Int J Adv Manuf Technol (2012) 60:787–797
DOI 10.1007/s00170-011-3593-y

ORIGINAL ARTICLE

Storage location assignment and order picking optimization in the automotive industry

Seval Ene · Nursel Öztürk

Received: 20 November 2010 / Accepted: 3 June 2011 / Published online: 30 August 2011

Abstract The objective of this study is to design storage assignment and order picking system using a developed mathematical model and stochastic evolutionary optimization approach in the automotive industry. It is performed in two stages. At the first stage, storage location assignment problem is solved with a class-based storage policy with the aim of minimizing warehouse transmissions by using integer programming. At the second stage, batching and routing problems are considered together to minimize travel cost in warehouse operations. A warehouse in the automotive industry is analyzed, and an optimum solution is obtained from an integer programming model. Due to the computational time required for solving the integer programming problem, a faster genetic algorithm is also developed to form optimal batches and optimal routes for the order picker. The main advantage of the algorithm is the quick response to production orders in real-time applications. The solutions showed that the proposed approach based on genetic algorithms can be applied and integrated to any kind of warehouse layout in automotive industry.

Keywords Order picking systems · Class-based storage · Routing · Genetic algorithms

1 Introduction

In the competitive atmosphere of today, successful supply chain management increases the competitive power of firms. One of the most important elements of the supply chain is the warehouse. Warehouses serve as a buffer in balancing out supply and demand. Products can be temporarily stocked in the warehouse, and customer or production demands can be supplied by picking parts from the relevant locations [1]. Approximately 20% of the money spent on logistics in the private sector goes to warehouse operations [2]. Among these operations, order picking is the highest-cost process because it is labour-intensive and it requires repetition [3, 4]. Studies have shown that the order picking operation makes up 60% of total warehouse costs, and even today, order-picking operations are still performed manually. Order pickers spend more than 50% of their time travelling between picking locations and the I/O point [5]. One method for minimizing these periods is to design a completely new warehouse. However, it is also possible to minimize these periods using less radical methods, such as by changing operational processes [1].

The efficiency of the order-picking process depends on factors such as storage systems, location and control mechanisms. Generally, there are four approaches for increasing the efficiency of the order-picking process to minimize the order picking time or to minimize the distance travelled in the warehouse. The first approach for minimizing travel time is to plan the order-picking route. The second approach is to separate the warehouse into zones. Thus, order pickers will only pick orders within the zone to which they have been assigned. The third approach is to store items in locations chosen according to the best usage of the shelves. Here, the relationship between the storage assignment rule and the routing method is important. The final approach is to batch the orders, which minimizes the distance travelled by picking the orders assigned to the same batch all in one go [1]. Among these methods, storage assignment, order batching and routing are considered the most important.

In this paper, storage assignment and order picking system is presented using a developed mathematical model and stochastic evolutionary optimization approach for the

S. Ene (✉) · N. Öztürk
Industrial Engineering Department, Faculty of Engineering and Architecture, Gorukle Campus, Uludag University,
16059 Bursa, Turkey
e-mail: sevalene@uludag.edu.tr

N. Öztürk
e-mail: nursel@uludag.edu.tr

automotive industry. This study is performed in two stages. At the first stage, storage location assignment problem is solved with a class based storage policy with the aim of minimizing warehouse transmissions. Locations are assigned using The General Algebraic Modeling System (GAMS) [6]. At the second stage, optimum solution of order batching and picker routing problem is obtained from integer programming model. Due to the computational time required for solving the integer programming problem, a faster genetic algorithm is also developed to form optimal batches and optimal routes for the order picker.

2 Literature review

Many techniques have been applied to storage assignment, order batching and routing problems. Berg and Zijm [7] presented the storage systems and warehouse management problems and the hierarchy of decision problems of storage systems in the stage of designing, planning and controlling. Petersen and Aaese [8] used a simulation model to compare principles for picking, storage and routing in manual order-picking systems. They examined the effects of the quantity and distribution of the order and the structure of the warehouse using sensitivity analyses and showed that batching is much more profitable if the orders are small in size. Chen and He [9] presented warehouse assignment strategies for storage systems with automation and developed a mathematical model for warehouse assignment optimization. In order to overcome big-sized problems, they have solved warehouse assignment problem with the method of particle swarm optimization that depends on Pareto optimal solution. Muppani and Adil [10] examined a system that includes a class-based storage method. They developed the simulated annealing algorithm for the integer programming that was made up for solving storage assignment and forming the classes. Muppani and Adil [11] developed a nonlinear integer programming for class-based storage arrangement taking into consideration area decrease, handling costs and storage area costs. They used the branch and bound algorithm for solving the developed nonlinear model. Le Duc and De Koster [12] developed a model presuming the average order picking travelling distance in a storage system in which a class-based storage is used and developed a mathematical model for warehouse zoning optimization problem using this average order picking travelling period as an objective function. Hsu, Chen and Chen [13] developed a batching approach based on genetic algorithms (GAs), which directly minimize the total distance travelled. Tsai, Liou and Huang [14] solved the optimal batch picking problems by using multiple GAs. Ho and Tseng [15] examined different batching methods that were formed according to the seed-order selection rule and accompanying order selection rules. Ho, Su and Shi [16] focused on seed-order selection and the accompanying order selection rules in a two-aisled order-picking warehouse. Hwang and Kim [17] developed an efficient order-batching algorithm that uses cluster analysis on each of the methods of s-shape, return and midpoint routing. In their studies, Caron, Marchet and Perego [18] compared different routing strategies in the order picking system in terms of approximate travelling time to the low level parts picker. Hwang et al. [19] developed analytical models for approximating the total distance using whether on the aisle number is one or dual for the s-shape, return and midpoint routing methods. Roodbergen and De Koster [1] presented heuristic methods for order picking routes in warehouses where two or more cross aisles exist and random storage is used. Won and Olafsson [20] developed a sequence that handles the batching and sequencing problems together for batching and order picking problems.

This study is different from previous studies because storage assignment, batching and routing problems are studied together instead of studying individually as in literature. Besides batching and routing, optimization is performed in one algorithm. In this paper, storage assignment and batching and routing were optimized together with the aim of minimizing warehouse handling for a specific warehouse in the automotive industry. It is performed in two stages. At the first stage, storage location assignment problem is solved with a class-based storage policy by using integer programming. Differently from previous studies, the disadvantage of class-based storage that it requires more storage space is overcome with random storage assignments in classes. At the second stage, optimum solution to the order-batching and picker-routing problem is obtained by genetic algorithms and genetic representation of solutions are encoded through order locations differently from previous studies. The genetic algorithm was coded using the C# programming language. The distance between locations in 3D warehouse is calculated via functions proposed in this paper, instead of approximate distances. The remainder of this paper is organized as follows. Section 3 presents the problem definition. Section 4 presents the mathematical model, which is developed for storage assignment. The proposed integer programming model and genetic algorithm for order batching and routing are given in Section 5. In Section 6, experimental results are analyzed and discussed. Section 7 summarizes the results of this study.

3 Problem definition

In this paper, a manual warehousing system in the automotive industry and the picker to part order picking system is studied. Figure 1 shows the warehouse studied in

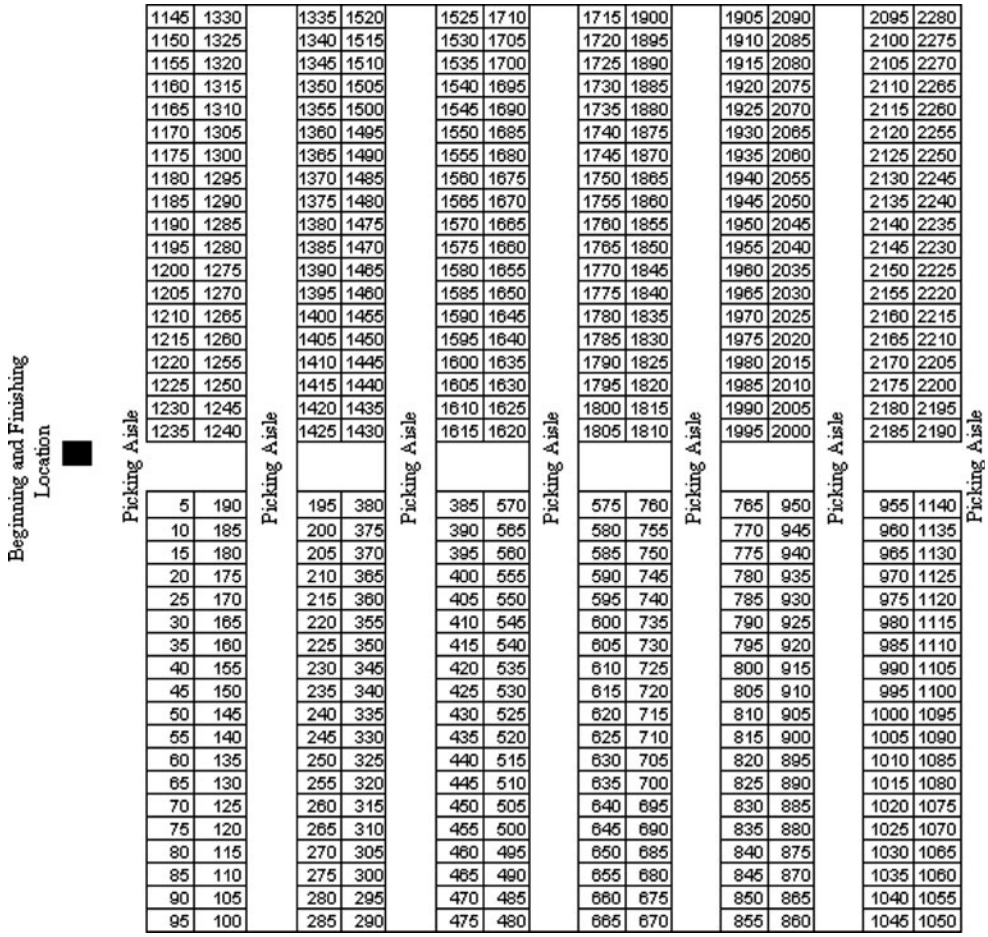


Fig. 1 3D warehouse studied in this paper

this paper. There are two blocks and seven picking aisles in the assembly supply warehouse. Except for the first and the last aisles, all aisles have 76 picking arrays, and all arrays have five layers. In the aggregate, the warehouse has 2,280 picking locations. Orders coming from the production line can be automatic or manual. It is known that all orders are accumulated in an order pool and data as order arrival time, order due time, order storage location is joint to the orders. The capacity of the order-picking vehicle is constant, and each batch cannot exceed this capacity.

In the present case, the closest open location and random storage are used together for storage assignment strategy. However, material arrival rates and order rates of materials are not considered in this strategy. Just some specific materials that have higher order rates are cumulated to aisles without addressing to locations. For these reasons, this strategy causes a chaotic structure in the warehouse. Only, the order arrival time and related production line are taken into account for batching and routing. Locations and routing procedures are not considered for picking lists. These are distinct problems seen in the warehouse. Therefore, the proposed approach is introduced to solve the above mentioned problems related to storage and order picking for the warehouse.

4 Storage assignment optimization

In this paper, the proposed storage assignment optimization for automotive assembly supply warehouses includes four steps:

Step 1 Determination of the material classes
Step 2 Development of the mathematical formulations for distances from locations to warehouse input door and to order pool
Step 3 Development of the mathematical model
Step 4 Application of different scenarios for automotive assembly supply warehouse

4.1 Material classes

Studies on class-based storage assignment policy show that better results are achievable with three classes. In this paper, order rates have been considered on the determining of the material classes. Three classes are determined according to order rates; class A, class B, and class C. Order rates are calculated as follows:

$$\text{Order Rate} = \frac{\text{Average daily consumption}}{\text{Case volume(unit)}} \qquad (1)$$

These classes will have an important role in determining assembly materials' weights for storage location assignment.

4.2 Proposed mathematical formulations for distances

The main objective of this study is to minimize total travel time for storing and order picking in assembly supply warehouse. On these grounds, distance computation is a critic issue in optimization. Horizontal and vertical distances are considered on developing of the mathematical formulations for distances. We use the following notations:

- i *1*,*2*,…,*I* for shelf columns
- j *1*,*2* for direction of shelf
- k *1*,*2*,…,*K* for the numbers of shelf division
- l *1*,*2*,…,*L* for shelf layers

The mathematical formulations for horizontal distances from locations to input door are:

$$D^1_{i,j,k} = \begin{cases} j = 1 \Rightarrow \alpha + \min\{(k \times \beta), (|2\gamma - k| \times \beta)\} + [(\delta - i) + 1] \times (\psi + \omega) \\ j = 2 \Rightarrow \alpha + \min\{(k \times \beta), (|2\gamma - k| \times \beta)\} + (\delta - i) \times (\psi + \omega). \end{cases} \tag{2}$$

The mathematical formulations for horizontal distances from locations to order pool are:

$$D^2_{i,j,k} = \begin{cases} j = 1 \Rightarrow |k - \gamma| \times \beta + (i - 1) \times (\psi + \omega) \\ j = 2 \Rightarrow |k - \gamma| \times \beta + (i \times \omega) + (i - 1) \times \psi. \end{cases} \tag{3}$$

The mathematical formulation for vertical distances from locations to order pool or locations to input door is:

$$D^3_l = l \times \varepsilon \tag{4}$$

The mathematical formulation for travel time from input door to locations is:

$$t^1_{i,j,k,l} = \frac{D^1_{i,j,k}}{v_h} + \frac{D^3_l}{v_v} \tag{5}$$

The mathematical formulation for travel time from locations to order pool is:

$$t^2_{i,j,k,l} = \frac{D^2_{i,j,k}}{v_h} + \frac{D^3_l}{v_v} \tag{6}$$

where,

- α Horizontal distance between shelf columns and I/O point
- β Horizontal distance between shelf divisions
- γ Location number in a shelf column layer
- δ Shelf column number in a warehouse block
- ψ Horizontal distance between shelf columns
- ω Width of a shelf column
- ε Vertical distance between shelf layers
- V_{h} Horizontal speed of the order-picking vehicle (m/s)
- V_{v} Vertical speed of the order-picking vehicle (m/s)

4.3 Proposed model

At first stage of this study, the mathematical model of the storage location assignment optimization for an automotive factory was developed. The aim of our model is to minimize total time for storing and order picking with consideration of assembly materials' class weights. The following notations are used in this model.

Indices:

- p *1*,*2*,…,*P* for cases of assembly materials
- i *1*,*2*,…,*I* for shelf columns
- j *1*,*2* for direction of shelf
- k *1*,*2*,…,*K* for the numbers of shelf division
- l *1*,*2*,…,*L* for shelf layers

Parameters:

- w^1_p Weighting factor for travel time from input door to storage location
- w^2_p Weighting factor for travel time from storage location to order pool
- $t^1_{i,j,k,l}$ Travel time from input door to storage location (s)
- $t^2_{i,j,k,l}$ Travel time from storage location to order pool (s)

$$0 \le w^1_p \le 1,\ 0 \le w^2_p \le 1$$

Decision Variables:

$$x_{p,i,j,k,l} = \begin{cases} 1 \text{ if case } p \text{ assigns to } i\text{th shelf column, } j\text{th shelf direction, } k\text{th shelf and } l \text{ shelf layer} \\ 0 \text{ Otherwise.} \end{cases}$$

Using the above indices, parameters and decision variables, the model is proposed as follows:

$$\min\ z=\sum_p\sum_l\sum_k\sum_j\sum_i w_p^1 x_{p,i,j,k,l}\ \ t_{i,j,k,l}^1+\sum_p\sum_l\sum_k\sum_j\sum_i w_p^2 x_{p,i,j,k,l}\ \ t_{i,j,k,l}^2 \tag{7}$$

Subject to

$$\sum_p x_{p,i,j,k,l} \quad \le 1 \quad \forall(i,j,k,l) \tag{8}$$

$$\sum_l\sum_k\sum_j\sum_i x_{p,i,j,k,l} \quad = 1 \quad \forall p \tag{9}$$

$$x_{p,i,j,k,l} \quad \in \{0,1\} \tag{10}$$

Constraint 8 specifies that number of cases assigned to each storage location cannot exceed 1. Constraint 9 states that each case must be assigned to only one location specified with indices. Constraint 10 specifies that decision variable x must be 0 or 1. Objective function (7) minimizes total travel time from input door to storage locations and travel time from storage locations to order pool with consideration of assembly materials class' weights.

5 Order batching and routing optimization

After performing storage assignment decisions, at the second stage of this study an integer programming model is proposed for batching and routing problem and optimal solution for small scaled problems obtained. The formulation for distance is as follows:

$p=1,2,\ldots..1140$ Location Numbers
$l=1,2,\ldots..1140$ Location Numbers

$$\begin{aligned}DH_{p,l}=\min\left([\mathrm{mod}(|p-l|,\varphi)/\eta]\times\beta,((\gamma-[\mathrm{mod}(|p-l|,\varphi)/\eta])\times\beta)\right)\\ +floor(|p-l|/\varphi)\times\psi\end{aligned} \tag{11}$$

$$DV_{p,l}=(\mathrm{mod}(p,\eta)+\mathrm{mod}(l,\eta))\times\varepsilon \tag{12}$$

$$D_{p,l}=DH_{p,l}+DV_{p,l} \tag{13}$$

The formulation for travel time is as follows:

$$t_{p,l}=\frac{DH_{p,l}}{v_h}+\frac{DV_{p,l}}{v_v} \tag{14}$$

where,

φ	Total location number in a shelf column
η	Layer number in every shelf column
γ	Location number in a shelf layer
β	Horizontal distance between shelf divisions
ψ	Horizontal distance between shelf columns
ε	Vertical distance between shelf layers
v_h	Picking vehicle's horizontal speed (m/s)
v_v	Picking vehicle's vertical speed (m/s)
$DV_{p,l}$	Vertical distance between location p and location l (m)
$DH_{p,l}$	Horizontal distance between location p and location l (m)

The problem environment mentioned above can be formulated mathematically as follows:

Index:

$p:1,2,\ldots\ldots\ldots,L$ for locations
$l:1,2,\ldots\ldots\ldots,L$ for locations
$b:1,2,\ldots\ldots\ldots,B$ for batches

Decision Variables:

$$x_{b,p}=\begin{cases}1 & \text{if the order of location } p \text{ is assigned to batch } b\\ 0 & \text{otherwise}\end{cases}$$

$$y_{p,l}^b=\begin{cases}1 & \text{if location } l \text{ is visited immediately after location } p \text{ in batch } b\\ 0 & \text{otherwise}\end{cases}$$

Decision variables determine the assignment of order locations to batches and the route of order pickers for each batch.

Parameters:

Cap Picking vehicle's capacity (units)
$t_{p,l}$ Travel time from location p to l (s)
n Number of orders

The objective function minimizes travel time for order picking operations.

$$\min z = \sum_{b=1}^{B}\sum_{p=1}^{L}\sum_{l=1}^{L} y_{p,l}^{b} t_{p,l} \tag{15}$$

$$\sum_{p=1}^{L} x_{b,p} \leq \text{Cap} \quad \forall b \tag{16}$$

$$\sum_{b=1}^{B} x_{b,p} = 1 \quad \forall p \tag{17}$$

$$x_{b,BL} = 1 \quad \forall b \tag{18}$$

$$x_{b,FL} = 1 \quad \forall b \tag{19}$$

$$\sum_{l=1}^{L} y_{p,l}^{b} = x_{b,p} \quad p \neq l \tag{20}$$

$$\sum_{l=1}^{L} y_{l,p}^{b} = x_{b,p} \quad p \neq l \tag{21}$$

$$u(p) - u(l) + n\sum_{p=1}^{L}\sum_{l=1}^{L} y_{p,l}^{b} + (n-2)\sum_{p=1}^{L}\sum_{l=1}^{L} y_{p,l}^{b} \leq n-1 \quad p \neq l \tag{22}$$

$$y_{l,p}^{b}, x_{b,p} \in \{0,1\} \tag{23}$$

Constraint 16 states that the number of orders assigned to a batch must not exceed the picking vehicle's capacity. Constraint 17 ensures that each order will be assigned to exactly one batch. Constraints 18 and 19 specify that all batches must include the beginning and ending locations, respectively. Constraint 20 states that assignments from location p to l in a batch must be equal to the batch's order number. Constraint 21 states that assignments from location l to p in a batch must be equal to the batch's order number. Constraint 22 is a sub-tour elimination constraint. Constraint 23 specifies that decision variables y and x must be 0 or 1.

This model is coded in GAMS [6], and an optimal solution for small-scaled problems is obtained.

However, the use of integer programming model in warehouse operations for exact solutions can be impractical and requires great computation time. Therefore, heuristic approaches are applied to batching and routing problems to get solutions quickly and frequently. One of these heuristics is genetic algorithms.

5.1 Genetic algorithms

Genetic algorithms were first introduced by Holland, who was inspired by the notion of natural and biological evolution. In genetic algorithms, concepts inspired by population genetics and evolution theory are used to construct the optimization algorithms. They attempt to optimize the fitness of a population of elements through recombination and mutation of genes. To apply the genetic evolutionary concept to a real-world optimization problem, two issues must be addressed: encoding the potential solutions and defining the fitness function (objective function) to be optimized.

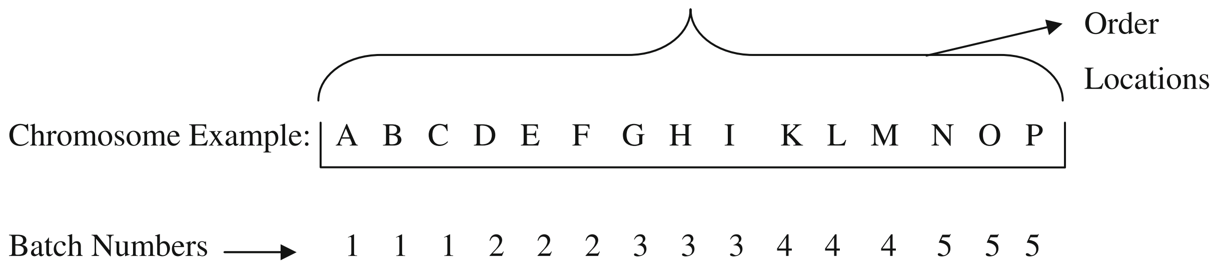
A solution, namely a chromosome, is encoded as a string composed of several components (genes). The initial population of chromosomes is either generated according to some principles or selected randomly. The algorithm performs an evaluation to measure the quality (fitness) of the potential solutions. Optimization using Genetic Algorithms is achieved by (a) selecting pairs of chromosomes with probabilities proportionate to their fitness and (b) matching them to create new offspring. In addition to matching (crossover), small mutations are induced in new offspring. The replacement of bad solutions with new solutions is based on some fixed strategies. The chromosomes evolve through successive iterations, called generations. The evaluation, optimization and replacement of solutions are repeated until the stopping criteria are satisfied [21].

5.2 Proposed algorithm

In this section, a genetic algorithm is proposed to solve the order batching and routing problem together in all kinds of warehouse layouts. Genetic algorithm is selected to take the advantage of evolutionary optimization approach. In the initial stage of designing GA, order data of production for a month's period is analyzed to determine chromosome length. It is discovered that running the algorithm for 5 min's period is enough to fulfil orders and avoid tardiness. There is an average of 15 orders in each 5-min period, and consequently, 15 genes in a chromosome. The proposed algorithm, which is designed for simultaneous batching and routing in all kinds of warehouse layouts, is as follows:

Step 1 Define a genetic representation for a feasible solution of the problem.

The genetic representation for the problem solution is encoded through a string composed of orders locations, which differs from the representation used in previous studies. For example, the chromosome (A, B, C, D, E, F, G, H, I, K, L, M, N, O, P) means that order locations A, B and C are assigned to batch 1; D, E and F to batch 2; G, H and I to batch 3; K, L and M to batch 4; N, O and P to batch 5.



Step 2 Create an initial population. Set t=0

$$PO(0) = x_1^0, x_2^0, x_3^0, \ldots\ldots.., x_N^0 \quad N: \text{Population size}$$

Step 3 Calculate fitness function $f(x_i^t)$ for all chromosomes in the population.

The fitness function is defined in terms of total travel time to pick orders in a batch that have the same line number and arrival time interval. By minimizing the total distance travelled, and accordingly the total travel time, we can minimize the cost of warehouse operations.

Step 4 Compute the probability of each chromosome in the population for the reproduction process.

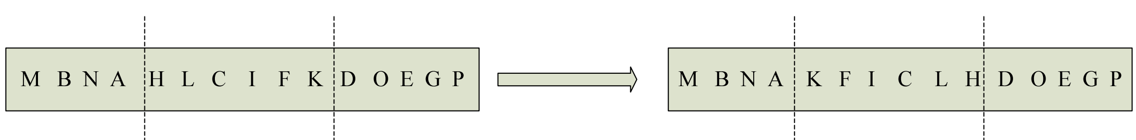
In the selection for reproduction, the roulette wheel selection mechanism is used. The probability is calculated as follows:

$$P(x_i^t) = \frac{f(x_i^t)}{\sum f}$$

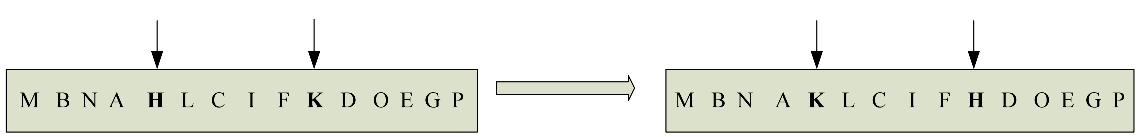
$P(x_i^t) \times N$: Reproduction number of related chromosome

Step 5 Apply crossover and mutation.

The crossover mechanism used in this study is reverse action, in which the genes between two randomly chosen points are reversed in order.



In the mutation mechanism, two randomly chosen points in the chromosome are swapped with each other.



The structure of the chromosome is considered in the selection of mutation and crossover mechanisms.

Step 6 If the stopping criteria have been reached, then stop. Otherwise, go to Step 3 and set t=t+1

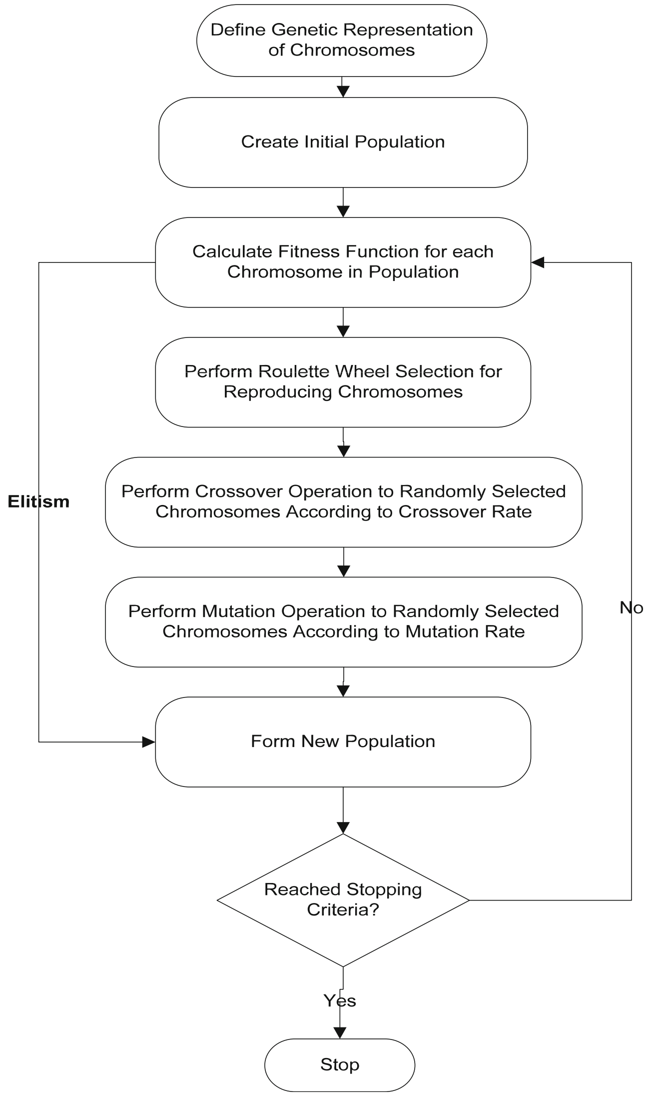


Fig. 2 Genetic algorithm flow chart

Figure 2 shows flow chart of genetic algorithm.

6 Experimental results and discussion

The system designed for the warehouse optimization system will work as shown in Fig. 3. First, incoming products will be assigned to warehouse locations by using storage assignment model. Orders from the production line will be accumulated in an order pool. By using this order pool to obtain all order information, such as arrival time, due time and order location, the algorithm will be run in a dedicated time interval. When results are obtained from the algorithm, the order picker will fulfil the order list.

Table 1 Weighting factors for material classes

Material classes	Weighting factor *w1*	Weighting factor *w2*
Class A	0.1	0.9
Class B	0.3	0.7
Class C	0.4	0.6

6.1 Storage assignment optimization results

Assignment Model given in Section 4.3 was coded using GAMS and run through different scenarios for automotive assembly supply warehouse. Under the present circumstances, it is assumed that some storage locations are full. Optimal solutions were computed for each scenario. Among these scenarios, the most realistic one is the one that uses daily data. The weighting factors (for travel time from input door to storage location: *w1* and weighting factor for travel time from storage location to order pool: *w2*) are derived after a set of experiments for scenario 1. Accepted weighting factors after the experiments are given in Table 1.

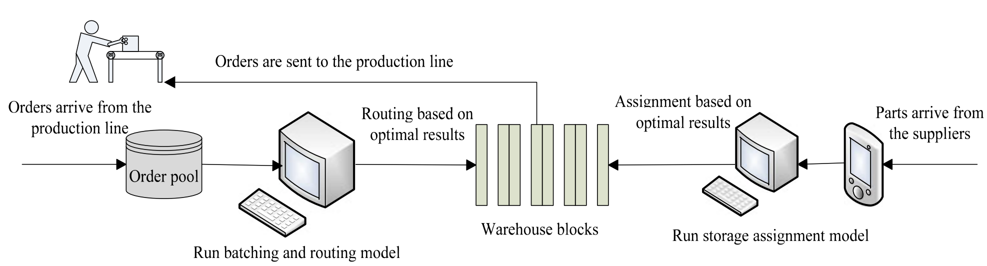
The program was run with weighting factors given in Table 1 for assembly supply warehouse daily data and optimum storage assignment results are obtained as shown in Fig. 4.

According to results presented in Fig. 4, material case with reference number R102 of class A is assigned to locations nearby each other and order pool. These results are obtained based on material class weights.

6.2 Order batching and routing optimization results

To evaluate the performance of the proposed genetic algorithm designed for the batching and routing problem, actual data are used for testing. An example dataset is presented in Table 2.

Fig. 3 Proposed warehouse optimization system



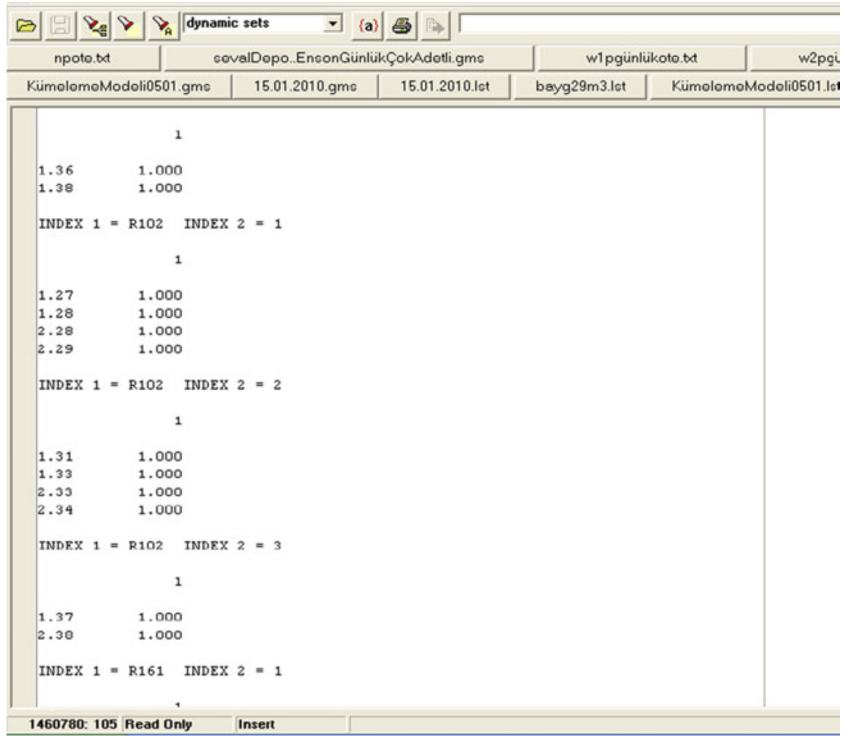


Fig. 4 Storage assignment results

Table 2 shows that this order dataset is from the 3rd production line and comes from the time interval 17:00–17:12 hours. Orders are requested, on average, 1 h prior to their due times. Due times are distributed in the interval 18:00–18:05 hours. The requested orders from the production line are always for one product.

The parameters of a genetic algorithm—crossover rate, mutation rate, elitism rate and population size—also affect the effectiveness of the algorithm. Parameter settings are derived by four factor experimental design. Three levels are determined for all parameters to get optimum picking time. Experiment summary is presented in Table 3.

Results of experimental design for critical parameters to obtain optimum picking time are presented in Table 4. The crossover rate is set to 0.7, the mutation rate to 0.008, population size to 125 and elitism rate to 0.04.

Table 5 shows batching and picker routing results derived from the proposed genetic algorithm using the example dataset. The algorithm is run at a 5-min due time interval. For purposes of batching, we must restrict the size of each batch to three items, the capacity of the picking vehicle.

Table 6 shows computational results of proposed genetic algorithm obtained from different datasets.

Table 2 Order information for an example dataset

Order number	Arrival time	Due time	Order location	Production line number of order
1	17:00	18:00	96	3
2	17:01	18:01	272	3
3	17:02	18:01	188	3
4	17:02	18:03	67	3
5	17:03	18:03	23	3
6	17:04	18:02	41	3
7	17:06	18:02	315	3
….	….	….	….	….
14	17:11	18:05	150	3
15	17:12	18:05	215	3

7 Conclusions

In order picking systems, storage assignment and batch picking are approaches aimed at minimizing order picker's

Table 3 Experiment summary

	Crossover rate	Mutation rate	Elitism rate	Population size
1	0.6	0.008	0.02	75
2	0.6	0.009	0.03	100
3	0.6	0.01	0.04	125
4	0.7	0.008	0.04	100
5	0.7	0.009	0.02	125
6	0.7	0.01	0.03	75
7	0.8	0.008	0.03	125
8	0.8	0.009	0.04	75
9	0.8	0.01	0.02	100

Table 4 Parameter settings for genetic algorithm

Parameters	Genetic algorithm
Crossover rate	0.7
Mutation rate	0.008
Elitism rate	0.04
Population size	125

travel time. Many methods have been developed by researchers. In this paper, these problems are discussed and optimized together for a specific warehouse in the automotive industry. In storage location assignment, storage location proximity is considered for obtaining exact solutions to minimize total travel time in an assembly supply warehouse. The distance between locations in a 3D warehouse is calculated using functions proposed in this paper, rather than approximate distances. We present an integer programming formulation that optimizes the storage assignment. Proposed model coded by GAMS and tested using data of an automotive factory. Two main factors: time interval and storage location proximity considered to get exact solutions for batching and routing problem in the warehouse. First, we represent an integer programming formulation that optimizes batching and routing problem together. However, due to the need for short computation time in real world problems, we also developed a genetic algorithm to approximate the results. Differently from previous studies genetic representation of solutions in Genetic Algorithm, designed in this paper, are encoded through order locations. The distance between locations in 3D warehouse is calculated via functions proposed in this paper, rather than approximate distances. Algorithm coded in C # programming language. The main advantage of the algorithm is the quick response to production orders in real-time applications. The solutions showed that the proposed approach based on GAs can be applied and integrated to any kind of warehouse layout in automotive industry. This paper demonstrated that developed models can be used by similar automotive manufacturers' warehouses effectively to reduce their supply chain costs.

Table 5 Batching and routing results for dataset example

Batching		Routing
Batch number	Items in batch	Picking route
1	3	272-78-67
2	3	215-41-66
3	3	315-122-96
4	3	150-175-23
5	3	200-12-188
Total Picking Time 858 s		

Table 6 Computational results for different datasets

Datasets	Number of orders	Number of batches formed	Picking time (s)	CPU time (s)
DS1	15	5	858	13
DS2	21	7	1694	40
DS3	30	10	2611	47
DS4	45	15	4619	54

Acknowledgement The authors thank TOFAŞ-FIAT Automotive Factory for providing data for the problem.

References

1. Roodbergen KJ, De Koster R (2001) Routing Methods for Warehouses with Multiple Cross Aisle. Int J Prod Res 39 (9):1865–1883
2. ELA/AT Kearney (2004) Excellence in logistics. ELA, Brussels
3. Hwang H, Baek W, Lee M (1988) Cluster Algorithms for Order Picking in an Automated Storage and Retrieval System. Int J Prod Res 26:189–204
4. Rana K (1991) Order-Picking in Narrow-Aisle Warehouse. Int J Phys Distrib Logist Manage 20(9):9–15
5. Tompkins JA, White JA, Bozer YA, Frazelle EH, Tanchoco JMA, Trevino J (1996) Facilities Planning. Wiley, New York
6. GAMS Software GmbH Eupener Str. 135–137. 50933 Cologne, Germany
7. Berg JP, Zijm WHM (1999) Models for Warehouse Management: Classification and Examples. Int J Prod Econ 59:519–528
8. Petersen CG, Aase G (2004) A Comparison of Picking, Storage and Routing Policies in Manual Order Picking. Int J Prod Econ 92:11–19
9. Chen Y, He F (2008) Research on particle swarm optimization in location assignment optimization. Proceedings of the 7th World Congress on Intelligent Control and Automation
10. Muppani VR, Adil GK (2008a) Efficient Formation of Storage Classes for Warehouse Storage Location Assignment: A Simulated Annealing Approach. Int J Manag Sci 36:609–618
11. Muppani VR, Adil GK (2008b) A Branch and Bound Algorithm for Class-Based Storage-Location Assignment. Eur J Oper Res 189:492–507
12. Le-Duc T, De Koster R (2005) Travel Distance Estimation and Storage Zone Optimisation in a 2-Block Class-Based Storage Strategy Warehouse. Int J Prod Res 43(17):3561–3581
13. Hsu CM, Chen KY, Chen MY (2005) Batching Orders in Warehouses by Minimizing Travel Distance with Genetic Algorithms. Comput Ind 56:169–178
14. Tsai CY, Liou JJH, Huang TM (2008) Using a Multiple-GA Method to Solve the Batch Picking Problem: Considering Travel Distance and Order Due Time. Int J Prod Res 46 (22):6533–6555
15. Ho YC, Tseng YY (2006) A Study on Order-Batching Methods of Order-Picking in a Distribution Centre with Two Cross-Aisles. Int J Prod Res 44(17):3391–3417

16. Ho YC, Su TS, Shi ZB (2008) Order Batching Methods for an Order-Picking Warehouse with Two Cross Aisles. Comput Ind Eng 55:31–347
17. Hwang H, Kim DG (2005) Order-Batching Heuristics Based on Cluster Analysis in a Low-Level Picker-to-Part Warehousing System. Int J Prod Res 43(17):3657–3670
18. Caron F, Marchet G, Perego A (1998) Routing Policies and COI Based Storage Policies in Picker-to Part Systems. Int J Prod Res 36(3):713–732
19. Hwang H, Oh YH, Lee YK (2004) An Evaluation of Routing Policies for Order-Picking Operations in Low-level Picker-to Part System. Int J Prod Res 42(18):3873–3889
20. Won J, Olafsson S (2005) Joint Order Batching and Order Picking in Warehouse Operations. Int J Prod Res 43(7):1427–1442
21. Goldberg DE (1989) Genetic algorithm in search. Optimization and Machine Learning. Addison-Wesley, New York