Management Science Letters 3 (2013) 603–612

Contents lists available at GrowingScience

Management Science Letters

homepage: www.GrowingScience.com/msl

Scheduling trucks in cross docking systems with temporary storage and dock repeat truck holding pattern using genetic algorithm

Ehsan Ghobadian[a*], Reza Tavakkoli-Moghaddam[b], Mahdi Naderi-Beni[a] and Hassan Javanshir[a]

[a]*Department of Industrial Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran*
[b]*Department of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran*

C H R O N I C L E

Article history:
Received October 15, 2012
Received in revised format
4 December 2012
Accepted 8 December 2012
Available online
December 9 2012

Keywords:
GRASP
Genetic algorithm
Temporary storage
Metaheuristics
Cross docking

A B S T R A C T

Cross docking is one of the most important issues in management of supply chains. In cross docking, different items delivered to a warehouse by inbound trucks are directly arranged and reorganized based on customer demands, routed and loaded into outbound trucks for delivery purposes to customers without virtually keeping them at the warehouse. If any item is kept in storage, it is normally for a short amount of time, say less than 24 hours. In this paper, we consider a special case of cross docking where there is temporary storage and implements genetic algorithm to solve the resulted problem for some realistic test problems. In our method, we first use some heuristics as initial solutions and then improve the final solution using genetic algorithm. The performance of the proposed model is compared with alternative solution strategy, the GRASP method.

1. Introduction

Cross docking plays an important role in supply chain management and there have been growing interests in this problem under different conditions (Barbarosoglu & Ozgur, 1999; Ghobadian et al., 2012). In cross docking, a warehouse is managed to deliver various items to various customers based on their demands. In cross docking, all commodities are routed and loaded into outbound trucks for delivery to customers with minimum waiting time at the warehouse. If any item is kept in storage, it is normally for a short amount of time, say less than 24 hours. In cross docking, the turnaround times for customer orders, inventory management expenditure, and warehouse space requirements are tried to be minimized.

Cross docking problem is modeled as a mixed integer programming, which is also categorized as NP-Hard problem (Feo & Resende, 1989; Mosheiov, 1998). Therefore, we need to implement some metaheuristics to solve such problem. Rohrer (1995) investigated cross docking problem and the

Corresponding author. Tel:+989123887038 *
E-mail: Ehsan_ghob@yahoo.com (E. Ghobadian)

doi: 10.5267/j.msl.2012.12.009

implementation of simulation techniques to solve such problem. Yu (2002) studied cross docking problem under variety assumptions. Yu and Egbelu (2008) explained some forms of cross docking or scheduling sequence for both inbound and outbound trucks through minimization of total operation time when a temporary storage buffer was considered at shipping dock. The product assignment to trucks and the docking sequences of the inbound and outbound trucks were all calculated, accordingly.

Vahdani and Zandieh (2010) investigated the implementation of five meta-heuristic algorithms including Tabu search (TS), genetic algorithm (GA), simulated annealing (SA), electromagnetism-like algorithm (EMA) and variable neighborhood search (VNS) to schedule the trucks in cross-dock systems for minimization of total operation time by considering a temporary storage buffer to hold temporarily items at the shipping dock and implemented response surface methodology (RSM) to tune their problem parameters. They also considered two types of objective functions to develop multiple objective decision making model. Vahdani et al. (2009) considered another cross docking problem where it was a scheduling the truck holdover recurrent dock cross-dock problem using robust meta-heuristics.

Soltani and Sadjadi (2010) proposed two different hybrid meta-heuristics namely hybrid simulated annealing and hybrid variable neighborhood search, to solve cross docking problem to reach the best possible sequence of truck pairs. They implemented various sample test problems to investigate the performance of the proposed methods, especially for large-sized problems. Nascimento et al. (2010) discussed the independent multi-plant, multi-period, and multi-item capacitated lot-sizing problem where transfers among various plants were allowed. They developed a Greedy Randomized Adaptive Search Procedure (GRASP) heuristic as well as a path-relinking intensification procedure to determine cost-effective solutions and proposed some heuristics to find some instances of the capacitated lot sizing problem with parallel machines. The results of the computational tests demonstrated that the proposed heuristics could outperform other heuristics previously explained in the literature. Boloori Arabani et al. (2011) developed some other types of meta-heuristics implementation for scheduling of trucks in a cross-docking system with temporary storage. Fig. 1 shows a sample of cross docking system.

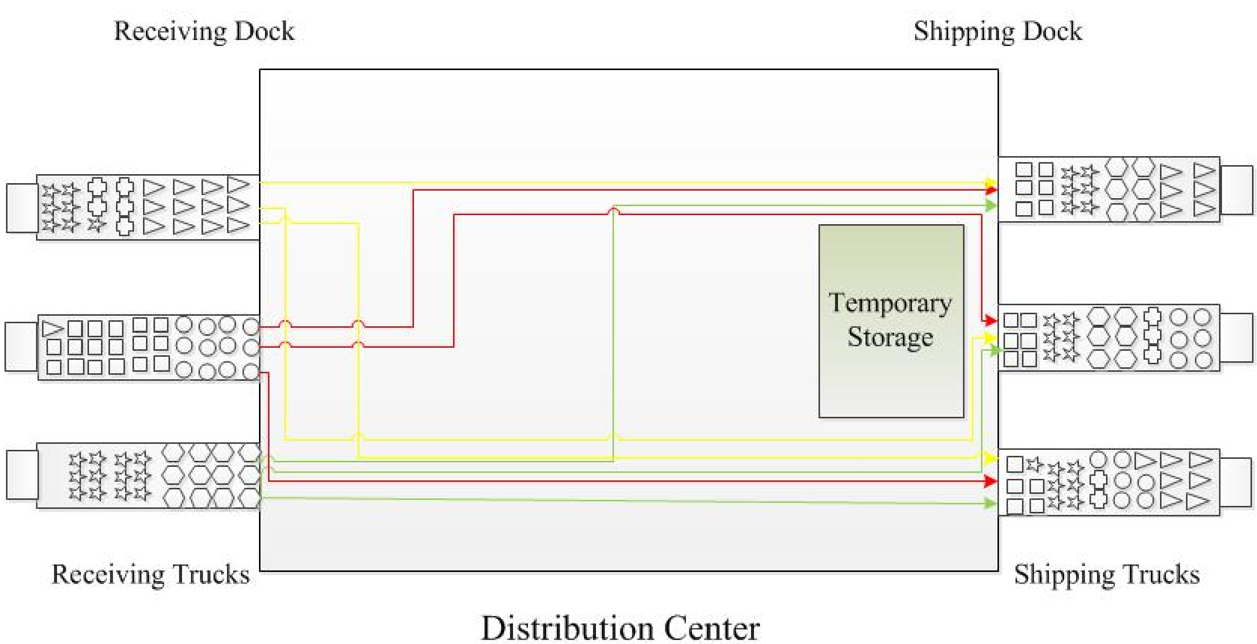

Fig .1. Cross docking distribution center (Ghobadian et al., 2012)

Yu (2002) presented another model by assuming that there was a temporary storage in cross docking system and each two groups of trucks of receiving and shipping of loaded could alternatively enter into cross-dock. The temporary storage permitted trucks to deliver more cargos, which are storage for future shipments. The other trucks, which are responsible for shipping cargos to final destinations could also use this temporary storage to meet final customers' needs.

In this paper, we reconsider cross docking problem earlier investigated by Ghobadian et al. (2012) and present genetic algorithm (GA) to find near optimal solution. We also compare the performance of GA of this paper with GRASP algorithm developed earlier. In this paper, we have adopted all notations and problem formulation earlier presented by Ghobadian et al. (2012).

2. The proposed method

Continuous Variables:

T Makespan,
U_{ij} Time at which the variable t_{ij} transferring receiving truck i to shipping truck j starts to unload from receiving truck i onto the receiving dock,
L_{ij} Time at which the variable t_{ij} transferring from receiving truck i to shipping truck j finished loading from the shipping dock into shipping truck j,

Integer Variables:

x_{ijk} Number of units of product type k which transfer from receiving truck i to shipping truck j,
t_{ij} Total number of units of products which transfer from receiving truck i to shipping truck j, where $\left(t_{ij} = \sum_{K=1}^{N} x_{ijk}\right)$,

Binary Variables:

$$v_{ij} = \begin{cases} 1 & \text{if any products transfer from receiving truck } i \text{ to shipping truck } j \\ 0 & \text{otherwise} \end{cases}$$

$$p_{iji'j'} = \begin{cases} 1 & \text{if any variable } t_{ij} \text{ immediately or directly precedes the variable } t_{i'j'} \text{ in the receiving sequence} \\ 0 & \text{otherwise} \end{cases}$$

$$p_{00i'j'} = \begin{cases} 1 & \text{if the variable } t_{i'j'} \text{ is placed at the first position in the receiving sequence} \\ 0 & \text{otherwise} \end{cases}$$

$$p_{00ij} = \begin{cases} 1 & \text{if the variable } t_{i'j'} \text{ is placed at the last position in the receiving sequence} \\ 0 & \text{otherwise} \end{cases}$$

$$v_{ij} = \begin{cases} 1 & \text{if any products transfer from receiving truck } i \text{ to shipping truck } j \\ 0 & \text{otherwise} \end{cases}$$

$$p_{iji'j'} = \begin{cases} 1 & \text{if any variable } t_{ij} \text{ immediately or directly precedes the variable } t_{i'j'} \text{ in the receiving sequence} \\ 0 & \text{otherwise} \end{cases}$$

$$q_{iji'j'} = \begin{cases} 1 & \text{if any variable } t_{ij} \text{ immediately or directly precedes the variable } t_{i'j'} \text{ in the shipping sequence} \\ 0 & \text{otherwise} \end{cases}$$

$$p_{00ij} = \begin{cases} 1 & \text{if the variable } t_{ij} \text{ is placed at the first position in the shipping sequence} \\ 0 & \text{otherwise} \end{cases}$$

$$p_{00ij} = \begin{cases} 1 & \text{if the variable } t_{ij} \text{ is placed at the last position in the shipping sequence} \\ 0 & \text{otherwise} \end{cases}$$

Data:

R = Number of receiving trucks in the set,

S = Number of shipping trucks in the set,

N =Number of product types in the set,

r_{ik} = Number of units of product type *k*, which is initially loaded in receiving truck i,

s_{jk} = Number of units of product type *k,* which is initially loaded for shipping truck *j*,

D = Delay time for truck change,

V = Moving or travel time of products from the receiving dock to the shipping dock,

M = Big number.

Mathematical Model

min   T

subject to

$$T \geq L_{ij}, \forall\, i, j \tag{1}$$

$$\sum_{j=1}^{S} x_{ijk} = r_{ik}, \forall\, i, k \tag{2}$$

$$\sum_{i=1}^{R} x_{ijk} = s_{jk}, \forall\, j, k \tag{3}$$

$$\sum_{k=1}^{N} x_{ijk} = t_{ik}, \forall\, i, j \tag{4}$$

$$t_{ij} \leq M v_{ij}, \forall\, i, j \tag{5}$$

$$v_{ij} = \sum_{i'=1}^{R} \sum_{j'=1}^{S} p_{iji'j'} + p_{ij00}\,, \forall\, i, j \tag{6}$$

$$v_{i'j'} = \sum_{i=1}^{R} \sum_{j=1}^{S} p_{iji'j'} + p_{00ij}\,, \forall\, i, j \tag{7}$$

$$v_{ij} = \sum_{i'=1}^{R}\sum_{j'=1}^{S} q_{iji'j'} + q_{ij00}, \forall i,j \tag{8}$$

$$v_{i'j'} = \sum_{i'=1}^{R}\sum_{j'=1}^{S} q_{iji'j'} + q_{00ij}, \forall i,j \tag{9}$$

$$\sum_{i'=1}^{R}\sum_{j'=1}^{S} p_{00i'j'} = 1, \tag{10}$$

$$\sum_{i=1}^{R}\sum_{j=1}^{S} p_{ij00} = 1, \tag{11}$$

$$\sum_{i'=1}^{R}\sum_{j'=1}^{S} q_{00i'j'} = 1, \tag{12}$$

$$\sum_{i=1}^{R}\sum_{j=1}^{S} q_{ij00} = 1, \tag{13}$$

$$p_{ijij} = 0, \forall i,j \tag{14}$$

$$q_{ijij} = 0, \forall i,j \tag{15}$$

$$U_{i'j'} \geq U_{ij} + t_{ij} - M\left(1 - p_{iji'j'}\right), \forall i,j,i',j' \text{ and where } i = i' \tag{16}$$

$$U_{i'j'} \geq U_{ij} + t_{ij} + D - M\left(1 - p_{iji'j'}\right), \forall i,j,i',j' \text{ and where } i \neq i' \tag{17}$$

$$L_{ij} \geq U_{ij} + V + t_{ij}, \forall i,j \tag{18}$$

$$L_{i'j'} \geq L_{ij} + t_{i'j'} - M\left(1 - q_{iji'j'}\right), \forall i,j,i',j' \text{ and where } j = j' \tag{19}$$

$$L_{i'j'} \geq L_{ij} + t_{i'j'} + D - M\left(1 - q_{iji'j'}\right), \forall i,j,i',j' \text{ and where } j \neq j' \tag{20}$$

All variables ≥ 0.

Eq. (1) specifies that makespan is greater equal to the time that the last product is loaded into the last scheduled shipping truck. Eq. (2) computes that total number of units of k^{th} product from receiving truck i are moved to all shipping trucks are the same as the number of products scheduled for receiving truck i. Similarly, Eq. (3) guarantees that, for shipping truck j, total number of outgoing products type k from all receiving trucks is the same as total number of incoming product type k. Variable t_{ij} used in Eqs. (16-20) measures the time of loading or unloading. Eq. (5) is to ensure an appropriate relationship between t_{ij} and v_{ij}. Based on Eq. (6), only one of t_{ij} when $v_{ij} = 1$ can immediately remain in the sequence compared with $t_{i'j'}$. Eq. (7) ensures that when $v_{ij} = 1$ only one of $t_{i'j'}$ is scheduled right after after t_{ij}. Similarly, Eq.(6-8) guaranty that only one of t_{ij} are directly in priority compared with other $t_{i'j'}$ when $v_{ij} = 1$. According to Eq. (9), only one of $t_{i'j'}$ happens right after t_{ij} when $v_{i'j'} = 1$.

According to Eq. (10), only one of receiving trucks' $t_{i'j'}$ is scheduled in the beginning of the sequence and Eq. (11) specifies that only one of receiving trucks' t_{ij} should remain in the last schedule. In addition, Eq. (12) and Eq. (13) assures that only one of the shipping trucks' $t_{i'j'}$ arrives at the beginning of the sequence and only one of t_{ij} comes last. Eq. (14) and Eq. (15) ensure that there is no consecutive sequence, which transfers products from the same receiving truck to the same shipping truck.

Eq. (16) and Eq. (17) provide a suitable sequence for unloading times for t_{ij} variables. If there is no change on receiving truck ($i = i'$) we use Eq. (16) and when there is receiving truck ($i \neq i'$) we requires to measure delay time using Eq. (17).

Eq. (18) setup a relationship between L_{ij} and U_{ij}, finally, Eq. (19) and Eq. (20) setup a valid loading time for t_{ij} based on the orders received. If there is no change between two consecutive shipments ($j = j'$) Eq. (19) becomes active, otherwise Eq. (20) is used to calculate delay time.

As we can observe, there are literally considerable number of binary variables, which make it impossible to solve the resulted problem for real-world applications. Ghobadian et al. (2012) used a hybrid of heuristic and metaheuristics to solve this problem. In this paper, we develop a genetic algorithm and compare our results with the hybrid GRASP method.

Holland (1975) is believed to be the first who introduced Genetic algorithm (GA), Goldberg (1989) developed the idea and it has been wiedly implemented for many problems. In this paper, we use GA to schedule trucks for our proposed cross docking problem and each Chromosom are first randomly generated and they are sent to dock post and for crossover, we use two functions. The first function uses one cut point and the second function implements two cut points, randomly. Obviously, when a truck is selected more than once, transformation is not accomplished. Fig. 1 shows details of our work.

Parent 1

<table><tr><td>3</td><td>8</td><td>9</td><td>5</td><td>1</td><td>10</td><td>2</td><td>7</td><td>4</td><td>6</td></tr><tr><td colspan="4">1</td><td colspan="6">2</td></tr></table>

First Child

<table><tr><td>3</td><td>8</td><td>9</td><td>5</td><td>10</td><td>1</td><td>2</td><td>4</td><td>6</td><td>7</td></tr><tr><td colspan="4">1</td><td colspan="6">4</td></tr></table>

Parent 2

<table><tr><td>8</td><td>7</td><td>3</td><td>1</td><td>10</td><td>5</td><td>2</td><td>4</td><td>6</td><td>9</td></tr><tr><td colspan="4">3</td><td colspan="6">4</td></tr></table>

Second Child

<table><tr><td>8</td><td>7</td><td>3</td><td>1</td><td>10</td><td>5</td><td>2</td><td>4</td><td>6</td><td>9</td></tr><tr><td colspan="4">3</td><td colspan="6">2</td></tr></table>

Fig. 2. Single point Crossover

For the implementation of the second function, two cuts are selected and we demonstrate the implementation through Fig. 3 as follows,

Parent 1

<table><tr><td>3</td><td>8</td><td>9</td><td>5</td><td>1</td><td>10</td><td>2</td><td>7</td><td>4</td><td>6</td></tr><tr><td colspan="4">1</td><td colspan="3">2</td><td colspan="3">3</td></tr></table>

First Child

<table><tr><td>3</td><td>8</td><td>9</td><td>5</td><td>1</td><td>2</td><td>10</td><td>7</td><td>4</td><td>6</td></tr><tr><td colspan="4">1</td><td colspan="3">5</td><td colspan="3">3</td></tr></table>

Parent 2

<table><tr><td>8</td><td>7</td><td>3</td><td>1</td><td>4</td><td>5</td><td>6</td><td>10</td><td>5</td><td>9</td></tr><tr><td colspan="4">4</td><td colspan="3">5</td><td colspan="3">6</td></tr></table>

Second Child

<table><tr><td>8</td><td>7</td><td>3</td><td>1</td><td>4</td><td>6</td><td>2</td><td>10</td><td>5</td><td>9</td></tr><tr><td colspan="4">4</td><td colspan="3">2</td><td colspan="3">6</td></tr></table>

Fig. 3. Double point Crossover

As we can observe from Fig. 3, there are two cuts in our selection strategy and two children are generated. In this example, we assume two gens of 4 and 7 are selected. Similar to the first function, if section 5 has no common gen with sections 1 and 3, transformation happens. Otherwise, the operations are accomplished with unallocated gens. To select two new parents, tournament selection strategy is employed. Fig. 4 shows details of mutation used for the proposed GA implementation.

Parent

8	7	3	1	4	6	2	10	5	9

Child Mutation

8	1	3	2	4	6	5	10	7	9

Fig. 4. Details of mutation operations

3. The results

In this section, we present details of the implementation of our GA code on test problems used earlier in other work (Ghobadian et al., 2012). We have used MATLAB to code the problem and implemented Intel ® Core™ DUE CPU processor with 4GB Ram. Table 1 demonstrates details of our findings on some test problems. Fig. 5 compares the performance of these two methods.

Table 1
Makespan obtained by the GRASP and Heuristics for the test problem (CPU times are seconds)

Test Problem	Receiving Trucks	Shipping Trucks	Number of Products	GRASP Metaheuristic	CPU	GA	CPU
1	6	4	7	2694	13.53	3230	22.98
2	8	9	6	5713	49.14	6163	35.09
3	10	11	8	7257	80.21	8007	43.83
4	9	10	7	4699	63.03	5424	37.70
5	11	12	10	10803	112.14	12153	56.86
6	12	12	10	11922	117.22	13047	55.86
7	12	13	14	8118	184.19	10743	73.96
8	13	12	14	4306	173.38	6931	68.17
9	13	14	14	9496	229.12	12571	82.22
10	14	12	15	12723	188.05	14898	75.34
11	14	15	16	8388	260.76	11313	90.96
12	15	13	12	15017	184.94	16817	74.56
13	12	14	13	13828	163.26	14653	65.00
14	16	15	16	6950	296.75	10700	104.73
15	17	18	12	9110	341.91	12860	105.07
16	18	18	14	10280	396.88	13505	112.34
17	18	19	15	15217	503.64	18592	122.34
18	19	19	16	22245	536.36	25245	125.19
19	20	19	15	11121	611.92	15996	138.10
20	20	20	17	13372	655.40	17122	137.77

As we can observe from the results, in most cases, the GRASP model provides better objective values. However, in terms of CPU time, GA provides better performance compares with GRASP as observed from Fig. 6. According to the results of Fig. 6, GA presents a linear trend while we see an increasing trend for the implementation of GRASP as the size of problem increases. Therefore, we can conclude that for large-scale problems and in some cases, real world case studies, GA may a better alternative strategy since it can find feasible solution within an acceptable amount of time. Note that many cross docking problems need to be solved more often and the implementation of a metaheuristics method is only recommended that the feasible solutions can be computed as quickly as possible. This would also help improve the quality of solutions by analysis the problem under different circumstances. For instance, when there are some uncertainties associated with input parameters, one way is to run the problem under different scenarios and then make a final decision. In such events, GA may perform far better than alternative GRASP methodology.

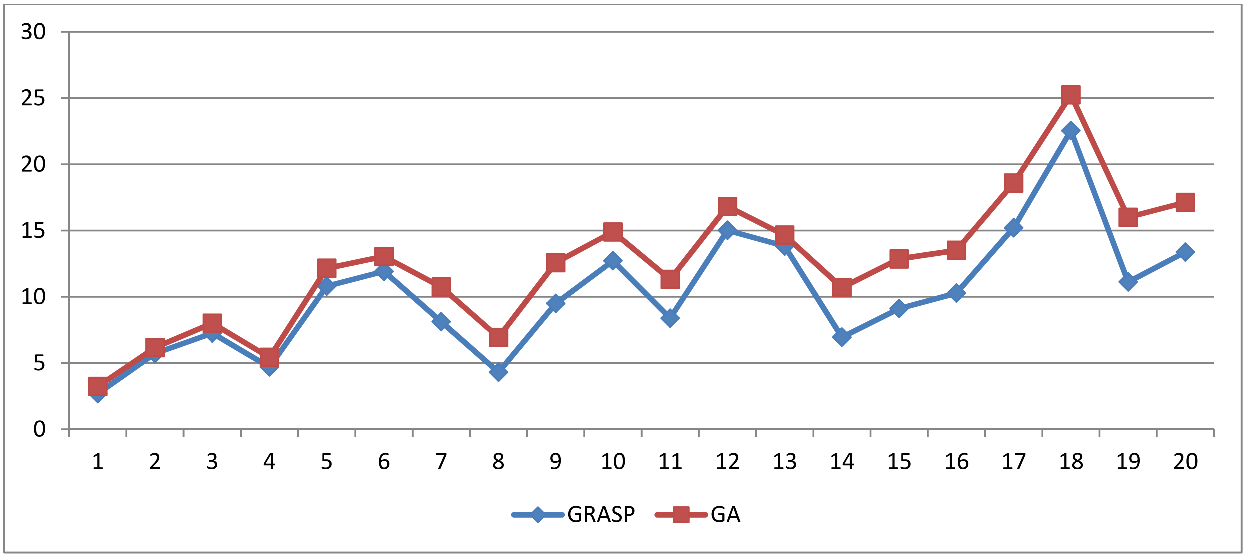

Fig. 5. Comparison of GA Vs GRASP in terms of objective function

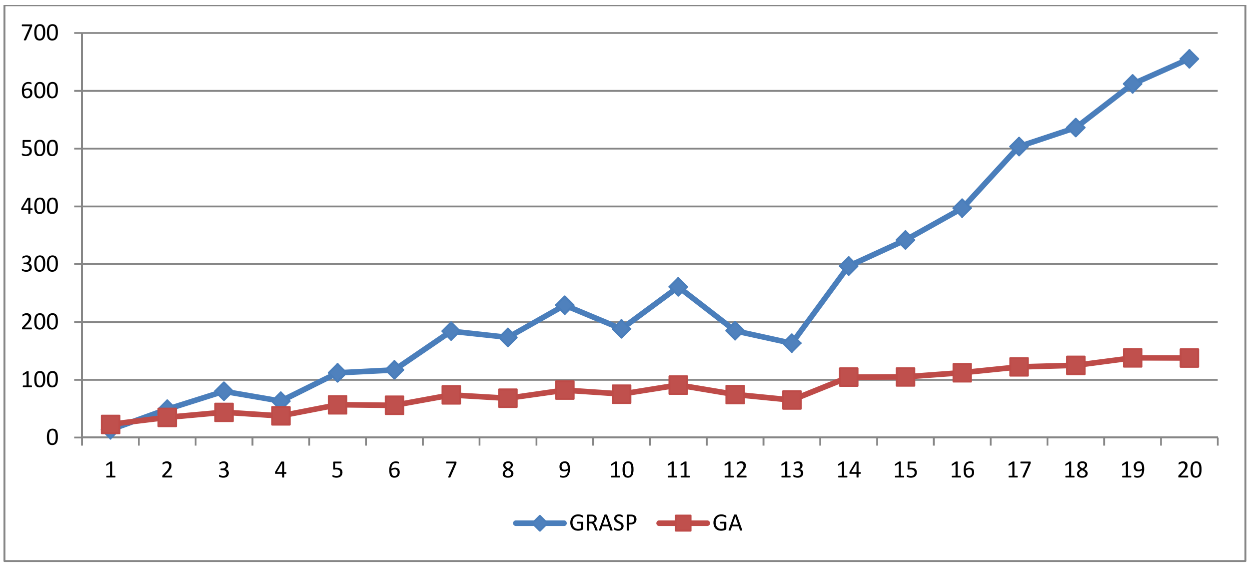

Fig. 6. Comparison of GA Vs GRASP in terms of CPU time

4. Conclusion

In this paper, we have considered a cross docking problem by considering temporary storage and repeat holding pattern in the system. The proposed model of this paper was formulated as mixed integer programming and genetic algorithm method was implemented to solve the resulted problem. The performance of the proposed model has been compared with a hybrid method using some randomly generated test problems. The preliminary results indicated that GRASP provides better objective values compared with the proposed GA problem. However, in terms of CPU time, GA provides much better results. The proposed model of this paper can be used for cross docking problems with more than one single objective function. Such a problem can be solved using multi objective GRASP methodologies to generate efficient Pareto solutions and we leave it for interested researchers as future research.

Acknowledgment

The authors would like to thank the anonymous referees for constructive comments on earlier version of this paper.

References

Apte, U.M., & Viswanathan, S. (2000). Effective cross docking for improve distribution efficiencies. *International Journal of logistics: Research and applications*, 3(3), 291-302.

Baker, K. R. (1974). Introduction to Sequencing and Scheduling. John Wiley & Sons, New York.

Barbarosoglu, G., & Ozgur, D. (1999). A tabu search algorithm for the vehicle routing problem. *Computer &Operations Research*. 26, 255-270.

Boloori Arbani, A.R., Fatemi Ghomi, S.M.T., & Zandieh, M. (2011).Meta-heuristics implementation for scheduling of trucks in a cross-docking system with temporary storage. *Expert systems with Applications*, 38(3), 1964-1979.

Ghobadian, E., Tavakkoli-Moghaddam, R., Javanshir, H., & Naderi-Beni, M. (2012). Scheduling trucks in cross docking systems with temporary storage and dock repeat truck holding pattern using GRASP method. *International Journal of Industrial Engineering Computations*, 3(5), 777–786.

Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley Longman.

Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor.

Javanshir, H., & Haghighi, M. (2011) A GRASP model in network design for two-stage supply chain. *Management Science Letters*, 1, 203-212.

Mosheiov, G. (1998). Vehicle routing with pick-up and delivery: tour-partitioning heuristics. *Computers & Industrial Engineering*, 34, 669-684.

Nascimento, M.C.V., Resende, M.G.C., & Toledo. F.M.B. (2010). GRASP heuristics with path-relinking for the multi-plant capacitated lot sizing problem. *European Journal of Operation Research, 200, 747-754.*

Pitsoulis, L.S., Resende, M.G. (2002). Greedy randomized adaptive search procedure. *Handbook of applied Optimization, Oxford University Press*, 168-183.

Resende, M.G., & Ribeiro, C.C. (2002). Greedy randomized adaptive search procedure. *Handbook in Metaheuristic, Kluwer Academic Publishers*, 219-249.

Rohrer, M. (1995). Simulation and cross docking. *In: Proceeding of the 1995 Winter Simulation Conference*, 846-849.

Soltani, R., & Sajadi, S.J. (2010). Scheduling trucks in cross-docking systems: A robust meta-heuristics approach. *Transportation Research - Part E*, 46, 650-666.

Feo, T.A., & Resende, M.G.C. (1989). A probabilistic heuristic for a computationally difficult set covering problem. *Operation Research Letters*, 8, 67-71.

Feo, T.A., & Resende, M.G.C. (1995). Greedy randomized adaptive search procedure. *Journal of Global Optimization*, 6, 109-133.

Feo, T.A., Resende, M.G.C., Smith, SH. (1994). A Greedy randomized adaptive search procedure for maximum independent set. *Operation Research*, 42, 860- 878.

Vahdani, B., Soltani, R., & Zandieh, M. (2009).Scheduling the truck holdover recurrent dock cross-dock problem using robust meta-heuristics. *International Journal of Advanced Manufacturing Technology* , 46, 769-783.

Vahdani, B., & Zandieh, M. (2010).Scheduling trucks in cross-docking system: Robust meta-heuristics. *Computers & Industrial Engineering*, 58, 12-24.

Yu, W., & Egbelu, P. J. (2008). Scheduling of inbound and outbound trucks in cross docking system with temporary storage. *European journal of Operational Research*, 184, 377-396.

Yu, W. (2002). *Operational strategies for cross docking systems*. Dissertation, Iowa state University. Ames, IA, USA.

Zapfel, G., Braune, R., & Bogl, M. (2010). *Metaheuristic Search Concepts: A tutorial with Applications to Production and Logistics*. Springer-Verlag Berlin Heidelberg, 75-82.