Lecture Notes in Management Science (2013) Vol. **5**: 125–132
5th International Conference on Applied Operational Research, Proceedings

ISSN 2008-0050 (Print), ISSN 1927-0097 (Online)

Scheduling of trucks in cross-docking systems: a hybrid meta-heuristic algorithm

B. Vahdani, R. Tavakkoli-Moghaddam, and S.M. Mousavi

Department of Industrial Engineering, College of Engineering,
University of Tehran, Tehran, Iran
b.vahdani@ut.ac.ir; tavakoli@ut.ac.ir; sm.mousavi@ut.ac.ir

Abstract. Cross-docking is a logistics technique that minimizes the storage and order picking functions of a warehouse while still allowing it to serve its receiving and shipping functions. In this paper, we propose a novel hybrid meta-heuristic algorithm for solving scheduling trucks in cross-docking problems. This algorithm comprises three components: an initial population generation method based on ant colony optimization (ACO), simulated annealing (SA) as an evolutionary algorithm employs a certain probability to avoid becoming trapped in local optimum, and variable neighborhood search (VNS) that involves three local search procedures to improve the population. Moreover, to demonstrate the effectiveness of the proposed methods especially for large-sized problems, various test problems are solved. The computational results demonstrate that our proposed algorithm performs far better than those of Yu and Egbelu (2008).

Keywords: logistics, distribution, truck scheduling, cross-docking, hybrid meta-heuristic; ant colony optimization, simulated annealing, variable neighborhood search

Introduction

A typical warehouse is a dynamic and intelligent distribution center, in which products and packages are processed in real time and moved in and out on schedule. A dynamic and intelligent warehouse is also a place, in which all distribution and logistics functions are tied together and inventory storage is minimal. The input and output are also precisely regulated and streamlined in an intelligent manner. In a today's distribution environment, the pressure is on making the operations more efficient. Operations of the distribution center consist of five basic functions: receiving, sorting, storing, picking and shipping. If the cooperation of these five elements is improved, costs can be reduced and productivity can be improved. However, the best way to reduce cost and improve efficiency is not by simply improving a function but by eliminating it if feasible. Cross docking has the potential of eliminating storage and picking, the two most expensive warehousing operations. Cross docking is a method of distribution management that helps companies to control their distribution operations better. One of the earliest technical papers on cross-docking systems was presented by Rohrer [2]. Waller et al. [3] developed models to predict the changes in the retailer's system-wide inventory levels as a result of cross-docking. Ma and Chen [4] also studied the cross-docking scheduling problem with the total completion time, a dynamic programming was designed with computational complexity of $O\ (nm^2m)$. Chen and Song [5] studied the two-stage hybrid cross-docking scheduling problem. Yu and Egbelu [1] suggested a cross-docking system that has a temporary storage area in front of the shipping dock. The objective of the study was to find the best truck docking sequence for both inbound and outbound trucks to minimize the total operation time or to maximize the throughput of the cross docking system. Vahdani and Zandieh [6] proposed five meta-heuristics using the best heuristic result of Yu and Egbelu [1] as the initial solution. In this paper, we propose a hybrid meta-heuristic customizable approach that allows the definition of scheduling algorithms by appropriately selecting and combining several different features derived from three main meta-heuristics (i.e., ACO, SA and VNS). Liao et al. [7] proposed two hybrid differential evolution algorithms for optimal inbound and outbound truck sequencing in the operations of cross-docking center. Melo et al. [8] considered the problem of redesigning a supply chain network with multiple echelons and commodities, and modeled as a large-scale mixed-integer linear program. Then, they proposed a TS algorithm for solving the presented model. Ma et al. [9] focused on a new shipment consolidation and transportation problem in cross-docking distribution networks by considering setup cost and time windows constraints. Alpan et al. [10] addressed a transshipment problem in a multi-door cross-docking warehouse and made an attempt to find the best schedule of transshipment operations in order to minimize the sum of inventory holding and truck replacement costs. Dondo et al. [11] presented a hybrid multi-echelon multi-item distribution network contained a multi-echelon vehicle routing problem with cross-docking in supply chain management by minimizing the total transportation cost. The purpose of this work is to evaluate the effectiveness of integrating such a specific subset of features into a configurable hybrid meta-heuristic algorithm. Moreover, the algorithm proposed in this paper gives far better solutions rather than the heuristics presented by Yu and Egbelu [1].

Problem definition

In this work, we consider scheduling of trucks in cross-docking systems, which was first proposed by Yu and Egbelu [1] to minimize the makespan of a cross docking system. Makespan is defined as the total operating time of the cross docking operation.

Model formulation

The following notations and variables are used in this paper.

Notations

R Number of inbound trucks in the set.
S Number of outbound trucks in the set.
N Number of product types in the set.
r_{ij} Number of units of product type k that was initially loaded in inbound truck i.
s_{jk} Number of units of product type k that was initially needed for outbound truck j.
D Truck changeover time.
V Moving time of products from the receiving dock to the shipping dock.
M Big number.

Continuous variables

T Makespan
C_i Time when inbound truck i enters the receiving dock.
F_i Time when inbound truck i leaves the receiving dock.
d_j Time when outbound truck j enters the shipping dock.
L_j Time when outbound truck j leaves the shipping dock.

Integer variables

x_{ijk} Number of units of product type k that transfer from inbound truck i to outbound truck j.

Binary variables

v_{ij} $\begin{cases} 1 & \text{if any products transfer from inbound trcuk } i \text{ to outbound truck } j \\ 0 & \text{otherwise.} \end{cases}$

p_{ij} $\begin{cases} 1 & \text{if inbound trcuk } i \text{ preceeds inbound truck } j \text{ in the inbound truck sequence} \\ 0 & \text{otherwise.} \end{cases}$

q_{ij} $\begin{cases} 1 & \text{if outbound trcuk } i \text{ preceeds outbound truck } j \text{ in the outbound truck sequence} \\ 0 & \text{otherwise.} \end{cases}$

The mathematical model for a cross-docking problem can be presented by:

min T

s.t.

$$T \geq L_j, \quad \forall\, j \tag{1}$$

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

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

$$x_{ijk} \leq M v_{ij}, \quad \forall\, i, j, k \tag{4}$$

$$F_i \geq c_i + \sum_{k=1}^{N} r_{ik}, \quad \forall\, i \tag{5}$$

$$c_j \geq F_i + D - M(1 - p_{ij}), \quad \forall i, j; i \neq j \tag{6}$$

$$c_i \geq F_j + D - Mp_{ij}, \quad \forall i, j, i \neq j, \tag{7}$$

$$p_{ii} = 0, \quad \forall i, \tag{8}$$

$$L_j \geq d_j + \sum_{k=1}^{N} s_{jk}, \quad \forall j \tag{9}$$

$$d_j \geq L_i + D - M(1 - q_{ij}), \quad \forall i, j, i \neq j, \tag{10}$$

$$d_i \geq L_j + D - Mq_{ij}, \quad \forall i, j, i \neq j \tag{11}$$

$$q_{ii} = 0, \quad \forall i \tag{12}$$

$$L_j \geq c_i + V + \sum_{k=1}^{N} x_{ijk} - M(1 - v_{ij}), \quad \forall i, j \tag{13}$$

all variables ≥ 0.

Constraint (1) sets the makespan greater than or equal to the time the last scheduled outbound truck leaves the shipping dock. Constraint (2) ensures that the total number of units of product type *k* that transfer from inbound truck *i* to all outbound trucks is exactly the same as the number of units of product type *k* that was initially loaded in inbound truck *i*. Similarly, Constraint (3) ensures that the total number of units of product type *k* that transfer from all inbound trucks to outbound truck *j* is exactly the same as the number of units of product type *k* needed for outbound truck *j*. Constraint (4) just enforces the correct relationship between the x_{ijk} variables and the v_{ij} variables. Constraints (5)–(7) make a valid sequence for arriving and departing times for the inbound trucks based on their order. Constraint (8) ensures that no inbound truck can precede itself in the inbound truck sequence. Similar to Constraints (5)–(7) for inbound trucks, Constraints (9)–(11) function in a similar manner for the outbound trucks. Similar to Constraint (8), Constraint (12) ensures that no outbound truck can precede itself in the outbound truck sequence. Constraint (13) connects the leaving time for an outbound truck to the arriving time of an inbound truck if any products or items are transferred between the trucks.

Hybrid meta-heuristic algorithm

As previously noted, the objective of the study of a cross-docking system is to reach the minimum makespan. This goal is achieved through a suitable arrangement of inbound and outbound trucks. Prior to describe hybrid meta-heuristic algorithms, it is necessary to show the solutions representation scheme in these algorithms. As shown in Fig. 1, it consists of two parts such that the first part is related to the outbound trucks and the second part is related to the inbound trucks.

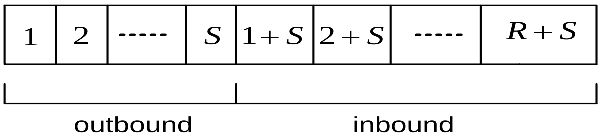

Fig. 1. Scheme of solutions

The hybrid algorithm proposed here combines three previously presented methods. The main idea of the algorithm can be described as follows. The search starts from some initial solution based on ACO and iterative moves are performed among neighboring solutions. At iteration, a random solution $P_0$is selected from the neighborhood and it is accepted with the probability given by Eq. (1). However, if $P_0$is not accepted, then neighborhood structure is changed in the same way as in VNS technique. In our experiments, all parameters were selected in the same way as presented in ACO, SA and VNS algorithms. The basic proposed hybrid algorithm structure is designed as shown Fig. 2.

```
Algorithm: hybrid algorithm
R_time← Set run time()
while run time <R_time do
 for each ant do
S* ← Generate initial solution()
l ← 1;
      for iterations ← 1 to a maximum number of iterations do
S ← S*;
k ← 0
T_k← Set initial temperature()
T_f← Set final temperature()
       while current temperature <T_f do
                 Shake procedure: find a random solution S′Є N_l (S);
                 Perform a local search on N_l (S′) to find a solution S″;
if f(S″) ≤ f(S) then
S* ← S″;
 1 ← 1;
                else
                   accept S′ as new solution with probability p(S′|T_k, S′)
                end if
                Adapt temperature (T_k)
 end while
l ← l+1;
end
      generate route; evaluate route
  end for
  verify for global or local best; evaporate pheromone in all trials;
  deposit pheromone on best global route.
end while
```

Fig. 2. Pseudo code of the proposed hybrid meta-heuristic algorithm

Experimental results

The results of running heuristics are presented in Table 1; however, the results shown in Table 2 are obtained by running the proposed algorithm. The results of the averaged RPD are reported in Table 3. As can be seen in these tables, the hybrid meta-heuristic algorithm is far better than the other heuristics. Moreover, to attain the best robustness of these algorithms, the Taguchi's robust design method is employed.

Table 1. Makespan obtained by the nine heuristic algorithms for test problems

	Problem size				*Heuristic solutions*									
Problem set	*No. of inbound trucks*	*No. of outbound trucks*	*No. of product types*	*Total No. of product types*	*RS1 SS1*	*RS1 SS2*	*RS1 SS3*	*RS2 SS1*	*RS2 SS2*	*RS2 SS3*	*RS3 SS1*	*RS3 SS2*	*RS3 SS3*	*Compound solution*
1	5	4	6	1030	1577	1577	1697	1609	1609	1609	1609	1609	1714	1577
2	9	9	9	2123	3824	3636	3872	3387	3423	3572	3387	3423	3774	3387
3	10	9	10	2164	4191	4132	4227	3784	3835	3918	3851	3993	3950	3784
4	11	10	10	3115	5474	5308	5413	5404	5452	5273	5381	5560	5273	5273
5	11	11	11	2200	4191	4153	4277	4440	3959	3985	4009	4235	3985	3959
6	11	12	11	2760	5065	5192	5122	4951	4927	4788	5099	5072	4751	4751
7	12	12	12	3060	5751	5959	5614	5766	5371	5784	5608	5556	5585	5371
8	13	11	13	2614	5185	4865	5316	4905	4621	4940	4711	4754	4865	4621
9	12	13	12	2782	5085	5179	5098	5166	5161	5179	5085	4990	5098	4990
10	14	12	10	2925	5360	5164	5337	5272	5552	5272	5272	5164	5272	5164
11	13	13	11	3454	5629	5847	5802	5650	5966	6130	5880	5847	5964	5629
12	14	14	13	5040	8259	8182	8764	8360	8036	8278	8259	8311	8652	8036
13	14	15	12	5655	9457	9164	9472	9120	9583	9531	9457	9103	9538	9103
14	15	13	13	4099	7097	7097	7133	7191	6997	7191	6874	6895	6988	6874
15	15	15	14	5060	8183	8549	8522	8388	8619	8317	8517	8619	8414	8183
16	16	13	15	5351	9218	9255	8927	9012	9099	9012	8838	9102	9197	8838
17	14	16	13	4609	7628	7725	7652	7689	7725	8050	7628	7725	7652	7628
18	16	16	11	4720	7702	7682	7914	7664	7793	7967	8028	7936	7563	7563
19	15	16	12	4603	7993	8202	7727	8260	8303	8166	7993	8303	7853	7727
20	16	17	16	5676	9626	9858	9803	9610	9468	9879	9677	9634	10102	9468
Average					6324.75	6336.3	6384.45	6281.4	6274.95	6342.05	6258.15	6291.55	6309.5	6096.3

Table 2. Best and average makespan obtained by the hybrid meta-heuristic

Test problem	*best makespan*	*average makespan*	*CPU time*	*Compound solution (CPU time)*
1	1577	1577	25.2377	0.1239
2	3332	3349.278	123.7314	1.8763
3	3545	3577.444	265.0704	3.0356
4	4707	4778.222	174.3322	2.2378
5	3663	3732.556	160.6066	2.2047
6	4480	4522.5	242.9322	3.3497
7	4930	5005.944	228.9875	3.3179
8	4385	4415.889	283.0079	4.0549
9	4622	4697.778	236.348	3.6479
10	4776	4816.111	141.0388	2.3489
11	5427	5458.556	341.4474	6.3471
12	7637	7761.412	382.9658	7.2145
13	8593	8709.222	271.6105	5.0089
14	6349	6502.444	302.8039	6.6719
15	7646	7937.667	315.0441	6.3149
16	8212	8349.333	315.9842	3.7891
17	7311	7382.324	258.1796	5.6791
18	7231	7361.944	204.5715	4.7892
19	7359	7444.444	284.6046	5.3421
20	9076	9188.667	372.4112	7.6491
Average	5742.9	5828.437	285.17	5.1345

Table 3. Average relative percentage deviation (*RPD*) for algorithms

Test problem	*Compound solution of the heuristic*	*Hybrid meta-heuristic*
1	**0**	**0**
2	0.0165	**0.0051**
3	0.0674	**0.0091**
4	0.1202	**0.0151**
5	0.0808	**0.0189**
6	0.0604	**0.0094**
7	0.0894	**0.0154**
8	0.0538	**0.0070**
9	0.0796	**0.0163**
10	0.0812	**0.0083**
11	0.0372	**0.0058**
12	0.0522	**0.0162**
13	0.0593	**0.0135**
14	0.0826	**0.0241**
15	0.0702	**0.0381**
16	0.0762	**0.0167**
17	0.0433	**0.0097**
18	0.0459	**0.0181**
19	0.0500	**0.0116**
20	0.0431	**0.0124**
Average	0.060465	**0.01354**

Conclusions

In this paper, we considered the problem of scheduling of trucks in cross-docking systems, which had been initiated by Yu and Egbelu [1]. To solve the considered problem, we proposed a novel hybrid meta-heuristic algorithm. The main hybrid meta-heuristic algorithm had three unique features: its population-based evolutionary searching ability by ant colony optimization (ACO), its ability to balancing exploration and exploitation by simulated annealing (SA) and its local improvement ability by variable neighborhood search (VNS). In the proposed hybrid meta-heuristic algorithm, the balance between the global exploration and the local exploitation was stressed. This method had several abilities in searching solution space. The computational results revealed that the proposed hybrid meta-heuristic algorithm yielded far better results than the heuristic algorithm presented by Yu and Egbelu [1].

References

Yu, W., Egbelu, P.J.: Scheduling of inbound and outbound trucks in cross docking systems with temporary storage. European Journal of Operational Research, 184, 377–396 (2008).

Rohrer, M.: Simulation and cross docking. In: Proceedings of the 1995 Winter Simulation Conference, 846–849 (1995).

Waller, MA, Cassady, RC, Ozment, J: Impact of cross-docking on inventory in a decentralized retail supply chain. Transportation Research - Part E. 42, 359–382 (2006).

Ma, D.Y., Chen, F.: Dynamic Programming Algorithm on Two Machines Cross Docking Scheduling. Journal of Shanghai Jiao Tong University, 41(5): 852–856 (2007).

Chen, F., Song, K.L.: Minimizing makespan in two-stage hybrid cross docking scheduling problem, Computers & Operations Research, 36(6), 2066–2073 (2009).

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

Liao TW, Egbelu PJ, Chang PC. Two hybrid differential evolution algorithms for optimal inbound and outbound truck sequencing in cross docking operations. Applied Soft Computing 2012; 12:3683–97.

Melo MT, Nickel S, Saldanha-da-Gama F. A tabu search heuristic for redesigning a multi-echelon supply chain network over a planning horizon. International Journal of Production Economics 2012; 136:218–30.

Ma H, Miao Z, Lim A, Rodrigues B. Cross docking distribution networks with setup cost and time window constraint. Omega 2011; 39:64–72.

Alpan G, Ladier A-L, Larbi R, Penz B. Heuristic solutions for transshipment problems in a multiple door cross docking warehouse. Computers and Industrial Engineering 2011; 61:402–8.

Dondo R, Méndez CA, Cerdá J. The multi-echelon vehicle routing problem with cross docking in supply chain management. Computers and Chemical Engineering 2011; 35(12):3002–24.