

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

Ehsan Ghobadian^{a*}, Reza Tavakkoli-Moghaddam^b, Hassan Javanshir^a and Mahdi Naderi-Beni^a

^aDepartment of Industrial Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran

^bDepartment of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran

ARTICLE INFO

Article history:

Received 18 May 2012

Received in revised format

7 July 2012

Accepted July 31 2012

Available online

5 August 2012

Keywords:

GRASP

Scheduling

Temporary storage

Metaheuristics

Cross docking

ABSTRACT

Cross docking play an important role in management of supply chains where items delivered to a warehouse by inbound trucks are directly sorted out, reorganized based on customer demands, routed and loaded into outbound trucks for delivery to customers without virtually keeping them at the warehouse. If any item is held in storage, it is usually for a short time, which is normally less than 24 hours. The proposed model of this paper considers a special case of cross docking where there is temporary storage and uses GRASP technique 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 GRASP method. The preliminary test results indicate that the GRASP method performs better than alternative solution strategies.

© 2012 Growing Science Ltd. All rights reserved

1. Introduction

Cross docking is one of the most important issues in supply chain management and there have been growing interests in this problem under different conditions (Barbarosoglu & Ozgur, 1999). Cross docking is a warehouse management idea where items delivered to a warehouse by inbound trucks are directly sorted out, reorganized based on customer demands, routed and loaded into outbound trucks for delivery to customers without virtually keeping them at the warehouse. If any item is held in storage, it is usually for a short time, which is normally less than 24 hours. This way, the turnaround times for customer orders, inventory management cost, and warehouse space requirements are reduced. Yu (2002) in his dissertation discussed cross docking problem under various assumptions. Yu and Egbelu (2008) determined the most suitable cross docking or scheduling sequence for both inbound and outbound trucks by minimizing total operation time when a temporary storage buffer is considered at shipping dock. The product assignment to trucks and the docking sequences of the inbound and outbound trucks were all determined, simultaneously.

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

Cross docking problem is normally formulated as a mixed integer problems and they are mathematically considered as NP-Hard problem (Feo & Resende, 1989; Mosheiov, 1998). Therefore, there is a need to use some metaheuristics to solve such problem. Rohrer (1995) discussed cross docking problem and the implementation of simulation to solve such problem.

Vahdani and Zandieh (2010) presented the implementation of five meta-heuristic algorithms including genetic algorithm (GA), tabu search (TS), simulated annealing (SA), electromagnetism-like algorithm (EMA) and variable neighborhood search (VNS) to schedule the trucks in cross-dock systems to minimize total operation time when a temporary storage buffer to hold temporarily items is located at the shipping dock. They used response surface methodology (RSM) methodology to tune their problem parameters. They also considered two kinds 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 hybrid meta-heuristics, hybrid simulated annealing and hybrid variable neighborhood search, to solve cross docking problem by achieving the best possible sequence of truck pairs. They used different sample problems to demonstrate the effectiveness 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 different plants were allowed. They developed a Greedy Randomized Adaptive Search Procedure (GRASP) heuristic as well as a path-relinking intensification procedure to detect cost-effective solutions for this problem. They also proposed some heuristics to solve some instances of the capacitated lot sizing problem with parallel machines. The results of the computational tests showed that the proposed heuristics outperform other heuristics previously described in the literature.

Bolori Arabani et al. (2011) developed another some meta-heuristics implementation for scheduling of trucks in a cross-docking system with temporary storage. Fig. 1 shows a sample of cross docking system.

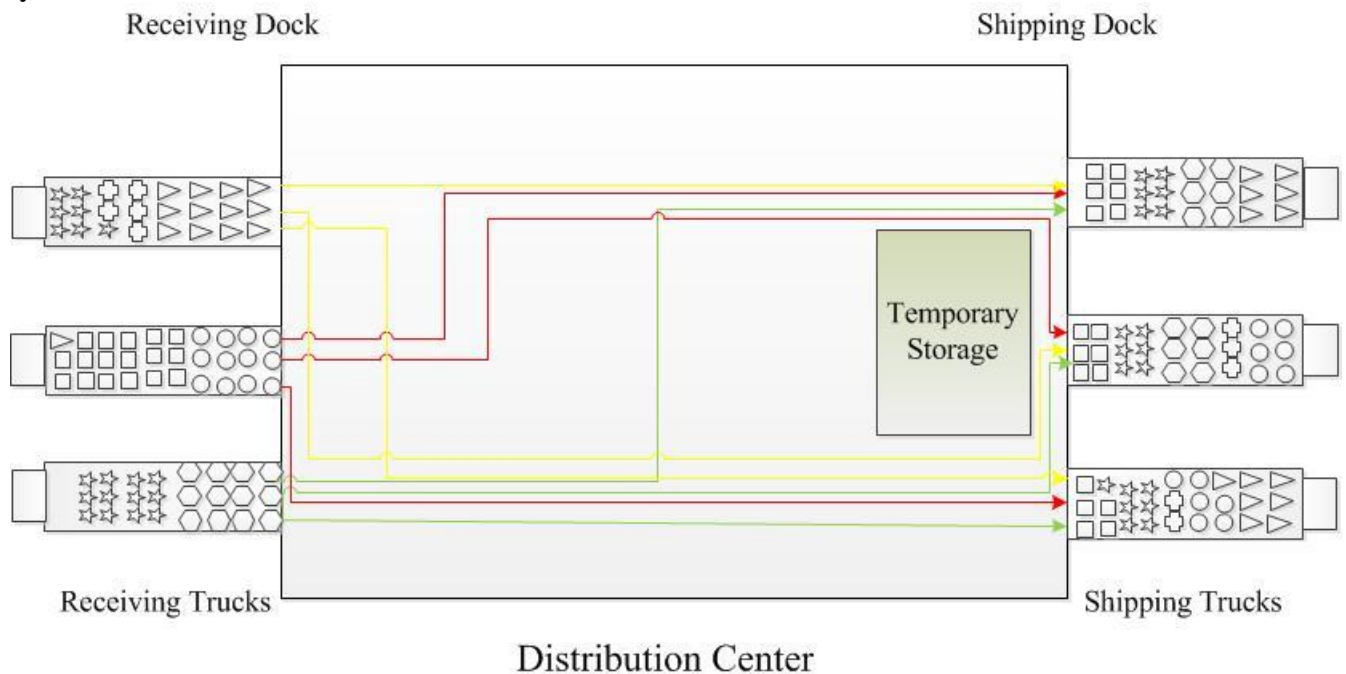


Fig .1. Cross docking distribution center

Yu (2002) proposed a model by assuming that there is a temporary storage in cross docking system and each two groups of trucks of receiving and shipping of loaded can alternatively enter into cross-dock. The temporary storage allows 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.

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 $(t_{ij} = \sum_{k=1}^N x_{ijk})$,

Binary Variables:

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

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

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

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

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

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

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

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

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

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

$$T \geq L_{ij}, \forall i, j \quad (1)$$

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

$$\sum_{i=1}^R x_{ijk} = s_{ik}, \forall i, k \quad (3)$$

$$\sum_{k=1}^N x_{ijk} = t_{ij}, \forall i, j \quad (4)$$

$$t_{ij} \leq Mv_{ij}, \forall i, j \quad (5)$$

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

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

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

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

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

$$\sum_{i'=1}^R \sum_{j'=1}^S p_{ij'00} = 1, \quad (11)$$

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

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

$$p_{ijij} = 0, \forall i, j \quad (14)$$

$$q_{ijij} = 0, \forall i, j \quad (15)$$

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

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

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

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

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

All variables ≥ 0 .

According to Eq. (1), makespan is greater equal to the time that the last product is loaded into the last scheduled shipping truck. Eq. (2) assures that total number of units of k^{th} product from receiving truck i are shipped to all shipping trucks are the same as the number of products scheduled for receiving truck i . Similarly, Eq. (3) assures 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) computes the time of loading or unloading. Eq. (5) guarantees an appropriate relationship between t_{ij} and v_{ij} . According to Eq. (6) only one of t_{ij} when $v_{ij} = 1$ can immediately stay in the sequence compared with $t_{i'j'}$. Eq. (7) guarantees that when $v_{ij} = 1$ only one of $t_{i'j'}$ is scheduled immediately after t_{ij} . Similarly, Eq.(6-8) assures 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 $v_{i'j'} = 1$.

Eq. (10) assures that only one of receiving trucks' $t_{i'j'}$ is scheduled in the beginning of the sequence and Eq. (11) assures that only one of receiving trucks' t_{ij} should stay in the last schedule. Similarly, Eq. (12) and Eq. (13) assures that only one of the shipping trucks' $t_{i'j'}$ comes at the beginning of the sequence and only one of t_{ij} comes last. Eq. (14) and Eq. (15) guarantee 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 an appropriate 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 need to compute delay time using Eq. (17).

Eq. (18) setup a good relationship between L_{ij} and U_{ij} and finally, Eq. (19) and Eq. (20) setup a valid loading time for t_{ij} based on the orders received. If there is no changes 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. For instance, there are 3 receiving truck and 3 shipping truck with 8 products, there will be 307 decision variables including 207 binary variables and 19 continuous variables. There are also 304 constraints, 189 inequality and 115 equality constraints. Yu (2002) proposed the following heuristic approach to solve the proposed model.

PHASE 1

- 1 initiate set of receiving truck, shipping truck, receiving and shipping truck in relation with them and quantity of products are associated between two shipping and receiving truck;
- 2 **for** shipping truck $i=0, \dots, n$ compare with receiving trucks $i=0, \dots, n$ **do**
- 3 **for** compare receiving truck with shipping trucks select one of strategies 1, 2 or 3;
- 4 **for** Select the receiving truck which has the largest relationship with the certain shipping truck.
- 5 update shipping truck and receiving truck list;
- 6 **end for;**
- 7 select the shipping truck which has the smallest relation with receiving trucks and remove it from shipping list. Update receiving trucks that have relationship with the certain shipping truck;
- 8 **end for;**
- 9 **end for;**

PHASE 2;

- 1 condition 1;
- 2 start from the first, **for** each two sequence shipping truck **do**;
- 3 **if** have one t_p^r in share;
- 4 change sequence
- 5 **elseif** there is more than one t_p^r in share and $t_p^r < D$;
- 6 change sequence;
- 7 **end if;**
- 8 condition 2;
- 9 start from the last shipping sequence;
- 10 **if** t_p^r in the former sequence set is smaller than truck change time **do**
- 11 revise sequence;
- 12 **end if;**

3. GRASP algorithm

Greedy randomized adaptive search procedure (GRASP) (Feo, 1989, 1994, 1995; Resende & Ribeiro, 2002; Pitsoulis & Resende, 2002) starts with an empty solution and as the search algorithm continues, new solutions are added to solution set until there is no further solution found. There are many applications of GRASP to solve engineering problems (Zapfel et al., 2010; Javanshir & Haghghi, 2011). There are some similarities between GRASP and Greedy construction method but there are some important changes between these two methods. GRASP uses restricted candidate list (RCL) and allows the algorithm diversified solution sets. The primary objective is to simplify the process of adding new solutions. Fig. 2 shows details of GRASP algorithm.

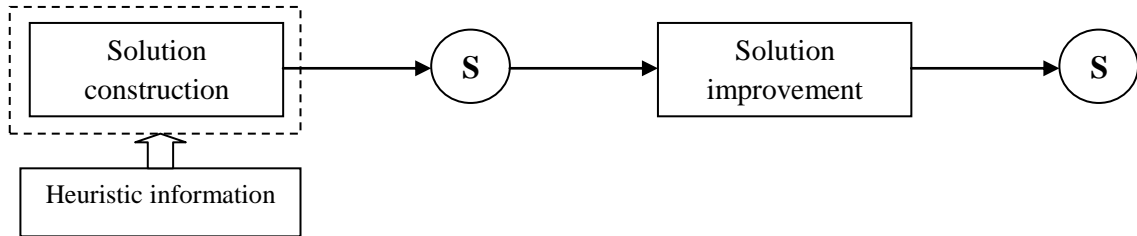


Fig. 2. GRASP-solution Processing

The solution procedure starts with an empty set and evaluates all elements based on performance function and their effects on quality of solution. These elements are sorted based on their performances in candidate list. Fig. 3 shows three levels of the operations.

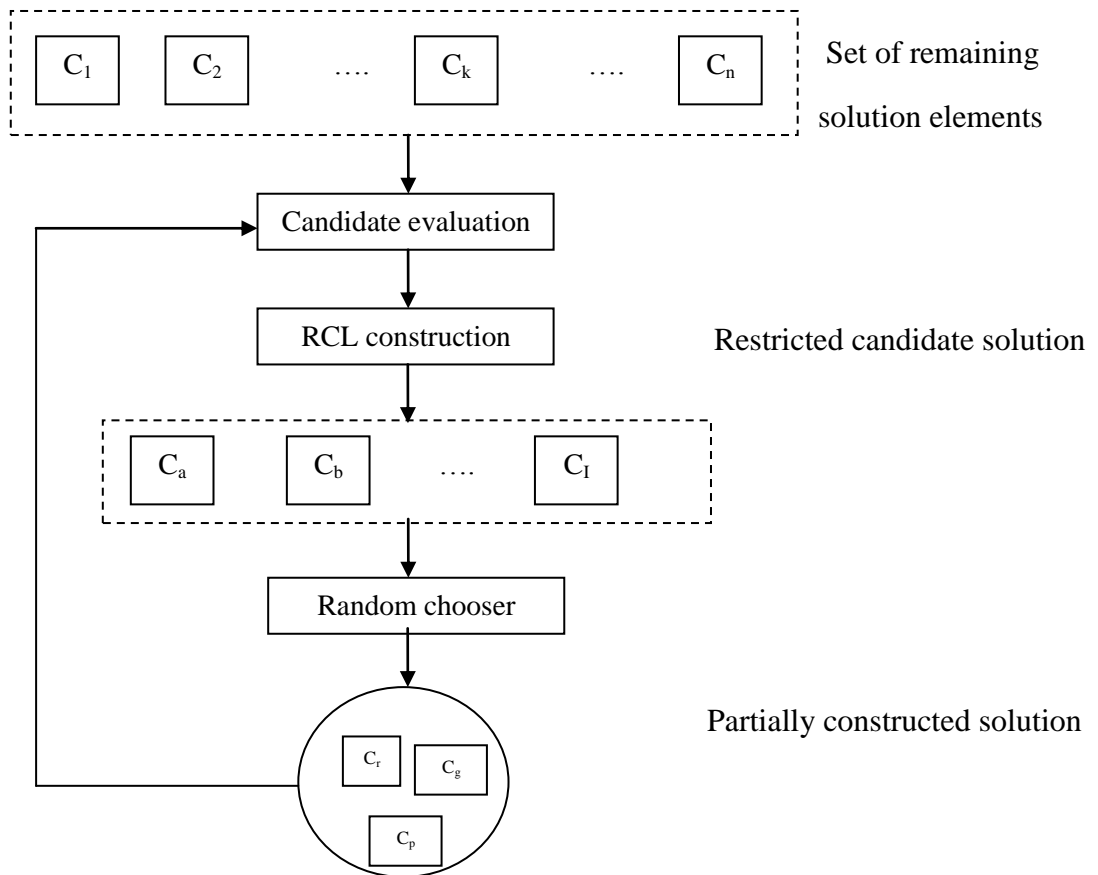


Fig. 3. Solution construction of the GRASP metaheuristic

Let g be objective function, to build construction list, we need to have g_{min} and g_{max} with $g_{min} = \min\{g(c_i)\}, \forall c_i \in C$ and $g_{max} = \max\{g(c_i)\}, \forall c_i \in C$. We use these two limits for the implementation of our GRASP method. For the proposed model of this paper, there are two RCL

namely cardinality-based and value Based. The cardinality-based method includes k best alternative solutions. For instance, if $k = 5$, there are five best elements in RCL and from this list, one sample is selected randomly. The value based method uses a parameter $\alpha \in [0 1]$ to build RCL. Suppose we deal with an optimization problem and all elements must maintain a value less than their limits to become eligible for RCL list, we use $\mu = g_{\min} + \alpha(g_{\max} - g_{\min})$ with $g(c_j) = [g_{\min}, \mu]$. When $\alpha = 1$ the method is completely random and when $\alpha = 0$ the algorithm is greedy. The following shows the summary of the proposed GRASP algorithm,

Step 1: Execute three heuristic algorithms 1, 2 and 3

Step 2: Compute c_j

Step 3: Compute g_{\min} and g_{\max}

Step 4: Add generated solution to RCL if $c_j \leq g_{\min} + \alpha(g_{\max} - g_{\min})$

Step 5: Select one of shipping truck and update the information

Step 6: If the number of solutions obtained is equal to S (number of shipping trucks) stop, otherwise goto step 1

The proposed GRASP algorithm first generates three heuristic solutions proposed by Yu (2002) and in each stage, we choose and dispatch one truck. Based on the calculated c_j we calculate g_{\min} and g_{\max} and based on the values of c_j we add any solution with $c_j \leq g_{\min} + \alpha(g_{\max} - g_{\min})$. One advantage of GRASP method is that the proposed GRASP method has only two parameters including the value of α and termination criteria and the implementation of this algorithm chooses $\alpha = 0.2$.

4. The results

The proposed GRASP method has been implemented using three strategies. All programs were coded using personal computer with 4GB RAM and Intel ® Core™ DUE CPU processor. Table 1 shows details of our implementations on some test problems. As we can observe from the results, in most cases, the proposed model of this paper provides better objective values.

Table 1

Makespan obtained by the GRASP and Heuristics for the test problem

Test Problem	Receiving Trucks	Shipping Trucks	Number of Products	GRASP Metaheuristic	heuristic Yu strategy 1	heuristic Yu strategy 2	heuristic Yu strategy 3	Best Heuristic
1	6	4	7	2694	2756	2756	2756	2756
2	8	9	6	5713	5713	5788	5788	5713
3	10	11	8	7257	7482	7257	7257	7257
4	9	10	7	4674	4749	4809	4907	4749
5	11	12	10	10878	11178	11253	11253	11178
6	12	12	10	11997	12372	12084	12372	12084
7	12	13	14	8118	8666	8419	8609	8419
8	13	12	14	4306	5074	4758	4682	4682
9	13	14	14	9571	9854	9854	10142	9854
10	14	12	15	12798	13031	12995	13179	12995
11	14	15	16	8538	9144	8388	8913	8388
12	15	13	12	15167	15631	15242	15242	15242
13	12	14	13	13828	13603	14005	13970	13603
14	16	15	16	7025	7586	7362	7587	7362
15	17	18	12	9185	10018	9901	9760	9760
16	18	18	14	10430	10880	10580	11169	10580
17	18	19	15	15367	15558	15900	16003	15558
18	19	19	16	22545	23453	22845	22865	22845
19	20	19	15	11271	11721	11344	11779	11344
20	20	20	17	13297	13550	13597	13447	13447

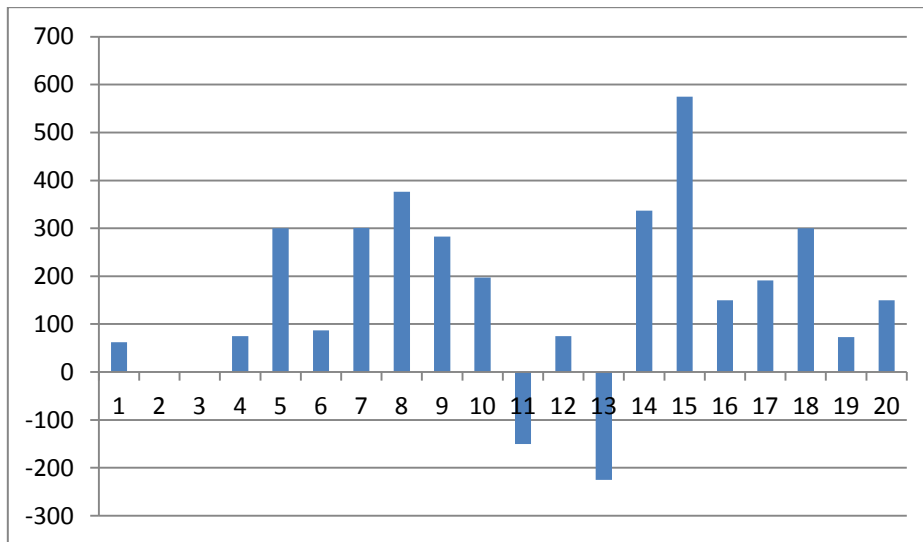


Fig. 4. Objective function difference between GRASP and Yu's heuristic

Fig. 4 shows the difference between the objective functions of the proposed GRASP and Yu's heuristics results. As we can observe, except two cases, there are 16 cases where GRASP performs better than best solution strategies provided by Yu's heuristic method, in two case there is no difference and only in two cases, Yu's method beats GRASP method.

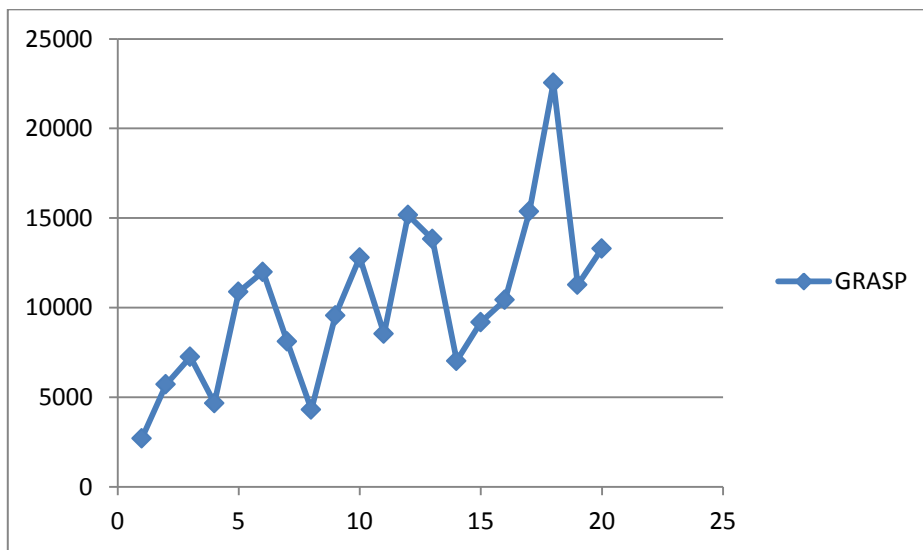


Fig. 5. GRASP performance for the test problem

Fig. 5 shows the performance of the proposed GRASP CPU time in seconds for 20 test problem. Note that the CPU time does not increase significantly as inputs of changes. In other words, the proposed GRASP can be easily implemented for some real-world case studies in reasonable amount of time.

5. Conclusion

In this paper, we have presented an extended 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 GRASP method was developed to solve the resulted problem. The performance of the proposed model has been compared with Yu's method using some randomly generated test problems. The preliminary results indicate that the proposed GRASP provides better objective values compared with alternative method. Besides, the proposed GRASP seems to be

applicable for real-world application in reasonable amount of time. 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. We would like to also thank Mr Vahdani for providing assistance in part of this research work.

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.
- Bolloori 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.
- 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.