01 Sep 2017

Loading Time Flexibility in Cross-Docking Systems

Dincer Konur
Missouri University of Science and Technology, konurd@mst.edu

Mihalis M. Golias

Recommended Citation

D. Konur and M. M. Golias, "Loading Time Flexibility in Cross-Docking Systems," *Procedia Computer Science*, vol. 114, pp. 491-498, Elsevier, Sep 2017.
The definitive version is available at https://doi.org/10.1016/j.procs.2017.09.011

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 114 (2017) 491–498

Procedia Computer Science

www.elsevier.com/locate/procedia

Complex Adaptive Systems Conference with Theme: Engineering Cyber Physical Systems, CAS
October 30 – November 1, 2017, Chicago, Illinois, USA

Loading Time Flexibility in Cross-docking Systems

Dincer Konur[a]*, Mihalis M. Golias[b]

[a]Engineering Management and Systems Engineering, Missouri University of Science and Technology, Rolla, MO, 65401
[b]Civil Engineering, University of Memphis, Memphis, TN, 38152

Abstract

In this study, we investigate truck-to-door assignment problem for loading outgoing trucks in a cross-docking system with flexible handling times. Specifically, a truck's loading time depends on the number of workers assigned to the outbound door, where the truck is being loaded. An optimization problem is formulated to jointly determine the number of workers and the trucks to be loaded at each door. The resulting problem is a nonlinear integer programming model. Due to the complexity of this model, two evolutionary heuristic methods are proposed for solution. First heuristic method is based on truck assignments while the second heuristic is based on worker assignments. A numerical study is conducted to compare the two heuristic methods.

Peer-review under responsibility of the scientific committee of the Complex Adaptive Systems Conference with Theme: Engineering Cyber Physical Systems.

Keywords: Cross-docking; Flexibility; Heuristics

1. Introduction and Literature Review

With the increasing need for efficient and timely transportation with less inventory in today's competitive industry, cross-docking has become a very popular practice in different supply chains. Cross-docking enables shipment consolidation, which decreases transportation costs as fewer but more utilized trucks are used for freight shipment. Furthermore, warehousing costs are reduced due to the fast movements of items, with minimized inventory holding, from different origins to different destinations through the cross-dock facility. Typical operations in a cross-dock facility are unloading the products from incoming trucks, sorting and storing products, and loading the products to

1877-0509
Peer-review under responsibility of the scientific committee of the Complex Adaptive Systems Conference with Theme: Engineering Cyber Physical Systems.
10.1016/j.procs.2017.09.011

outgoing trucks based on customer demands. Cross-docking is noted to be successfully practiced by many industries and result in cost savings along a supply chain [1,2,3].

It is not surprising that there is a growing literature on cross-docking. We refer the reader to recent review papers[4,5,6] for overviews of the types of cross-docking problems that have been analyzed in the literature. In this study, a truck-to-door assignment problem at a cross-dock facility is considered. The aim of truck-to-door assignment problem is to find the optimal assignment of incoming and/or outgoing trucks to the dock doors at a cross-dock facility[5]; and, different than the truck scheduling problem at cross-dock facilities, time dimension is not regarded in truck-to-door assignment problems[5]. Specifically, truck-to-door assignment problem is observed in the outbound doors of a cross-dock facility. It is generally the case in practice that trucks are waiting to be loaded at the outbound area of a cross-dock facility. Nevertheless, it should be noted that, when it is assumed that all of the incoming trucks are available at the beginning of a planning period, this problem can be observed in the inbound doors as well.

It is assumed that each truck has a different loading time at each door, which is noted to represent operational characteristics at cross-dock facilities more realistically[6]. We refer the reader to the review by Shuib et al.[7] on quantitative approaches to the dock door assignment problems at cross-dock facilities. In this study, we consider controllable loading times for the cross-dock truck-to-door assignment problem. In particular, our focus is on the assignment of outgoing trucks to outbound doors. It is assumed that there is a resource that can be allocated to each door at the cross-dock facility. This resource can be the number of material handling equipment such as forklifts or the number of workers available at doors. Based on the number of resources allocated to each door, truck loading times at each door change. In particular, it is assumed that truck loading times at a specific door are non-increasing discrete functions of the number of resources allocated to that door. We consider that the flexible resource is the number of workers that can be assigned to each door for the loading process. Then the cross-dock operator needs to simultaneously determine the number of workers allocated to each door and the truck-to-door assignments.

We formulate a non-linear integer programming problem for the cross-dock operator's truck-to-door assignment problem with controllable loading times. The objective of the cross-dock operator's problem is the minimization of total labor costs required to load all of the trucks at the cross-dock facility. It is noted that the truck-to-door assignments are very important for the efficiency of cross-dock operations as well as for reducing labor hours[8]. In this study, both truck-to-door assignments and worker allocations at each outbound door are considered as the decision variables of the cross-dock operator. We propose two evolutionary heuristic methods for the problem. Numerical studies are conducted to compare the performance of these two solution methods.

The rest of the paper is organized as follows. In Section 2, the mathematical model is stated. Section 3 explains the solution methods. A set of numerical studies is conducted in Section 4 for comparing different solution methods. We conclude with a summary of our contributions and future research directions in Section 5.

2. Problem Formulation

Consider a cross-dock facility with n outbound doors and let each door be indexed by $i, i \in I$, such that $I = \{1,2,\dots,n\}$. There are a set of m empty trucks available at the cross-dock facility to be loaded at the outbound doors. Let each truck be indexed by $j, j \in J$, such that $J = \{1,2,\dots,m\}$. The loading time of a truck consists of moving the load to specific outbound door and loading it using material handling equipment or labor. Since outgoing trucks may require varying loads, truck loading times are distinct for each truck given that the same material handling resources are allocated for each truck. Furthermore, as each door may be of different importance for each truck (due to transferring its load from specific inbound doors), each truck has a different loading time at each door. Nonetheless, the cross-dock operator is capable of controlling each truck's loading time at a specific door by changing the number of material handling resources available at that door. In particular, we consider the number of workers as the material handling resource; however, forklifts can also be considered as material handling resources. Let k_i denote the number of workers allocated at door i such that $k_i = \{0,1,2,\dots\}$ $\forall i \in I$. Then, the loading time of truck j at door i as a discrete function of k_i, $h_{ij}(k_i)$, is defined as follows

$$h_{ij}(k_i) = \begin{cases} h_{ij}^{k_i} & if\ k_i \le l_j - 1, \\ h_{ij}^{l_j} & if\ k_i \ge l_j, \end{cases} \tag{1}$$

such that $h_{ij}^0 = \infty$, $h_{ij}^{k-1} \geq h_{ij}^k$ for $k \leq l_j$, and $h_{ij}^k = h_{ij}^{k+1}$ for $k \geq l_j$, where l_j denotes the maximum number of workers that can simultaneously load truck j. Defining $h_{ij}^0 = \infty$ means that any truck j cannot be served at door i if there is no worker available at door i, i.e., $k_i = 0$. Defining l_j captures the fact that when more than a specific number of workers are assigned for loading truck j, congestion is created and no reduction in truck j's loading time is observed. It is also important to note there that $h_{ij}(k_i)$ is not restricted to be a linearly decreasing function, i.e., it can be the case that $h_{ij}^{k-1} - h_{ij}^k \neq h_{ij}^k - h_{ij}^{k+1}$ for $k \leq l_j$. In practice, it would be expected that $h_{ij}^{k-1} - h_{ij}^k \geq h_{ij}^k - h_{ij}^{k+1}$. We continue our analysis with the more general definition of $h_{ij}(k_i)$ given in Equation (1).

Assuming that reducing loading times with no cost, nevertheless, is unrealistic as the cross-dock operator would, otherwise, allocate the maximum possible number of workers for each door to achieve the minimum loading time for each truck. This approach may maximize the labor costs. Particularly, we assume that each worker assigned to a door works at that door until the last truck completes its service. A cross-dock operator, in this setting, is responsible of assigning each truck to a door and allocate a specific number of workers at each door so as to minimize the total labor cost associated with the truck loading. Let us define

$$x_{ij} = \begin{cases} 1 & truck\ j\ is\ assigned\ to\ door\ i, \\ 0 & otherwise. \end{cases}$$

The total labor cost at door i then reads as $\alpha k_i \sum_{j \in J} x_{ij}\, h_{ij}(k_i)$, where α is the labor cost per worker per time unit. The cross-dock operator's optimization problem can be formulated as follows:

$$\textbf{(P)} \quad min \quad TLC(\boldsymbol{X}, \boldsymbol{k}) = \alpha k_i \sum_{j \in J} x_{ij}\, h_{ij}(k_i)$$
$$s.t. \quad \sum_{j \in J} x_{ij} = 1 \qquad \forall j \in J$$
$$x_{ij} \in \{0,1\} \qquad \forall i \in I, \forall j \in J$$
$$k_i \in \{0,1,2,3,\dots\} \qquad \forall i \in I$$

where $\boldsymbol{X}$ and $\boldsymbol{k}$ denote the $n \times m$ matrix of x_{ij} and n vector of k_i values, respectively, and $TLC(\boldsymbol{X}, \boldsymbol{k})$ is the total labour cost function. The objective minimizes the total labour cost required to serve all trucks. The first set of constraints assures that each truck is served at a door. The second and third sets of constraints define the binary and integer restrictions of the decision variables, respectively. In the reminder of the paper, we assume that $\alpha = 1$ per worker per unit time. This assumption is not restrictive as α is a constant multiplier of the objective function in the optimization problem. Next section discusses evolutionary methods to solve problem **(P)**.

3. Problem Formulation

It is worth noting that **(P)** is a non-linear integer programming problem. In particular, **(P)** is an NP-hard problem (the special case of **(P)**, when $h_{ij}(k_i)$ is linear, can be formulated as a quadratic assignment problem, which is known to be NP-hard[9]). In what follows, we, therefore, propose two evolutionary heuristic methods for **(P)**. In particular, we develop two genetic algorithms (GA), each of which has the same main steps: (i) chromosome representation and initialization, (ii) objective function evaluation, (iii) genetic operations, and (iv) termination. The GAs differ from each other in their chromosome representations, which consequently alter the executions of steps (ii) and (iii). The details of these GAs are explained next.

3.1. Assignment Based Genetic Algorithm

In the first GA we consider, the chromosomes are defined using the assignment decision variables of **(P)**, i.e., the binary x_{ij} variables; thus, we refer to this GA as the assignment based GA (A-GA). The details of the four aforementioned steps for A-GA are as follows.

i. ***Chromosome Representation:*** As noted previously, the binary assignment variables are used to represent chromosomes in A-GA. Figure 1 illustrates the chromosome representation of a given assignment matrix, where five incoming trucks (IT) are assigned to two inbound doors (ID). The population size in each generation is selected to be 3. Initially, we randomly generate a chromosome and mutate it twice as explained in step (iii).

ii. ***Objective Function Evaluation:*** The objective function value of a chromosome can be calculated as follows. Given the assignment matrix, one can enumerate over all possible numbers of workers for each door separately. For each door, the number of workers, which results in the minimum loading costs for that door, is selected. At the end of this process, k_i values are determined. Then, one can calculate the objective function value for a given chromosome using the k_i and x_{ij} values. The objective function values for each chromosome in a generation are calculated. After this, the chromosome with the minimum objective function value is selected as the parent of the next generation and genetic operations are applied on the parent chromosome to generate the new generation.

iii. ***Genetic Operations:*** In this step of A-GA, cross-over operations are not used since cross-over operations are observed to result in situations where a truck is assigned to two doors or a truck is not assigned to any door. That is, cross-over operations can result in infeasible assignments, which is computationally time consuming to construct a feasible assignment out of an infeasible assignment. Therefore, a chromosome is mutated using two mutation operations: swap and insert. In swap mutations, doors of two randomly selected trucks are exchanged. In insert mutations, a randomly selected truck's door is randomly changed to another door. For the chromosome given in Figure 1, in Figure 2, swap mutation, where trucks 1 and 4 exchange doors, and insert mutation, where truck 2's door is changed, are illustrated. Swap and insert mutations are used once on each parent chromosome to get the new generation.

iv. ***Termination:*** A-GA is terminated if there is no reduction in the objective function value of the 5,000 consecutive parent chromosomes.

Truck-to-Door Assignment

	IT 1	IT 2	IT 3	IT 4	IT 5
ID 1	1	0	1	0	0
ID 2	0	1	0	1	1

Chromosome

ID 1					ID 2				
IT 1	IT 2	IT 3	IT 4	IT 5	IT 1	IT 2	IT 3	IT 4	IT 5
1	0	1	0	0	0	1	0	1	1

Fig. 1. Chromosome Representation for A-GA

Swap Mutation

Before Swap						After Swap					
	IT 1	IT 2	IT 3	IT 4	IT 5		IT 1	IT 2	IT 3	IT 4	IT 5
ID 1	1	0	1	0	0	ID 1	0	0	1	1	0
ID 2	0	1	0	1	1	ID 2	1	1	0	0	1

Insert Mutation

Before Insert						After Insert					
	IT 1	IT 2	IT 3	IT 4	IT 5		IT 1	IT 2	IT 3	IT 4	IT 5
ID 1	1	0	1	0	0	ID 1	1	1	1	0	0
ID 2	0	1	0	1	1	ID 2	0	0	0	1	1

Fig. 2. Illustrations of Swap and Insert Mutations for A-GA

3.2. Resource Based Genetic Algorithm

In the second GA we consider, the chromosomes are represented using the resource decision variables of **(P)**, i.e., the integer k_i variables; thus, we refer to this GA as the resource based GA (R-GA). The details of the four aforementioned steps for R-GA are as follows.

i. ***Chromosome Representation and Initialization:*** Each chromosome is represented by a n vector of integer k_i values. Figure 3 illustrates a chromosome of R-GA with 5 inbound doors. Initially, 1,000 chromosomes are generated and their objective values are evaluated as explained in step (ii). In R-GA, since chromosome creation is relatively simpler compared to A-GA chromosome creation, a set of chromosomes are evaluated as an initialization process. The chromosome with the best objective function value is selected as the parent of the first generation.

ii. ***Objective Function Evaluation:*** The objective function value of a chromosome is calculated as follows. k_i values, one can determine $k_i h_{ij}^{k_i}$ $\forall i \in I, j \in J$. Then, $x_{ij} = 1$ when $i = \underset{i \in I}{\operatorname{argmin}}\{k_i h_{ij}^{k_i}\}$ $\forall j \in J$. After determining x_{ij} values, the objective function value for a given chromosome can be calculated. Similar to A-GA, the chromosome with the minimum objective function value is selected as the parent of the next generation and genetic operations are applied on the parent chromosome to generate the new generation.

iii. ***Genetic Operations:*** To obtain a new generation, we mutate the parent chromosome and apply cross-overs as follows. First, we mutate the parent chromosome using two mutations: we randomly select a door and increase or decrease the number of workers by one and, we randomly select two doors and exchange the number of workers at these doors. Figure 4 illustrates these two mutations for the chromosome given in Figure 3. At the end of mutations, there are three chromosomes, and, then, using these chromosomes, we execute cross-over operations. In particular, with 0.8 probability, we randomly pick two chromosomes and generate a chromosome by randomly selecting the number of workers assigned to each door from the two chromosomes considered for cross-over. At the end of this step, a population with at most 6 and at least three chromosomes are generated.

iv. ***Termination:*** R-GA is terminated if there is no reduction in the objective function value of the 5,000 consecutive parent chromosomes.

Workers Assigned to Inbound Doors

	ID 1	ID 2	ID 3	ID 4	ID 5
k_i	3	2	2	4	1

Fig. 3. Chromosome Representation for R-GA

Increase/Decrease Mutation

Before Increase						After Increase					
	ID 1	ID 2	ID 3	ID 4	ID 5		ID 1	ID 2	ID 3	ID 4	ID 5
k_i	3	2	2	**4**	1	k_i	3	2	2	3	1

Exchange Mutation

Before Exchange						After Exchange					
	ID 1	ID 2	ID 3	ID 4	ID 5		ID 1	ID 2	ID 3	ID 4	ID 5
k_i	3	**2**	2	4	**1**	k_i	3	1	2	4	2

Fig. 4. Illustrations of Increase/Decrease and Exchange Mutations for R-GA

4. Numerical Analysis

In this section, we conduct a set of numerical studies to compare the evolutionary methods. The GAs are coded in Matlab 2014. 8 different data sets, illustrated in Table 1, are considered and two problem instances are solved using each method in each data set.

Table 1. Data Set Specifications

Data Set	n	m	Data Set	n	m
1	5	50	5	15	150
2	5	100	6	15	200
3	10	50	7	20	150
4	10	100	8	20	200

The maximum number of workers that can simultaneously work on each truck at any door is set to be 5. The loading time generation for any truck at any door for specific number of workers is presented in Table 2.

Table 2. Handling Time Generation

h_{ij}^1	h_{ij}^2	h_{ij}^3	h_{ij}^4	h_{ij}^5
$U[30,60]$	$U[0.6h_{ij}^1, 0.4h_{ij}^1]$	$U[0.7h_{ij}^2, 0.6h_{ij}^2]$	$U[0.8h_{ij}^3, 0.7h_{ij}^3]$	$U[0.9h_{ij}^4, 0.8h_{ij}^4]$

It follows from Table 2 that loading time of each truck at each door when one worker is assigned to each door is random between 30 and 60 minutes. The reduction in loading times due to each additional worker is calculated randomly as in noted in Table 2. We note that loading times will decrease as the number of workers assigned to each door increases, however, it is possible that loading cost, i.e., $k_i h_{ij}^{k_i}$ values can be decreasing with additional workers and increasing with more additional workers. This enables us to capture the congestion effects. As noted by Shabay and Steiner[10], a linearly decreasing function for processing times is not realistic as it cannot capture the diminishing marginal values principle, which, for the problem of interest, implies that the rate of reduction in loading times decreases. Handling time representation in Table 2 obeys the diminishing marginal values principle. Furthermore, when loading costs considered, Table 2 can imply a convex (decreasing up to a point and then increasing) function, which, as noted before, captures the congestion effects.

In Table 3, we document the results of each solution method for each problem instance solved. In particular, objective function values (obj.), number of generations at termination (Gen.), and computation time in seconds (CPU) are given. The following observations are due to Table 3:

- On average, R-GA resulted in the best objective function value,
- R-GA outperformed A-GA for all of the problem instances solved,
- R-GA resulted in less computational time than A-GA on average.

Based on these observations, we can conclude that evolutionary methods such as the GAs discussed in this paper can be efficient tools to solve truck-to-door assignment problem with controllable loading times at cross-dock facilities. These methods are efficient in terms of computational times and they can result in high quality solutions. Furthermore, we suggest use of R-GA method as it is faster and gives better results compared to A-GA.

Table 3. Comparison of Solution Methods

		A-GA			R-GA		
Dataset	Instance	Obj.	Gen.	CPU	Obj.	Gen.	CPU
1	1	1659	7514	9	1659	5002	7
	2	1702	6297	8	1693	5001	8
2	1	3459	8895	11	3439	5001	13
	2	3271	7949	10	3271	5015	14
3	1	1543	9599	20	1543	5169	9
	2	1606	8141	17	1606	5048	9
4	1	3099	13766	30	3083	5066	17
	2	3135	14327	32	3131	5037	17
5	1	4536	24292	86	4524	5136	31
	2	4541	11622	37	4521	5622	35
6	1	5961	44018	179	5952	5125	40
	2	6132	23666	82	6099	5101	41
7	1	4492	17121	71	4467	5341	38
	2	4532	25462	112	4511	5894	43
8	1	6057	52445	294	6017	6379	57
	2	5993	31481	153	5968	5829	55
avg.		3857	19162	72	3843	5298	27

In what follows, we compare the makespan (the finish time of the last truck) at the doors for the solution found by each method. Table 4 documents the minimum, average, and maximum makespan of the solution found by each method for each problem instance. Based on makespan comparison, we can say that all of the solution methods give similar results. Particularly, the maximum makespans are reasonable and average makespans are very close to each other. Therefore, one can conclude that the evolutionary heuristic methods, additional to being fast and providing good objective function values, do not give unbalanced solutions in terms of makespans at the doors.

Table 4. Makespan Comparison of Solution Methods

		Min Makespan		Avg. Makespan		Max Makespan	
Dataset	Instance	A-GA	R-GA	A-GA	R-GA	A-GA	R-GA
1	1	90	90	133	133	210	210
	2	84	58	122	174	209	481
2	1	180	150	290	413	655	933
	2	111	173	278	280	680	636
3	1	24	24	110	110	287	287
	2	17	17	112	112	236	236
4	1	56	43	131	160	376	342
	2	49	57	180	180	413	381
5	1	42	50	145	155	369	331
	2	28	44	132	161	513	514
6	1	66	66	206	219	451	418
	2	70	68	237	232	675	650
7	1	35	44	121	134	444	408
	2	29	23	113	120	330	292
8	1	32	71	182	188	579	485
	2	35	28	175	171	463	493

5. Conclusions

In this study, a truck-to-door assignment at the outbound doors of cross-dock facility is analyzed. We allow resource flexibility in the loading of the outgoing trucks at the outbound doors. In particular, we assume that the number of workers allocated at each door is a decision variable of the cross-dock operator, that is, a discrete resource is considered. It is assumed that as the number of workers loading a truck increases, the truck loading time is non-increasing.

A non-linear integer programming problem is formulated for the cross-dock operator's truck-to-door assignment problem with controllable truck loading times. The objective of this problem is the minimization of total labor costs associated with the outbound loading by deciding on which truck will be loaded at which door and the how many of workers will be allocated at each door. Two evolutionary heuristic methods are proposed. A set of numerical studies are conducted to compare the solution methods. It is observed that the evolutionary methods are very efficient in terms of solution time and the resource based genetic algorithm finds better quality solutions compared to the assignment based genetic algorithm. Furthermore, the solutions achieved by each method are similar in terms of makespan at the doors.

Future research directions would be in analyzing truck-scheduling problem with resource flexibility. Furthermore, integration of inbound and outbound truck-to-door assignment with resource flexibility remains as a future research area.

Acknowledgements
This work is partially supported by the US Department of Transportation through the Center for Advanced Intermodal Technologies, a University-Transportation-Center led by the University of Memphis, Intermodal Freight Transportation Institute, and the Missouri University of Science and Technology. Any opinions, findings, and conclusions or recommendations expressed in this material do not necessarily reflect the views of the sponsors.

References

1. Gumus, M., Bookbinder, J. H. Cross-docking and its implications in location-distribution systems. Journal of Business Logistics 2004;25:199-228.
2. Kinnear, E.. Is there any magic in cross-docking? Supply Chain Management: An International Journal 1997; 2:49-52.
3. Waller, M. A., Cassady, C. R., Ozment, J. Impact of cross-docking on inventory in a decentralized retail supply chain. Transportation Research Part E 2006; 42:359-382.
4. Agustina, D., C.K.M. Lee, Piplani, R. A review: Mathematical modles for cross docking planning. International Journal of Engineering Business Management 2010; 2:47-54.
5. Belle, J. V., Valckenaers, P., Cattrysse, D. Cross-docking: State of the art. Omega 2012; 40 (6): 827-846.
6. Boysen, N., Fliedner, M., 2010. Cross dock scheduling: Classification, literature review and research agenda. Omega 2010; 38: 413-422.
7. Shuib, A., Ashikin, W. N., Fatthi, W. A. A review on quantitative approaches for dock door assignment in cross-docking. International Journal on Advanced Science, Engineering, Information Technology 2012; 2: 30-34.
8. Yu, V. F., Sharma, D., Murty, K. G. Door allocations to origins and destinations at less-than truckload trucking terminals. Journal of Industrial and Systems Engineering 2008; 2: 1-15.
9. Garey, M. R., Johnson, D. S. Computers and Intractability: A Guide to the Theory of NP Completeness. W. H. Freeman & Co., New York, NY, 1979.
10. Shabtay, D., Steiner, G. A survey of scheduling with controllable processing times. Discrete Applied Mathematics 2007; 155: 1643-1666.