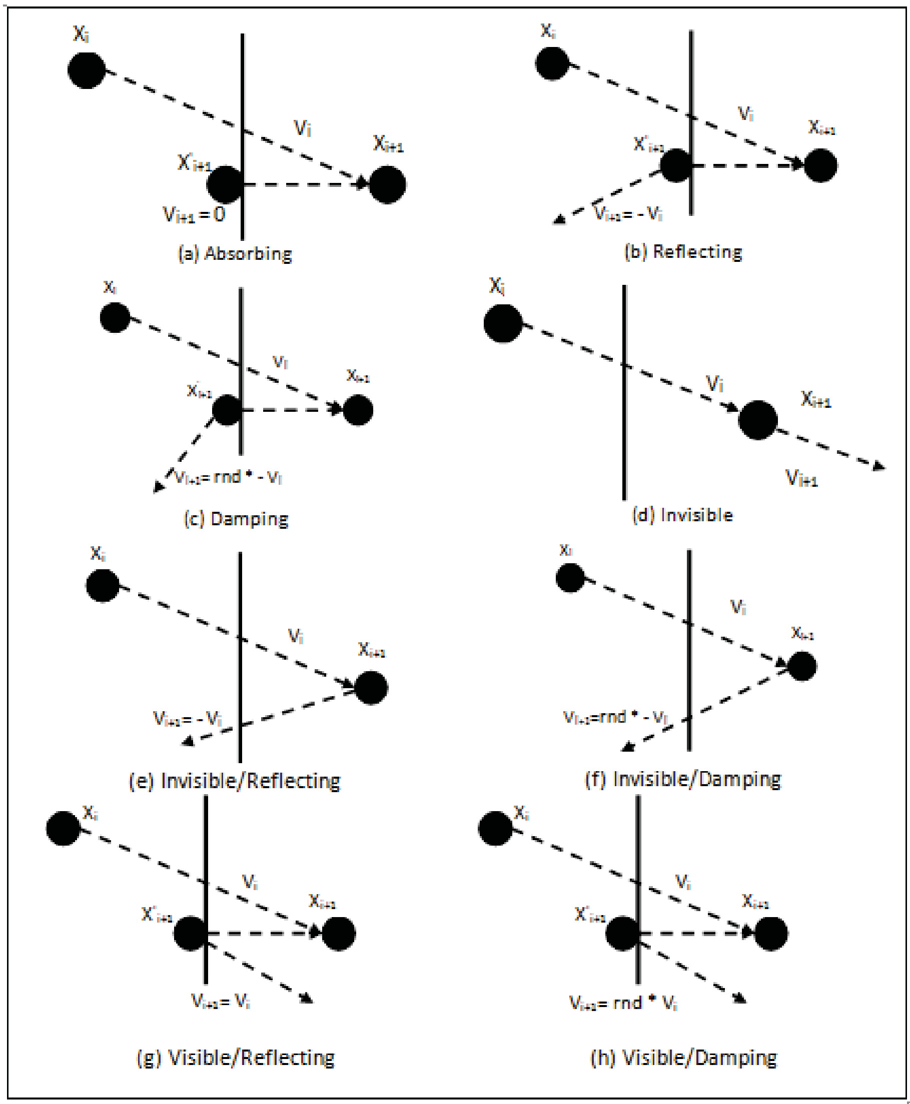
## 2.2. New Boundary Conditions for PSO

PSO is a heuristic optimization technique which has been applied to various optimization problems. However, it is not able to handle constrained optimization problems directly. There are some methods to manipulate particles' velocities or particles' positions for solving constrained problem. These are absorbing, reflecting, damping and invisible methods. Two other methods are also improved by using reflecting, damping and invisible methods (Xu and Rahmat-Samii, 2007; Li, Xe, Xie, Li, Zhou and Li, 2010). In this paper, two new boundary methods are suggested.

The absorbing method; when a particle moves outside the allowable solution space in one of the dimensions, it is relocated at the boundary of the solution space in that dimension, and the velocity in that dimension is zeroed.

Reflecting method has some similarity with absorbing method, but the difference is to change the sign of the velocity in that dimension. Damping method; when a particle moves outside the allowable solution space in one of the dimensions, it is relocated at the boundary and the velocity is reversed and multiplied with random number between 0 and 1. The invisible method; a particle is allowed to stay outside the solution space. The fitness value is not evaluated, and position and velocity of particle is not changed.

**FIGURE 3: Boundary Methods**




The Invisible/Reflecting method; a particle is allowed to stay outside the solution space; however, the fitness evaluation of that position is skipped and a bad fitness value is assigned to it. The sign of the velocity component in that dimension where the particle flies outside the boundary is changed (Xu and Rahmat-Samii, 2007).

The Invisible/Damping method has the same procedure as the Invisible/Reflecting method, but the difference is that velocity is also multiplied with random number between 0 and 1. For the last two methods, two new boundary conditions, when a particle moves outside the allowable solution space in one of the dimensions, it is relocated at the boundary of the solution space in that dimension, and the velocity of the particle is not manipulated in the

Visible/Reflecting method. The difference between Reflecting method and Visible/Reflecting method is not to change the sign of the velocity of the particle. For implementing the Visible/Damping method, when a particle moves outside the allowable solution space in one of the dimensions, the particle is relocated and the sign of the velocity is not changed like the Visible/Reflecting method. But the velocity of particle is also multiplied with random number between 0 and 1. Figure 3 visualizes the movement of particles.

### 2.3. New Assignment Algorithm for Particles' Initial Minimum Values of PSO

PSO can determine a function's minimum or maximum value between boundaries. However, a point which ensures the constraints is required on the first iteration. Otherwise, optimal value of the function cannot be determined or it can take long time to find the optimal value. In other words, if global best position or local best position cannot be determined, PSO is useless and does not find optimal value of the function. If the appropriate solution is determined, PSO can search to find better solution around that point.
Another difficulty of applying the standard PSO algorithm is the increasing of the dimensions of particles that can be affected to find the optimal solution of function. The allowable solution space will be expanded and particles will lose time to search inappropriate solutions and even cannot reach optimal solution.
In order to overcome these difficulties, an algorithm has been suggested to use these MLWLP which has different height of storage locations. The new assignment algorithm helps to reduce allowable solution spaces. By using this algorithm, all particles are distributed in constricted solution spaces. Thus, not only global best position or local best position can easily be determined but also particles cannot lose time to search inappropriate solutions. The algorithm is shown in the figure 4.

**FIGURE 4 :The New Assignment Algorithm**

1. Step: Equalities for each variable can be determined.

$$\mathrm{Ty_i} = \mathrm{n_{Yi}}\,\mathrm{a_Y}\left(\frac{\mathrm{P_i}}{2} + \sum_{\mathrm{i=i+1}}^{\mathrm{n}} \mathrm{Pi}\right) \qquad \mathrm{i} = 1,2,..\mathrm{n}$$

$$\mathrm{T_X} = -\frac{\mathrm{w_x}\,(\mathrm{w_x}+\mathrm{n_x}\,(\mathrm{a_x}+\mathrm{w_x}/2))}{2\,\mathrm{n_x}\,(\mathrm{a_x}+\mathrm{w_x}/2)} + \frac{(\mathrm{n_x}(\mathrm{a_x}+\mathrm{w_x}/2)+2\mathrm{w_x})^2}{\mathrm{n_x}(\mathrm{a_x}+\mathrm{w_x}/2)\,(\mathrm{r}+1)^2}\sum_{i=1}^{D}\mathrm{d}^2$$

$$\mathrm{T_{Zi}} = \mathrm{a_{Zi}}\mathrm{Pi}\,\mathrm{z_i}/2 \qquad \mathrm{i} = 1,2,..\mathrm{n}$$

2. Step: Slopes are calculated for all equalities by using determined equations.
3. Step: To compare the slopes of the all axes for each group.

$$\text{Transition X/ }\mathrm{Y_i} = \text{Slope }\mathrm{T_X}\text{/ Slope }\mathrm{T_{Yi}} \qquad \mathrm{i} = 1,2,..\mathrm{n}$$

$$\text{Transition X/ }\mathrm{Z_i} = \text{Slope }\mathrm{T_X}\text{/ Slope }\mathrm{T_{Zi}} \qquad \mathrm{i} = 1,2,..\mathrm{n}$$

$\text{Transition } Y_i / Z_i = \text{Slope } T_{Yi} / \text{Slope } T_{Zi} \quad i = 1,2,..n$

4. Step: Determine the average of all transitions.

$\overline{X/Y} = \sum_{i=1}^{n}(X/Y_i)/n \quad i = 1,2,..n$

$\overline{X/Z} = \sum_{i=1}^{n}(X/Z_i)/n \quad i = 1,2,..n$

$\overline{Y/Z} = \sum_{i=1}^{n}(Y_i/Z_i)/n \quad i = 1,2,..n$

5. Step: The average of the required storage spaces is equal to the 3th root of multiplied of *Average X/Y*, *Average X/Z* and *Average Y/Z*. The result is number of shelves along x-axis ($n_x$). If this value is greater than the maximum number of shelves along the x-axis ($nx_{\text{Max}}$), number of shelves along x-axis ($n_x$) is reduced to the maximum number of shelves ($nx_{\text{Max}}$).

$nx = (\overline{N_i} / (\overline{X/Y} * \overline{X/Z} * \overline{Y/Z}))\ \text{^}(1/3)$

$nx = \begin{cases} nx_{\text{Max}}, & \text{nx}_{\text{Max}} < nx \\ nx, & other \end{cases}$

6. Step: The required spaces of each group are divided by ($n_x$). These values are equal to the square root of $\text{Transition } Y_i / Z_i$ . The results are maximum number of the storage areas along the y-axis for group i items.

$nz_i = (N_i / (nx * \text{Transition } Y_i / Z_i))\ \text{^}(1/2)$

7. Step: The required spaces of each group divided by ($n_x$) times $nz_i$ equals to $ny_i$

$ny_i = (N_i / nx * nz_i)$

8. Step: If all results ensures the constraints (go to Step 11). If results do not ensure the constraints, the number of shelves along x-axis ($n_x$) is increased until to reach the maximum number of shelves (go to Step 6). If these numbers ($n_x$) equals to the maximum number of shelves (go to Step 9)

9. Step: If all results ensures the constraints (go to Step 11). If results do not ensure the constraints, number of the storage areas allocated to group i items along the z-axis is increased until to reach maximum number of the storage areas along the z-axis ($nz_i$) for group i items (go to Step 7). If these numbers ($nz_i$) equals to maximum number of the storage areas along the z-axis for group i items (go to Step 10)

10. Step: If all results ensures the constraints (go to Step 11). If results do not ensure the constraints, number of the storage areas allocated to group i items along the y-axis is increased until to reach maximum number of the storage areas along the y-axis ($ny_i$) for group i items (go to Step 7). If these numbers ($ny_i$) equals to maximum number of the storage areas along the y-axis for group i items

11. Step: If all results ensure the constraints, minimum initial values are obtained. These minimum initial values can be used for assigning the random initial values of the particles.

## 3. EXAMPLES OF MULTIPLE-LEVEL WAREHOUSE LAYOUT DESIGN PROBLEM

The formulated warehouse design model is applied to ten different distribution-type warehouse examples and the solutions are obtained by using the design methodology for MLWLP. Before applying the model, all items are categorized according to the turnover rate for all examples. In this phase, all storage products' groups are determined. The required numbers of storage locations for each group are calculated by using the turnover rate. For all illustrations, all items are separated into three groups which are group A, group B and group C. The accessibility to the warehouse door decreases from group A items to group B and C items. The throughput of the warehouse is 120,000 palletized products in a year. The storage system is back to back storage systems. The width of 1.1 m and length of 0.9 m for a storage space in a shelf is determined. The width of a pair of shelves is 2.2 meter and this value is used in the calculations. The width of sub-aisle is 2.0 m and the width of main aisle is 4 m. The material handling cost is 1.13 * $10^{-3}$ \$/m. The above-mentioned data are same for all examples (Onut et al., 2008).

**TABLE 3: The Data of Different Warehouse Examples**

| | Ex.1 | Ex.2 | Ex.3 | Ex.4 | Ex.5 | Ex.6 | Ex.7 | Ex.8 | Ex.9 | Ex.10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $L_{Xmax}$ | 42 | 31 | 31 | 42 | 52 | 52 | 63 | 63 | 63 | 52 |
| $L_{Ymax}$ | 36 | 23 | 23 | 27 | 36 | 32 | 32 | 32 | 32 | 32 |
| $L_{Zmax}$ | 14 | 14 | 14 | 11 | 11 | 11 | 9 | 8 | 12 | 10 |
| $P_A$ | 0.6 | 0.7 | 0.5 | 0.5 | 0.7 | 0.6 | 0.6 | 0.7 | 0.5 | 0.6 |
| $P_B$ | 0.3 | 0.2 | 0.3 | 0.3 | 0.2 | 0.3 | 0.3 | 0.2 | 0.3 | 0.3 |
| $P_C$ | 0.1 | 0.1 | 0.2 | 0.2 | 0.1 | 0.1 | 0.1 | 0.1 | 0.2 | 0.1 |
| $N_A$*1000 | 3 | 3,5 | 3,5 | 4 | 5,5 | 5 | 6 | 6 | 6,5 | 5,5 |
| $N_B$*1000 | 2 | 2,5 | 2,5 | 3 | 4,5 | 4 | 4 | 4,5 | 5 | 3,5 |
| $N_C$*1000 | 1 | 0,9 | 1,5 | 2 | 1,9 | 2 | 2 | 2,5 | 3 | 1,5 |
| $n_{Za\text{-}max}$ | 14 | 12 | 12 | 10 | 8 | 10 | 9 | 6 | 8 | 7 |
| $n_{Zb\text{-}max}$ | 14 | 14 | 10 | 9 | 7 | 11 | 8 | 8 | 9 | 7 |
| $n_{Zc\text{-}max}$ | 10 | 10 | 10 | 8 | 9 | 10 | 7 | 8 | 10 | 8 |
| $a_{Za}$ | 1 | 1.1 | 1.1 | 1.1 | 1.3 | 1.1 | 1 | 1.2 | 1.4 | 1.4 |
| $a_{Zb}$ | 1 | 1 | 1.3 | 1.2 | 1.4 | 1 | 1.1 | 1 | 1.3 | 1.3 |
| $a_{Zc}$ | 1.4 | 1.4 | 1.4 | 1.4 | 1.2 | 1.1 | 1.2 | 1 | 1.2 | 1.2 |

The differences between all examples are the probabilities of the groups, the size of the groups and the height of pallet for each group. The area where the examples warehouses will be constructed on, are different dimensions so the maximum size of warehouses also are not same for all examples. In other words, the maximum height of warehouse, the maximum number of shelves along the x-axis and y-axis cannot be equal. Table 3 shows the dimensions of the area, required number of storage spaces allocated to group i items, the height of the each pallet groups and the maximum height of the each groups.

## 4. DISCUSSION AND COMPUTATIONAL RESULTS

To evaluate the performance of the PSO with the new assignment algorithm and two new boundaries conditions, above-mentioned data are inserted to the model formulation as the parameters. PSO with assignment algorithm and standard PSO are coded by using MS Visual Studio package program and run on an Intel Core i3 CPU, 2.13 GHz with 4 GB RAM. Forty particles are used in each iteration and the maximum number of iterations is determined as 200. Both learning factors ($c_1$ $and$ $c_2$) are 2. The maximum velocity of particle is 2 and nonnegative inertia factor is 0.9.

PSO with the new assignment algorithm is applied for all examples. After that, the standard should be run for all examples in order to make a good consideration.

The PSO with the new assignment algorithm proposed in this study to overcome the difficulties of applying the standard PSO which are mentioned in section 3.3., on constricted models.

The new assignment algorithm is used to generate initial minimum values of the particles and its minimum initial values of particles are given in Table (4). These values are assigned minimum number of shelves along the x-axis, minimum number of the storage areas along the y-axis and z-axis. Thus, allowable solution space is constricted. The upper limits of the values of the particles are physical constraints of the construction area of the warehouse. The maximum height of the warehouse is determined either by height which is determined by authority or the chosen handling equipments' technical specification.

**TABLE 4: The Minimum Initial of Particles**

| | $n_X$ | $n_{Ya}$ | $n_{Yb}$ | $n_{Yc}$ | $n_{Za}$ | $n_{Zb}$ | $n_{Zc}$ |
|---|---|---|---|---|---|---|---|
| Example 1 | 20 | 9 | 8 | 8 | 9 | 7 | 4 |
| Example 2 | 15 | 10 | 8 | 5 | 12 | 11 | 6 |
| Example 3 | 15 | 10 | 9 | 6 | 12 | 10 | 9 |
| Example 4 | 20 | 10 | 9 | 11 | 10 | 9 | 5 |
| Example 5 | 25 | 14 | 13 | 9 | 8 | 7 | 5 |
| Example 6 | 25 | 10 | 10 | 9 | 10 | 8 | 5 |
| Example 7 | 30 | 12 | 9 | 9 | 9 | 8 | 4 |
| Example 8 | 30 | 17 | 10 | 7 | 6 | 8 | 6 |
| Example 9 | 27 | 16 | 11 | 8 | 8 | 9 | 7 |
| Example 10 | 25 | 16 | 10 | 8 | 7 | 7 | 4 |

# TABLE 5: The Results of PSO With Assignment Algorithm

| | | COSTS | | | | | | | | | CPU TIMES | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Invisible | Absorbing | Reflecting | Damping | Invisible/ Damping | Invisible /Reflecting | Visible/ Reflecting | Visible/ Damping | | Invisible | Absorbing | Reflecting | Damping | Invisible/ Damping | Invisible /Reflecting | Visible/ Reflecting | Visible/ Damping |
| 1st Example | BC | 15165 | 15165 | 15165 | 15165 | 15165 | 15165 | 15165 | 15165 | BT | 2 | 8 | 12 | 8 | 6 | 9 | 2 | 6 |
| | AVC | 15179 | 15179 | 15198 | 15185 | 15180 | 15184 | 15171 | 15199 | AVT | 19 | 22 | 21 | 22 | 23 | 23 | 19 | 19 |
| | SDC | 25 | 30 | 63 | 42 | 31 | 38 | 22 | 52 | SDT | 7 | 9 | 8 | 13 | 8 | 7 | 6 | 9 |
| | PHR | 57% | 77% | 70% | 73% | 60% | 67% | 87% | 63% | BIN | 13 | 48 | 74 | 47 | 31 | 55 | 16 | 34 |
| 2nd Example | BC | 16567 | 16567 | 16567 | 16567 | 16567 | 16567 | 16567 | 16567 | BT | 15 | 1 | 13 | 5 | 5 | 1 | 10 | 5 |
| | AVC | 16637 | 16624 | 16672 | 16660 | 16651 | 16635 | 16618 | 16627 | AVT | 16 | 11 | 14 | 12 | 21 | 16 | 16 | 15 |
| | SDC | 51 | 69 | 129 | 135 | 75 | 71 | 71 | 70 | SDT | 6 | 6 | 8 | 7 | 11 | 9 | 8 | 8 |
| | PHR | 3% | 13% | 13% | 27% | 17% | 23% | 27% | 10% | BIN | 56 | 7 | 79 | 30 | 28 | 6 | 55 | 30 |
| 3rd Example | BC | 18650 | 18650 | 18650 | 18650 | 18650 | 18650 | 18650 | 18650 | BT | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 |
| | AVC | 18653 | 18650 | 18653 | 18653 | 18661 | 18650 | 18650 | 18650 | AVT | 10 | 5 | 8 | 5 | 18 | 7 | 6 | 5 |
| | SDC | 14 | 0 | 9 | 8 | 23 | 0 | 0 | 0 | SDT | 8 | 7 | 7 | 4 | 11 | 6 | 5 | 5 |
| | PHR | 93% | 100% | 87% | 90% | 77% | 100% | 100% | 100% | BIN | 2 | 2 | 3 | 14 | 5 | 2 | 1 | 5 |
| 4th Example | BC | 20236 | 20735 | 20236 | 20236 | 20735 | 20236 | 20236 | 20236 | BT | 1 | 6 | 3 | 9 | 24 | 10 | 6 | 1 |
| | AVC | 20652 | 20940 | 20839 | 20837 | 20849 | 20663 | 20662 | 20675 | AVT | 18 | 2 | 11 | 7 | 24 | 16 | 12 | 10 |
| | SDC | 189 | 55 | 140 | 215 | 114 | 199 | 293 | 279 | SDT | 8 | 2 | 10 | 8 | 10 | 7 | 9 | 9 |
| | PHR | 17% | 0% | 3% | 10% | 0% | 17% | 30% | 27% | BIN | 11 | 35 | 20 | 55 | 138 | 57 | 38 | 3 |
| 5th Example | BC | 22905 | 22905 | 22905 | 22905 | 22905 | 22905 | 22905 | 22905 | BT | 1 | 5 | 1 | 2 | 11 | 6 | 3 | 1 |
| | AVC | 23013 | 22995 | 22985 | 22985 | 23038 | 22988 | 22979 | 22966 | AVT | 16 | 10 | 12 | 12 | 18 | 13 | 13 | 12 |
| | SDC | 60 | 64 | 67 | 67 | 83 | 77 | 70 | 68 | SDT | 9 | 6 | 8 | 9 | 7 | 7 | 7 | 9 |
| | PHR | 23% | 30% | 37% | 37% | 17% | 33% | 47% | 53% | BIN | 7 | 34 | 7 | 14 | 62 | 35 | 18 | 2 |
| 6th Example | BC | 20253 | 20253 | 20253 | 20253 | 20253 | 20253 | 20253 | 20253 | BT | 10 | 12 | 19 | 8 | 16 | 9 | 7 | 5 |
| | AVC | 20331 | 20311 | 20319 | 20305 | 20343 | 20308 | 20303 | 20306 | AVT | 22 | 15 | 18 | 15 | 23 | 20 | 21 | 18 |
| | SDC | 40 | 62 | 43 | 42 | 75 | 48 | 48 | 44 | SDT | 7 | 6 | 9 | 7 | 7 | 6 | 9 | 6 |
| | PHR | 20% | 30% | 13% | 27% | 23% | 43% | 47% | 33% | BIN | 61 | 70 | 111 | 50 | 94 | 55 | 40 | 34 |
| 7th Example | BC | 22555 | 22555 | 22555 | 22555 | 22555 | 22555 | 22555 | 22555 | BT | 13 | 4 | 9 | 5 | 25 | 11 | 5 | 7 |
| | AVC | 23128 | 23002 | 23208 | 23133 | 23213 | 23000 | 22881 | 22972 | AVT | 19 | 11 | 18 | 12 | 23 | 16 | 15 | 15 |
| | SDC | 230 | 751 | 767 | 717 | 135 | 323 | 332 | 673 | SDT | 7 | 5 | 7 | 7 | 7 | 7 | 7 | 7 |
| | PHR | 13% | 63% | 40% | 43% | 30% | 33% | 50% | 60% | BIN | 84 | 24 | 51 | 32 | 149 | 66 | 32 | 46 |
| 8th Example | BC | 25980 | 25980 | 25980 | 25980 | 25980 | 25980 | 25980 | 25980 | BT | 1 | 1 | 1 | 7 | 1 | 1 | 1 | 1 |
| | AVC | 26238 | 26408 | 26546 | 26570 | 26355 | 26230 | 26422 | 26330 | AVT | 14 | 8 | 7 | 6 | 15 | 9 | 8 | 9 |
| | SDC | 248 | 300 | 187 | 188 | 218 | 238 | 287 | 284 | SDT | 10 | 7 | 6 | 6 | 11 | 7 | 8 | 8 |
| | PHR | 47% | 30% | 7% | 7% | 23% | 47% | 27% | 37% | BIN | 4 | 2 | 7 | 41 | 4 | 2 | 7 | 10 |
| 9th Example | BC | 27011 | 27011 | 27011 | 27011 | 27011 | 27011 | 27011 | 27011 | BT | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | AVC | 27032 | 27063 | 27063 | 27080 | 27016 | 27013 | 27020 | 27063 | AVT | 11 | 9 | 12 | 10 | 16 | 12 | 11 | 11 |
| | SDC | 95 | 159 | 159 | 180 | 7 | 5 | 33 | 159 | SDT | 8 | 7 | 7 | 8 | 12 | 9 | 9 | 9 |
| | PHR | 77% | 90% | 90% | 87% | 70% | 90% | 93% | 90% | BIN | 5 | 4 | 5 | 10 | 4 | 9 | 9 | 4 |
| 10th Example | BC | 23914 | 23914 | 23914 | 23914 | 23946 | 23914 | 23914 | 23914 | BT | 14 | 9 | 20 | 9 | 24 | 18 | 7 | 7 |
| | AVC | 24196 | 24428 | 24502 | 24389 | 24535 | 24295 | 24248 | 24426 | AVT | 17 | 10 | 11 | 11 | 19 | 21 | 14 | 10 |
| | SDC | 234 | 337 | 260 | 324 | 206 | 349 | 329 | 322 | SDT | 8 | 10 | 9 | 8 | 11 | 10 | 9 | 9 |
| | PHR | 30% | 17% | 13% | 13% | 0% | 37% | 43% | 20% | BIN | 91 | 53 | 115 | 55 | 143 | 102 | 45 | 44 |

**TABLE 6: The Best Results of PSO With Assignment Algorithm**

| | 1st Example | 2nd Example | 3rd Example | 4th Example | 5th Example | 6th Example | 7th Example | 8th Example | 9th Example | 10th Example |
|---|---|---|---|---|---|---|---|---|---|---|
| BC | 15165 | 16567 | 18650 | 20236 | 22905 | 20253 | 22555 | 25980 | 27011 | 23914 |
| AVC | 15171 | 16618 | 18650 | 20662 | 22966 | 20303 | 23002 | 26230 | 27020 | 24248 |
| SDC | 22 | 71 | 0 | 293 | 68 | 48 | 751 | 238 | 33 | 329 |
| PHR | 87% | 27% | 100% | 30% | 53% | 47% | 63% | 47% | 93% | 43% |
| BT | 2 | 10 | 1 | 6 | 1 | 7 | 4 | 1 | 1 | 7 |
| AVT | 19 | 16 | 6 | 12 | 12 | 21 | 11 | 9 | 11 | 14 |
| SDT | 6 | 8 | 5 | 9 | 9 | 9 | 5 | 7 | 9 | 9 |
| BIN | 16 | 55 | 1 | 38 | 2 | 40 | 24 | 2 | 9 | 45 |
| Boundary Condition | Visible/ Reflecting | Visible/ Reflecting | (Absorbing) (Invisible /Reflecting) (Visible/ Reflecting) (Invisible/ Damping) | Visible/ Reflecting | Visible/ Damping | Visible/ Reflecting | Absorbing | Invisible /Reflecting | Visible/ Reflecting | Visible /Reflecting |

In each example, 8 boundary conditions for PSO with the new assignment algorithm are used to solve the problem. For each boundary condition for each example, the program runs on 30 times. The program was run on all examples and the results are shown in Table 5.

Each sample was evaluated in terms of the best cost (BC), the average cost (AVC), the standard deviation of cost (SDC), the percentage of hit rate (PHR), the best time (BT), the average time (AVT), the standard deviation time (SDT) and iteration number (BIN).

The percentage of hit rate provides the ratio between the number of runs yielded the optimum and the total numbers of experimental trials. These eight performance criteria are divided into two groups and then evaluated.

The first group consists of the best cost, the average cost, the standard deviation of cost and the percentage of hit rate. The second group consists of other performance criteria such as the best time, the average time, the standard deviation time and best iteration number.

Table 6 summarizes the results of PSO with the new assignment algorithm for all examples. It can be seen in Table 6, the best costs are between \$15.165 and \$27.011 and the average costs are between \$15.171 and \$27.020. The lowest standard deviation of costs is 0 and this value is obtained in the third example. The biggest standard deviation of costs is \$751 which is calculated in seventh example.

When evaluating the percentage of hit rate, the values of PHR are between 27% and %100. The one hundred percent hit rate is obtained in third example by four different boundary conditions. The worst percentage of hit rate is the result of the second example.

The best time is 1 ms and the worst time is 10 ms in all examples. The best average time of the best result is 6. The best iteration numbers are between 1 and 55.

For the third example, absorbing, visible reflecting, invisible reflecting and invisible damping boundary conditions have the same hit rates and the best costs. Seven of best solutions are obtained by the visible reflecting boundary condition. Two of the best solutions are found by invisible reflecting. Visible damping, invisible damping and absorbing boundary condition also found the best solution only once.

After applying PSO with the new assignment algorithm for all examples, the standard PSO should be run for all examples in order to make a good consideration. The standard PSO program was run on all examples and for each boundary condition for each example, the program runs on 30 times.

## TABLE 7: The Results of Standard PSO

| | | COSTS | | | | | | | | | CPU TIMES | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Invisible | Absorbing | Reflecting | Damping | Invisible/ Damping | Invisible /Reflecting | Visible/ Reflecting | Visible/ Damping | | Invisible | Absorbing | Reflecting | Damping | Invisible/ Damping | Invisible /Reflecting | Visible/ Reflecting | Visible/ Damping |
| 1st Example | BC | 15165 | 15165 | 15165 | 15165 | 15165 | 15165 | 15165 | 15165 | BT | 24 | 18 | 18 | 7 | 22 | 11 | 20 | 22 |
| | AVC | 15270 | 15278 | 15292 | 15228 | 15287 | 15242 | 15244 | 15289 | AVT | 21 | 24 | 20 | 23 | 25 | 22 | 23 | 22 |
| | SDC | 118 | 123 | 149 | 83 | 121 | 104 | 108 | 115 | SDT | 5 | 8 | 7 | 7 | 8 | 7 | 6 | 5 |
| | PHR | 7% | 20% | 13% | 27% | 10% | 7% | 13% | 13% | BIN | 155 | 180 | 101 | 38 | 125 | 52 | 112 | 125 |
| 2nd Example | BC | 16752 | 16670 | 16567 | 16884 | 16708 | 16912 | 16722 | 16830 | BT | 45 | 15 | 47 | 60 | 41 | 54 | 57 | 22 |
| | AVC | 17011 | 17125 | 17077 | 17160 | 17069 | 17160 | 17108 | 17061 | AVT | 35 | 34 | 38 | 35 | 40 | 36 | 32 | 41 |
| | SDC | 196 | 189 | 202 | 134 | 168 | 179 | 195 | 162 | SDT | 7 | 10 | 10 | 18 | 12 | 14 | 17 | 11 |
| | PHR | 0% | 0% | 3% | 0% | 0% | 0% | 0% | 0% | BIN | 171 | 41 | 173 | 170 | 146 | 186 | 163 | 58 |
| 3rd Example | BC | 18678 | 18677 | 19043 | 18650 | 19043 | 19092 | 18677 | 18677 | BT | 15 | 30 | 20 | 87 | 80 | 5 | 50 | 55 |
| | AVC | 18678 | 18677 | 19043 | 18650 | 19043 | 19092 | 18677 | 18677 | AVT | 15 | 30 | 20 | 87 | 80 | 5 | 50 | 57 |
| | SDC | 0 | 0 | 0 | 0 | 0 | 0 | 10 | 0 | SDT | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| | PHR | 0% | 0% | 0% | 3% | 0% | 0% | 0% | 0% | BIN | 25 | 55 | 30 | 125 | 90 | 10 | 60 | 60 |
| 4th Example | BC | 20836 | 20757 | 20887 | 20757 | 20928 | 20763 | 20730 | 20236 | BT | 32 | 3 | 42 | 23 | 15 | 32 | 40 | 19 |
| | AVC | 20964 | 21344 | 21306 | 21174 | 20981 | 21246 | 21398 | 21078 | AVT | 24 | 24 | 27 | 24 | 23 | 27 | 27 | 22 |
| | SDC | 134 | 512 | 315 | 387 | 53 | 390 | 393 | 658 | SDT | 13 | 12 | 13 | 13 | 7 | 7 | 12 | 7 |
| | PHR | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 3% | BIN | 173 | 16 | 189 | 102 | 65 | 175 | 141 | 108 |
| 5th Example | BC | 23759 | 23084 | 23198 | 23043 | 23011 | 23076 | 23230 | 23076 | BT | 21 | 32 | 25 | 18 | 21 | 15 | 12 | 13 |
| | AVC | 24291 | 23998 | 23932 | 23887 | 23823 | 24076 | 23968 | 23935 | AVT | 19 | 24 | 17 | 17 | 24 | 28 | 23 | 23 |
| | SDC | 660 | 528 | 500 | 469 | 441 | 476 | 449 | 477 | SDT | 6 | 9 | 8 | 7 | 5 | 6 | 10 | 9 |
| | PHR | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | BIN | 140 | 177 | 139 | 92 | 121 | 84 | 67 | 75 |
| 6th Example | BC | 20429 | 20283 | 20486 | 20283 | 20253 | 20353 | 20253 | 20353 | BT | 32 | 12 | 25 | 12 | 24 | 26 | 36 | 11 |
| | AVC | 21323 | 21542 | 21615 | 21271 | 21437 | 21497 | 21573 | 21436 | AVT | 23 | 23 | 20 | 25 | 23 | 22 | 26 | 21 |
| | SDC | 735 | 863 | 1084 | 743 | 681 | 824 | 933 | 800 | SDT | 8 | 9 | 8 | 7 | 10 | 10 | 10 | 9 |
| | PHR | 0% | 0% | 0% | 0% | 3% | 0% | 3% | 0% | BIN | 198 | 69 | 115 | 67 | 132 | 139 | 199 | 64 |
| 7th Example | BC | 23125 | 22555 | 23255 | 23222 | 23442 | 23222 | 23230 | 23214 | BT | 28 | 27 | 24 | 7 | 11 | 37 | 20 | 66 |
| | AVC | 24948 | 24626 | 24468 | 24850 | 24905 | 24515 | 24617 | 24732 | AVT | 22 | 22 | 24 | 24 | 28 | 24 | 23 | 28 |
| | SDC | 870 | 1110 | 1020 | 1095 | 882 | 979 | 862 | 990 | SDT | 8 | 11 | 9 | 10 | 8 | 10 | 10 | 11 |
| | PHR | 0% | 7% | 10% | 0% | 0% | 0% | 0% | 0% | BIN | 178 | 152 | 134 | 40 | 60 | 198 | 109 | 29 |
| 8th Example | BC | 26721 | 26772 | 26721 | 26721 | 27266 | 25980 | 26705 | 26802 | BT | 33 | 37 | 37 | 37 | 37 | 38 | 13 | 25 |
| | AVC | 26721 | 26772 | 26721 | 26721 | 27266 | 26351 | 26705 | 26802 | AVT | 33 | 37 | 37 | 37 | 37 | 29 | 13 | 25 |
| | SDC | 0 | 0 | 0 | 0 | 0 | 523 | 0 | 0 | SDT | 0 | 0 | 0 | 0 | 0 | 12 | 0 | 0 |
| | PHR | 0% | 0% | 0% | 0% | 0% | 3% | 0% | 0% | BIN | 110 | 197 | 110 | 110 | 110 | 188 | 75 | 135 |
| 9th Example | BC | 27735 | 27141 | 27792 | 27092 | 28435 | 27890 | 27532 | 28337 | BT | 33 | 1 | 15 | 20 | 11 | 30 | 22 | 33 |
| | AVC | 28106 | 27772 | 28396 | 28519 | 28905 | 28354 | 28392 | 28665 | AVT | 15 | 18 | 25 | 15 | 23 | 30 | 19 | 20 |
| | SDC | 322 | 891 | 499 | 668 | 570 | 360 | 426 | 303 | SDT | 11 | 23 | 9 | 11 | 8 | 2 | 11 | 11 |
| | PHR | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | BIN | 198 | 7 | 84 | 111 | 61 | 167 | 124 | 180 |
| 10th Example | BC | 25276 | 24418 | 24671 | 24394 | 25647 | 24638 | 24386 | 24394 | BT | 9 | 31 | 14 | 26 | 17 | 19 | 34 | 15 |
| | AVC | 25462 | 24903 | 24847 | 24665 | 25761 | 25143 | 24557 | 24734 | AVT | 12 | 23 | 23 | 24 | 21 | 18 | 28 | 26 |
| | SDC | 262 | 450 | 174 | 250 | 161 | 713 | 204 | 228 | SDT | 4 | 9 | 8 | 6 | 5 | 2 | 7 | 7 |
| | PHR | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | BIN | 55 | 175 | 79 | 144 | 98 | 101 | 179 | 85 |

**TABLE 8: The Best Results of Standard PSO**

| | 1st Example | 2nd Example | 3rd Example | 4th Example | 5th Example | 6th Example | 7th Example | 8th Example | 9th Example | 10th Example |
|---|---|---|---|---|---|---|---|---|---|---|
| BC | 15165 | 16567 | 18650 | 20236 | 23011 | 20253 | 22555 | 25980 | 27092 | 24386 |
| AVC | 15228 | 17077 | 18650 | 21078 | 23823 | 21437 | 24626 | 26351 | 28519 | 24557 |
| SDC | 83 | 202 | 0 | 658 | 441 | 681 | 1110 | 523 | 668 | 204 |
| PHR | 27% | 3% | 3% | 3% | 0% | 3% | 7% | 3% | 0% | 0% |
| BT | 7 | 47 | 87 | 19 | 21 | 24 | 27 | 38 | 20 | 34 |
| AVT | 23 | 38 | 87 | 22 | 24 | 23 | 22 | 29 | 15 | 28 |
| SDT | 7 | 10 | 0 | 7 | 5 | 10 | 11 | 12 | 11 | 7 |
| BIN | 38 | 173 | 125 | 108 | 121 | 132 | 152 | 188 | 111 | 179 |
| Boundary Condition | Damping | Reflecting | Damping | Visible/ Damping | Invisible/ Damping | Invisible/ Damping | Absorbing | Invisible /Reflecting | Damping | Visible /Reflecting |

The best results of the standard PSO for ten examples without changing the initial values are shown in Table 7. Table 8 summarizes the best results of the standard PSO.

As seen in Table 8, the best hit rate of this algorithm is 27% which is obtained in the first example. For the first example, the best solution, obtained by the given in Table 8, is damping boundary condition for PSO. The lowest hit rate is 0% which is obtained in fifth, ninth and tenth examples. The hit rate of fifth example, ninth example and tenth example show that the standard PSO could not find the optimal solutions for these examples. For the third example, the result is obtained only in one of the thirty runs, and the value is an optimal result.

The biggest standard deviation value is 1110 in the seventh example and the lowest standard deviation value is 0 in the third example.

The worst time is 87 ms and the best time is 7 ms in the all examples. The best average time of the best result is 22. The best iteration numbers are between 38 and 188.

Three of best solutions are obtained by the damping boundary condition. Two of best results are found by the invisible damping boundary condition. The reflecting boundary condition, the absorbing boundary condition, visible damping boundary condition, the visible reflecting boundary condition and the invisible reflecting boundary condition also found the best solution only time.

As a result, PSO with the new assignment algorithm not only improves on the hit rate but also helps to find optimal solutions for each example. By using the assignment algorithm, both the average cost and the standard deviation of cost of PSO algorithm with the new assignment algorithm are lower than the standard PSO's. The reasons of the low standard deviation are the higher hit rates and the lower average costs.

PSO with the new assignment algorithm's iteration numbers and solutions are lower than the standard PSO's.

Considering the results of the best time, the average time, the standard deviation of time and the best iteration number, PSO with the new assignment algorithm has higher successes according to the results of standard PSO's.

To conclude, Table 6 and Table 8 show that PSO algorithm by using the new assignment algorithm has many advantages. This algorithm gives not only better results, but also makes more hits and solves in better time at all boundary conditions.

For each example, the number of shelves (a pair of back to back shelves), the storage spaces in each shelf for all groups, the height of the each groups and the total cost of the material handling are shown in the table (9).

**TABLE 9: The Results of Examples**

| | $n_X$ | $n_{Ya}$ | $n_{Yb}$ | $n_{Yc}$ | $n_{Za}$ | $n_{Zb}$ | $n_{Zc}$ | Cost ($) | Length (m) | Width (m) |
|---|---|---|---|---|---|---|---|---|---|---|
| Ex.1 | 9 | 12 | 8 | 8 | 14 | 14 | 7 | 15.165 | 37,8 | 25,2 |
| Ex. 2 | 12 | 13 | 8 | 4 | 12 | 14 | 10 | 16.567 | 50,4 | 22,5 |
| Ex. 3 | 15 | 10 | 9 | 6 | 12 | 10 | 9 | 18.650 | 63 | 22,5 |
| Ex. 4 | 17 | 12 | 10 | 8 | 10 | 9 | 8 | 20.236 | 71,4 | 27 |
| Ex. 5 | 21 | 17 | 16 | 7 | 8 | 7 | 7 | 22.905 | 88,2 | 36 |
| Ex. 6 | 16 | 16 | 12 | 7 | 10 | 11 | 9 | 20.253 | 67,2 | 31,5 |
| Ex. 7 | 21 | 16 | 12 | 7 | 9 | 8 | 7 | 22.555 | 88,2 | 31,5 |
| Ex.8 | 28 | 18 | 11 | 6 | 6 | 8 | 8 | 25.980 | 117,6 | 31,5 |
| Ex. 9 | 25 | 17 | 12 | 6 | 8 | 9 | 10 | 27.011 | 105 | 31,5 |
| Ex. 10 | 22 | 18 | 12 | 5 | 7 | 7 | 7 | 23.914 | 92,4 | 31,5 |

## CONCLUSION

The multi-level warehouse layout problem with different z-axis storage spaces is not investigated in the literature. This study aims to suggest the methodology of designing a multiple-level warehouse which minimizes the total material handling costs by considering different number of the storage areas allocated to different groups along the z-axis. According to placed groups, shelf heights and shelf capacities may vary. In other words, groups having different pallet heights may have different heights of shelves and different capacities of the shelves.

Nonlinearity in the variables and the constraints make the designing problems difficult to solve within a short time. So as to overcome difficulty, PSO algorithm is applied for such designing problems. The first contribution of the PSO algorithm in this study is, to struggle with the constraints using the new two boundary conditions for finding an optimal solution. Most of the examples' best solutions are found in two new boundary conditions. By using the new boundary conditions, the particles are searching solutions along the boundary of constraints.

The second contribution is that, the solution space is constricted by the new assignment algorithm to find the optimal solution within a short time. The PSO particles could search optimal solution in a constricted solution space and also can easily find optimal solution. Moreover, by using the new assignment algorithm, the hit rate of PSO algorithm can be increased and the iteration number and time can be decreased.

Furthermore, the length and the width of multiple-level warehouse examples are found by using boundary conditions for PSO algorithm. In the model, not only the area of the warehouse is determined, but also the rack system of warehouse is defined. While designing the multiple-level warehouse, the handling costs are considered in the calculation.

In this study, the number of a dock is taken from decision makers. For further research, the number of docks can be determined by using simulation or heuristic methods. Besides, vehicle waiting costs and idle times of material handling systems can be considered while determining the number of docks by again simulation or heuristic methods. When designing the multiple-level warehouse, construction cost and material handling equipment costs are excluded from equations. For future directions these costs can be added to the existing study.

## REFERENCES

Ashayeri, J. & Gelders, L.F., (1985), “Warehouse design optimization”, **European Journal of Operational Research**, 21, pp.285–294.

Baker, P. & Canessa, M., (2009), “Warehouse design: A structured approach”, **European Journal of Operational Research**, 193(2), pp.425–436.

Cakmak, E., Gunay, S.N., Aybakan G., Tanyas M., (2012), “Determining the Size and Design of Flow Type and U-Type Warehouses”, **Procedia - Social and Behavioral Sciences**, 58, pp.1425–1433.

Caron F., Marchet G. & Perego A. (2000), “Layout Design in Manual Picking Systems: A Simulation Approach”, **Integrated Manufacturing System**, 94-104.

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.

Gu, J., Goetschalckx, M. & Mcginnis, L., (2007), “Research on warehouse operation: A comprehensive review”, **European Journal of Operational Research**, 177(1), pp.1–21.

Gu, J., Goetschalckx, M. & McGinnis, L.F., (2010), “Research on warehouse design and performance evaluation: A comprehensive review”, **European Journal of Operational Research**, 203(3), pp.539–549.

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), pp.327–338.

Huertas, J.I., Ramírez, J.D. & Salazar, F.T., (2007), “Layout evaluation of large capacity warehouses”, **Facilities**, 25(7/8), pp.259–270.

Lai, K.., Xue, J. & Zhang, G., (2002), “Layout design for a paper reel warehouse: A two-stage heuristic approach”, **International Journal of Production Economics**, 75(3), pp.231–243.

Lambert, D.M., Stock, J.R., Ellram, L.M. (Eds.), (1998), **Fundamentals of Logistics Management**, McGraw-Hill, Singapore. pp.268.

Larson, T.N., March, H. & Kusiak, A., (1997), “A heuristic approach to warehouse layout with class-based storage”, **IIE Transactions**, 29(4), pp.337–348.

Le-Duc, T. & De Koster, R. (2005), “Travel distance estimation and storage zone optimization in a 2-block class-based storage strategy warehouse”, **International Journal of Production Research**, 43, 3561-3581.

Li, H., Xe, H., Xie, X., Li, L., Zhou, J., Li X., (2010), "A New Boundary Condition for Particle Swarm Optimization", **Journal of Converge Information Technology**, 5(9), pp.215–221.

Macro, J.G. & Salmi, R.E., (2002), "A Simulation Tool to Determine Warehouse Efficiencies and Storage Allocations", In **Proceedings of the 2002 Winter Simulation Conference.** pp. 1274–1281.

Onut, S., Tuzkaya, U., Dogac, B., (2008), "A particle swarm optimization algorithm for the multiple-level warehouse layout design problem", **Computers & Industrial Engineering**, 54(4), pp.783–799.

Park, Y.H. & Webster, D.B., (1989), "Modelling of three-dimensional warehouse systems" **International Journal of Production Research**, 27(6), pp.985–1003.

Poli, R., Kennedy, J., Blackwell,T., (2007), "Particle Swarm Optimization An Overview", **Swarm Intelligence**, 1, pp. 33-57.

Queirolo, F., Tonelli, F., Schenone, M., Nan, P., Zunino, I., (2002), "Warehouse Layout Design: Minimizing Travel Time with A Genetic And Simulative Approach - Methodology And Case Study", In **Proceedings 14th European Simulation Symposium**.

Roodbergen, K.J., (2001), "Layout and routing methods for warehouses", Ph.D. thesis, RSM Erasmus University, the Netherlands.

Roodbergen K.J. & Vis F.A., (2006), "A model for warehouse layout", **IIE Transactions**, 38, 799-811.

Rosenblatt M.J., Roll Y., (1984), "Warehouse design with storage policy considerations", **International Journal of Production Research**, 22(5), pp.809-821.

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(3), pp.515–533.

Sun, C., Zeng J., Chu, S., Roddick, F.R., (2011), "Solving Constrained Optimization Problems by an Improved Particle Swarm Optimization", In **Second International Conference on Innovations in Bio-inspired Computing and Applications**, 35, pp.124-128.

Ting C., Wu K., Chou H., (2014), "Particle Swarm Optimization Algorithm for The Berth Allocation Problem", **Expert Systems with Applications**, 41(4), pp.1543–1550.

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

Xu, S. & Rahmat-Samii, Y., (2007), “Boundary Conditions in Particle Swarm Optimization Revisited”, **IEEE Transactıons on Antennas and Propagatıon**, 55(3), pp.760–765.

Yang, L., Sun Y., (2004), “Expected Value Model for A Fuzzy Random Warehouse Layout Problem”, In **IEEE International Conference Proceedings on Fuzzy Systems**. pp. 751–756.

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), pp.731–744.

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), pp.413–425.

Zhang, G.Q. & Lai, K.K., (2010), “Tabu search approaches for the multi-level warehouse layout problem with adjacency constraints” **Engineering Optimization**, 42(8), pp.775–790.

Zhu, J., (2009), “A Modified Particle Swarm Optimization Algorithm” **Journal of Computers**, 4(12), pp.1231–1236.