Singapore Management University

Institutional Knowledge at Singapore Management University

Research Collection School Of Computing and Information Systems | School of Computing and Information Systems

8-2019

Integrated assignment and routing with mixed service mode cross-dock

Vincent YU

Aldy GUNAWAN
Singapore Management University, aldygunawan@smu.edu.sg

Eric I. JUNAIDI

Audrey T. WIDJAJA

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research

Part of the Artificial Intelligence and Robotics Commons

Citation

YU, Vincent; GUNAWAN, Aldy; JUNAIDI, Eric I.; and WIDJAJA, Audrey T.. Integrated assignment and routing with mixed service mode cross-dock. (2019). *14th International Congress on Logistics and SCM Systems ICLC 2019, Taipei, Taiwan, August 19-22*.
Available at: https://ink.library.smu.edu.sg/sis_research/4475

This Conference Paper is brought to you for free and open access by the School of Computing and Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Computing and Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email cherylds@smu.edu.sg.

Integrated Assignment and Routing with a Mixed Service Mode Cross-Dock

Vincent F. Yu, Eric Ivander Junaidi
Department of Industrial Management
National Taiwan University of Science & Technology, Taipei, Taiwan
E-mail: vincent@mail.ntust.edu.tw, ericivander55@gmail.com

Aldy Gunawan, Audrey Tedja Widjaja
School of Information Systems
Singapore Management University, Singapore
E-mail: aldygunawan@smu.edu.sg, audreyw@smu.edu.sg

Abstract. A mixed service mode cross-dock is a cross-dock facility that considers the use of flexible doors. Instead of having a specific task as an exclusive mode, each door can be used as a flexible door, either an inbound or an outbound door depending on the requirement. Having a mixed service mode cross-dock in an integrated assignment and routing problem is a new model in large field of cross-docking problems. Decisions that need to be made include doors' functionality, suppliers' assignments, customers' deliveries, and vehicles' routes with the objective of minimizing the total transportation and material handling costs. We develop a mathematical programming model and propose a Simulated Annealing (SA) algorithm to solve this new problem. Results from our own generated datasets show that our proposed SA is able to find all optimal solutions with lower computational times compared against those of commercial software, CPLEX. We further compare the total cost between a mixed service mode cross-dock and an exclusive service mode cross-dock. Our results show that the cost savings from using our strategy are as much as 1.05%.

Keywords: dock-door assignment, vehicle routing problem, cross-dock, simulated annealing

1. INTRODUCTION

As a development from a traditional distribution center, cross-dock is a facility that does not allow any inventory to be kept inside. Goods are only allowed to be placed inside the cross-dock for less than 24 hours with the purpose of a faster shipping procedure and zero inventory cost (Ladier and Alpan, 2016). A process called consolidation is a distinct characteristic that can easily differentiate cross-docks with other distribution centers. It replaces the process of storing and replenishing goods inside a regular distribution center. Cross-dock directly consolidates goods sent by suppliers and distributes them to customers according to their demands.

A retail industry is the most common industry to utilize a cross-dock facility. Walmart is a renowned example for cross-dock utilization (Stalk et al., 1992). Even though cross-dock is famous in the retail industry, other industries such as a distribution, postal, and manufacturing are also able to utilize a cross-dock (Boysen & Fliedner, 2010).

In this paper we extend the dock-door assignment and vehicle routing problem (DAVRP) (Enderer et al., 2017). In DAVRP, a truck that arrives at the cross-dock needs to be assigned to a particular inbound door. The commodities inside the truck are then unloaded, transferred to a particular outbound door, and delivered to the destinations (customers) based on their demand. The origins (suppliers) are assumed to use their own vehicles to deliver the commodities to the cross-dock, and therefore the vehicles' routes between origins and cross-dock do not need to be considered. Thus, there are two kinds of costs incurred in this problem: material handling and vehicle routing. In order to minimize these two costs, several decisions need to be made, such as the assignments of trucks coming from origins to which inbound door, the movement of commodities from inbound door to outbound door, the assignments of destinations to which outbound door, and the vehicle routes to deliver shipments from outbound doors to destinations.

Since one of the costs incurred is material handling cost, which is to transfer product from inbound to outbound doors, a new idea of using a mixed service mode cross-dock arose. In a mixed service mode cross-dock, a flexible door can be utilized as either an inbound or outbound door. This idea increases the flexibility inside the cross-dock since it can avoid any excessive movement of commodities diagonally, as shown in Figure 1. Thus, this research aims to develop a new problem in which utilizing a mixed service mode cross-dock in DAVRP and analyzes the results to see whether utilizing a mixed service mode cross-dock really benefits the company.

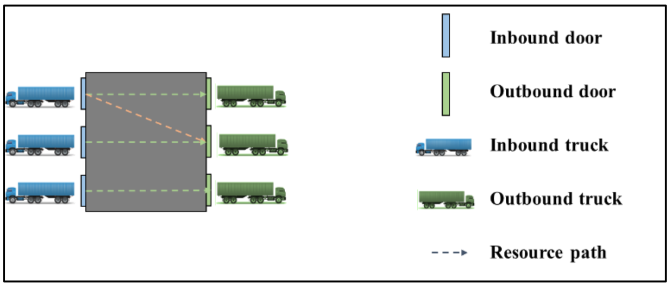

Figure 1: I-shaped cross-dock with exclusive service mode.

2. LITERATURE REVIEW

A typical cross-dock facility has two different sets of doors: inbound and outbound. The inbound doors are used to serve the incoming trucks that usually come from suppliers. The outbound doors are used for servicing the outgoing trucks that usually go to customers. In the I-shaped cross-dock (Figure 1), the inbound doors are placed on the left side of the cross-dock and the outbound doors are placed across the building, which is the right side of the cross-dock. In this setting, the cross-dock service mode is classified as an exclusive mode where the doors are already set to specific functions.

There are three more service modes other than the exclusive mode: mixed, exclusive, and given (Boysen and Fliedner, 2010). A mixed service mode allows each door to be used as either an inbound door or an outbound door. The purpose of this setting is to have the flexibility that an exclusive mode does not have. When a cross-dock has some doors functioning as inbound doors, some as outbound doors, and some as either inbound or outbound doors (i.e. flexible doors), the service mode is called the exclusive mixed mode. The last service mode is called a given mode, whereby the function of each door is made according to the truck's destination.

Bartz-Beielstein et al. (2006) proposed the usage of inbound door, outbound door, and flexible door. In this model, the multi-functional door can be used as both inbound and outbound doors. Bozer and Carlo (2008) tackled the same problem on a larger scale. Shakeri et al. (2008) developed a generic mathematical model for a flexible door in the assignment problem with the truck scheduling problem. The purpose is to provide a model that is generic enough to be extended to various types of cross-docking problems. Shakeri et al. (2010a) published a research with the previous built model as their baseline. Shakeri et al. (2010b) developed two heuristics for the assignment problem: a dependency ranking (DR) heuristic for truck sequencing and machine fitness (MF). Berghman et al. (2015) and Bodnar et al. (2015) studied the benefit of using a mixed service mode setting for a cross-dock. The result is based on a comparison against an exclusive mode as well as a mixed mode.

Solving the integrated version of routing and scheduling creates an opportunity to minimize the total cost. Mousavi et al. (2013) solved a multiple location routing problem with fuzzy environment for an integrated scheduling routing cross-docking problem. Dondo and Cerdá (2015) solved a mixed vehicle fleet assignment and scheduling at the cross-dock. Enderer et al. (2017) proposed a model to minimize the total of handling cost and routing cost of a scheduling routing problem.

3. PROBLEM DEFINITION

Let $O = \{1,2,\dots,|O|\}$ be a set of origins, $N^+ = \{0,1,2,\dots,|N|\}$ representing a set of destinations and the cross-dock (noted as 0), $N = N^+ \backslash \{0\}$, and $D = \{1,2,\dots,|D|\}$ represents a set of the dock-doors. Each destination n has its own demand de_n. Each door $d \in D$ has a capacity of cap, and the handling cost C_{ij} occurs for transferring a commodity from door i to door j. A vehicle needs to travel a distance of t_{ab} from node a to node b $(a,b \in N^+)$ under a constant cost per unit distance π. We utilize CO_{no} for mathematical model formulation purposes only, in which 1 indicates that commodity n is sent/provided by origin o and 0 otherwise.

Each origin has demand for exactly one type of commodity, and each commodity is provided by one origin. Thus, one origin may provide more than one commodity. The cross-dock is I-shaped, which means that half of the doors are located on the left side of the cross-dock, and the other half are located on the right side of the cross-dock.

The decision variables in this mathematical model are listed below.

- X_{od} : a binary variable set to 1 if origin o is assigned to door d and 0 otherwise $(o \in O, d \in D)$.
- Y_{ijn} : a binary variable set to be 1 if commodity n is moved from door i to door j and 0 otherwise $(i,j \in D, n \in N)$.
- Z_{abd} : a binary variable set to be 1 if there is a vehicle that departs from door d travels from destination a to destination b and 0 otherwise $(d \in D,\ a,b \in N^+)$.
- Q_{abd} : total load of the commodity remaining in a vehicle departing from door d that travels from destination a to destination b $(d \in D,\ a,b \in N^+)$.
- q_{nd} : amount of commodity n delivered by a vehicle that departs from door d $(n \in N, d \in D)$.
- P_d : a binary variable set to be 1 if door d is assigned to serve as an inbound door and 0 otherwise $(d \in D)$.
- R_d : a binary variable set to be 1 if door d is assigned to

serve as an outbound door and 0 otherwise $(d \in D)$.

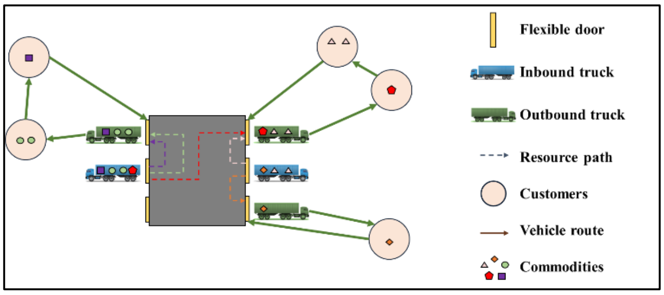

Figure 2: Integrated assignment and routing with mixed service mode cross-dock.

$$\min \sum_{n\in N}\sum_{i\in D}\sum_{j\in D} C_{ij} de_n Y_{ijn} + \sum_{d\in D}\sum_{a\in N^+}\sum_{b\in N^+} t_{ab}\pi Z_{abd} \tag{1}$$

$$\sum_{o\in O} X_{od} \geq 1 - M(1-P_d) \ \forall d\in D \tag{2}$$

$$MP_d \geq \sum_{o\in O} X_{od} \ \ \forall d \in D \tag{3}$$

$$\sum_{b\in N} Z_{0bd} \geq 1 - M(1-R_d) \ \forall d\in D \tag{4}$$

$$MR_d \geq \sum_{b\in N} Z_{0bd} \ \ \forall d \in D \tag{5}$$

$$P_d + R_d \leq 1 \ \forall d \in D \tag{6}$$

$$\sum_{d\in D} X_{od} = 1 \ \forall o \in O \tag{7}$$

$$\sum_{n\in N}\sum_{o\in O} X_{od} CO_{no} de_n \leq cap \ \forall d \in D \tag{8}$$

$$\sum_{o\in O} X_{oi} CO_{no} = \sum_{j\in D} Y_{ijn} \ \ \forall i \in D, \forall n \in N \tag{9}$$

$$\sum_{i\in D}\sum_{n\in N} Y_{ijn} de_n \leq cap \ \forall j \in D \tag{10}$$

$$Q_{abd} \leq cap \times Z_{abd} \ \ \forall a,b \in N^+, \forall d \in D \tag{11}$$

$$\sum_{d\in D}\sum_{a\in N^+} Z_{abd} = 1 \ \forall b \in N \tag{12}$$

$$\sum_{b\in N} Z_{0bd} \leq 1 \ \forall d \in D \tag{13}$$

$$\sum_{a\in N^+, a\neq n} Z_{and} - \sum_{b\in N^+, b\neq n} Z_{nbd} = 0 \ \forall n \in N^+, \forall d \in D \tag{14}$$

$$\sum_{i\in D} Y_{ijn} = \sum_{a\in N^+} Z_{anj} \ \ \forall j \in D, \forall n \in N \tag{15}$$

$$\sum_{a\in N^+} Q_{and} - \sum_{b\in N^+} Q_{nbd} = q_{nd} \ \ \forall n \in N, \forall d \in D \tag{16}$$

$$q_{nd} = de_n \sum_{a\in N^+} Z_{and} \ \ \forall d \in D, \forall n \in N \tag{17}$$

$$X_{od} \in \{0,1\} \ \forall o \in O, \forall d \in D \tag{18}$$

$$Y_{ijn} \in \{0,1\} \ \forall i, j \in D, \forall n \in N \tag{19}$$

$$Z_{abd} \in \{0,1\} \ \forall a,b \in N^+, \forall d \in D \tag{20}$$

$$Q_{abd} \geq 0 \ \forall a,b \in N^+, \forall d \in D \tag{21}$$

The objective function is to minimize the total material handling and vehicle routing costs formulated in Equation (1). Equations (2) and (3) enforce P_d equals to 1 if there are any origins assigned to door *d*. Equations (4) and (5) enforce R_d equals to 1 if there are any destinations assigned to door *d*. Equation (6) makes sure that each door only performs a maximum of one task (as an inbound door or as an outbound door). Equation (7) ensures that each origin is assigned to exactly one door (in which the door is then regarded as an inbound door). Equation (8) ensures the inbound door capacity limitation. Equation (9) ensures that all commodities in inbound doors are transferred to outbound doors. Equations (10) and (11) ensure the outbound door does not exceed its capacity.

Equation (12) ensures that each destination is visited exactly once by a vehicle that departs from an outbound door. Equation (13) ensures every vehicle only leaves the cross-dock at most once (multiple trips are not allowed). Equation (14) makes sure that if a vehicle visits a node (destination), then it will also leave that node. Equation (15) ensures all commodities in every outbound door are delivered to the destinations. Equations (16) and (17) ensure that every destination receives an amount of commodities according to their demand. Equations (18) to (21) limit the domain of the decision variables.

4. PROPOSED ALGORITHM

4.1 Solution Representation

In this problem we use a solution representation that consists of two parts: a door solution to determine the function of each door and a main solution to determine the assignment of origins to the inbound doors, destinations to the outbound doors, and the corresponding vehicles' routes. A door solution consists of a permutation of $|D|$ doors represented by numbers $(|O|+|N|+1,\ldots,|O|+|N|+|D|)$ and a "-1" to separate the doors' function. Doors placed at the left side of "-1" are treated as the inbound doors, while doors placed at the right side of "-1" are treated as the outbound door.

A main solution consists of a permutation of $|O|$ origins, $|N|$ destinations, and $|D|$ doors, represented by numbers $(1,\ldots,|O|)$, $(|O|+1,\ldots,|O|+\ |N|)$, and $(|O|+|N|+1,\ldots,|O|+|N|+|D|)$, respectively. Each origin/destination is assigned to an inbound/outbound door that is placed on its right side. If there is any origin/destination that does not have an inbound/outbound door on its right side, then it will be placed to an inbound/outbound door on its left side.

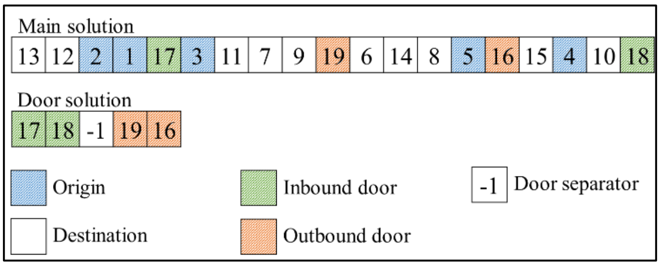

Figure 3: The main solution and the door solution for the simulated annealing algorithm.

Figure 3 illustrates a solution representation. There are 5 origins, 10 destinations, and 4 doors. The door solution consists of numbers 16 to 19, representing doors 1 to 4, respectively, with -1 as the separator. The main solution consists of numbers 1 to 5 (origins 1-5); numbers 6 to 15 (destinations 1-10); and numbers 16 to 19 (doors 1-4). Doors 2 and 3 (numbers 17 and 18 in Figure 3) are the inbound doors, while doors 4 and 1 (numbers 19 and 16 in Figure 3) are the outbound doors. Origins 1 and 2 (numbers 1 and 2 in Figure 3) are assigned to door 2 (number 17 in Figure 3), while origins 3, 5, and 4 (numbers 3, 5, and 4 in Figure 3) are assigned to door 3 (number 18 in Figure 3). Destinations 8, 7, 6, 2, and 4 (numbers 13, 12, 11, 7, and 9 in Figure 3) are served by a truck from door 4 (number 19 in Figure 3), while destinations 1, 7, 3, 10, and 5 (numbers 6, 14, 8, 15, and 10 in Figure 3) are served by a truck from door 1 (number 16 in Figure 3).

4.2 Simulated Annealing (SA)

We propose SA with three neighborhood moves: swap, insert, and inverse. Figure 4 illustrates how the moves are applied to the main solution and the door solution. Swap is performed by selecting two random points and then exchanging their positions. Insert is performed by moving the position of the second random point before the first random point. Inverse is performed by reversing the sequence between two random points, including those two random points.

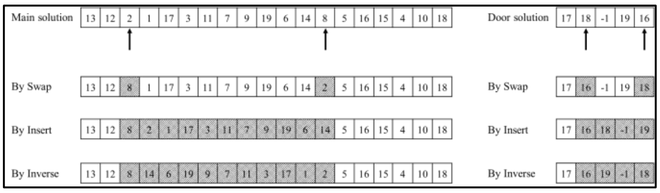

Figure 4: SA moves.

Five parameters are used to construct our proposed SA: T_0, $\propto$, MAXINNERLOOP, LIMIT, and LIMITINT. T_0 is the initial temperature; $\propto$ is a coefficient for reducing the temperature; MAXINNERLOOP is the number of iterations in each temperature; LIMIT is the number of successive temperature reductions before SA is terminated; and. LIMITINT is the number of successive non-improved global best solution, such that the next solution is generated from the best found solution so far instead of the previous accepted solution. Let S_0, S^*, and S' denote the current solution, best found solution so far, and the starting solution at each iteration, respectively.

In order to construct the initial solution for the door solution, we assign the first half of the doors as the inbound doors and the rest as the outbound doors. For the initial solution of the main solution, we follow a greedy manner. An origin with the highest total load is assigned to the first inbound door. We continue by finding subsequent origins as long as the capacity is enough; otherwise, we repeat for subsequent doors. The process of assigning the destinations to the outbound doors follows the same approach, which is based on their demand load.

At the beginning, as there are no other solutions, we set S^* and S' equal to S_0. $Temp$ is set to be equal to T_0 and will be reduced after MAXINNERLOOP. In every iteration, one of the SA moves is chosen randomly to generate a new solution. The probability is set to be equally likely in the beginning and will be changed over time depending on the result. The better the result is, the higher is the probability given to that particular move. The probability is updated every MAXINNERLOOP by using Equations (22) and (23), where S is a set of neighborhood moves S = *{swap, insert, inverse}*, P_s is the probability of the s^{th} neighborhood $(s \in S)$, f_s is the average fitness value of the s^{th} neighborhood $(s \in S)$, Acc_s

is the number of the s^{th} neighborhood being used $(s \in S)$, and $Obj(\gamma)$ is the fitness value of solution γ.

$$P_s = f_s / \sum_{k=1}^{3} f_k \quad \forall s \in S \tag{22}$$

$$f_s = \sum_{k=1}^{Acc_s} \frac{1.0}{Obj(\gamma)} / \text{Acc}_s \quad \forall s \in S \tag{23}$$

Every time a move is selected, we continue to calculate the objective value following Equation (1). Next, we calculate the objective value difference between S_0 and S'. Since the problem is categorized as a minimization problem, a negative value means we have a better solution, and so we update S'; otherwise, we update S' with probability $\exp(-\delta/Temp)$. If S_0 is also better than S^*, then we update S^* and set NoImprInt equal to 0; otherwise, we increase it by one. When LimitInt is reached, we set S' to be equal to S^* such that the solution in the next iteration is generated from S^*. SA is terminated when NoImpr reaches Limit.

Simulated Annealing

```
S0 ← Initial solution construction
S* ← S0
S' ← S0
Temp ← T0
NoImpr ← 0
NoImprInt ← 0
Pswap ← 1/3, Pinsert ← 1/3, Pinverse ← 1/3
while NoImpr < Limit do
   InnerLoop ← 0
   FoundBestSol ← false
      while InnerLoop < MaxInnerLoop do
         S0 ← swap/insert/inverse
         δ ← obj value of S0 - obj value of S'
         if δ < 0 then
            S' ← S0
            if S0 < S* then
               S* ← S0
               FoundBestSol ← true
               NoImpr ← 0
               NoImprInt ← 0
            else
               NoImprInt ← NoImprInt + 1
            end if
         else
            r ← rand[0,1]
            if r < exp(−δ/Temp) then
               S' ← S0
            else
               S0 ← S'
            end if
            NoImprInt ← NoImprInt + 1
         end if
            if NoImprInt > LimitInt then
               S0 ← S*
               S' ← S0
               NoImprInt ← 0
            end if
         InnerLoop ← InnerLoop + 1
      end while
      Temp ← Temp ×∝
      Calculate Ps and fs using Eq. (22) and (23)
      if FoundBestSol = false then
         NoImpr ← NoImpr + 1
      end if
   end while
   return S*
```

5. EXPERIMENTAL RESULTS AND DISCUSSIONS

5.1 Benchmark Instances and Parameter Selection

Since the benchmark instances are not available online (Enderer et al., 2017), we herein generate the instances. We follow Solomon (1987) VRPTW instances for the cross-dock's and destinations' locations (x and y coordinates), destinations' demand, and door capacity (which is vehicle capacity in Solomon's instances). We differentiate the problems as small, medium, and large problems based on the number of destinations. Table 1 summarizes the instances. Other generated parameters are as follows. The assignment of which commodity is provided by which origin done one by one in an order according to the origin number; vehicle traveling cost (π) = 1 (Enderer et al., 2017); Handling cost (C_{ij}) is calculated by following Equation (24) in which both α and β are set to 4, i is a set of doors located on the left side of the cross-dock, and j is a set of doors located on the right side of the cross-dock (Guignard et al., 2012). It is possible to move the commodities between doors that are located on the same side of the cross-dock; C_{ii} and C_{jj} are then calculated by the Pythagoras theorem (Figure 5).

$$C_{ij} = \alpha + \beta \times |i - j| \tag{24}$$

Table 1: Summary of instances.

Parameter	Small		Medium		Large	
$\|N\|$	10	15	25	50	75	100
$\|O\|$	5	10	15	20	25	30
$\|D\|$	6	6	16	16	28	28
cap	200	200	200	200	200	200
Total problem (r1,rc1,c1,c2)	4 × 10	4 × 6	4 × 4	4 × 2	4 × 1	4 × 1

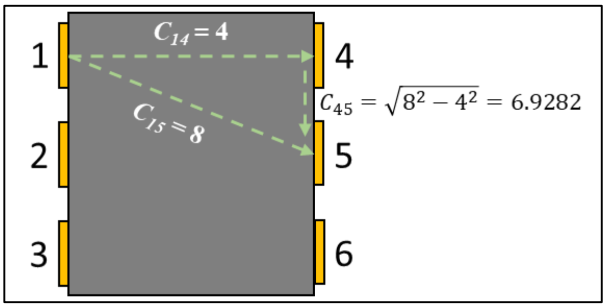

Figure 5: Calculation of handling cost.

We run all experiments on a PC with Intel Core i7-6700 CPU @ 3.40 GHz processor, 16.0 GB RAM, with SA coded in Microsoft Visual Studio C++ 2017 and the mathematical model solved in CPLEX 12.8.0.0. We first run the SA for the mixed service mode cross-dock and compare the results to CPLEX. Next, we solve the exclusive mode cross-dock using SA and compare the results between mixed and exclusive mode cross-dock.

Table 2: SA parameters.

Parameter	Values
T_0	{ 5, 10, **15** }
$\propto$	{ 0.8, 0.9, **0.99** }
MAXINNERLOOP	{ 1000, 3000, **5000** }
LIMIT	{ 50, 100, **200** }
LIMITINT	{ 50000, 100000, **200000** }

To decide the SA parameters, we set up a full factorial design with five factors: T_0, $\propto$, MAXINNERLOOP, LIMIT, and LIMITINT, with each having three levels. Table 2 lists the parameter values, with bold values indicating the selected one. We use this full factorial design upon 10 randomly selected problems, with 5 replications.

5.2 Computational Results

We limit our computational experiments to solve small instances in this paper. Table 3 shows the results. Column 1 indicates the problem name. Columns 2-6 report the results for the mixed service mode cross-dock. Columns 2 and 3 are the CPLEX results that represent cost and CPU time, respectively. Columns 4 and 5 are the SA's best results from 5 replications for cost and CPU time, respectively. Column 6 presents the gap calculation between the SA and CPLEX results, calculated by Equation (25). Columns 7 and 8 are the SA's best results (cost and CPU time, respectively) from 5 replications when solving the exclusive mode cross-dock. Column 9 presents the gap calculation between the cost of using mixed service mode cross-dock and the cost of using exclusive service mode cross-dock. Equation (26) calculates the gap with a negative value indicating that the cost of using mixed service mode cross-dock is lower than the one of the exclusive service mode cross-dock.

CPLEX can obtain the optimal solution for all problems, and SA is able to have the same results as CPLEX does for all problems at faster computational times. When comparing the costs of using the mixed service mode cross-dock and the exclusive service mode cross-dock, even though for most of the cases there is no difference in terms of cost, there is still some savings up to 1.05% that we can achieve for some cases. These experimental results prove that it is worth it for a company to try to use a mixed service mode cross-dock instead of using an exclusive service mode cross-dock, because it can save some costs and further increase profits.

$$Gap = \frac{Cost_{SA} - Cost_{CPLEX}}{Cost_{CPLEX}} \times 100\% \qquad (25)$$

$$Gap = \frac{Cost_{mixed} - Cost_{exclusive}}{Cost_{exclusive}} \times 100\% \qquad (26)$$

6. CONCLUSION AND FUTURE RESEARCH

This research proposes a new model in the cross-docking industry, which is the integrated assignment routing problem in a mixed service mode cross-dock. We propose an SA algorithm with an adaptive neighborhood and an intensification strategy. The results are also compared with those of the exclusive service mode cross-dock. For the instances of mixed service mode cross-dock, our proposed SA is able to obtain all 64 optimal solutions.

When comparing the mixed service and exclusive service modes, 4 out of 64 problems are better off being solved in a mixed service mode environment. By comparing the total cost incurred in both service modes, we see that using a mixed service mode cross-dock obtains a lower cost compared against the one of the exclusive mode cross-dock. The cost savings from using a mixed service mode environment are as high as 1.05%. We will extend the experiments to medium and large instances in future.

Table 3: Results of small instances.

Problem	Mixed Service Mode					Exclusive Service Mode		Gap Mixed to Exclusive
	CPLEX		SA		Gap SA to CPLEX	SA		
	Cost	CPU (s)	Cost	CPU (s)		Cost	CPU (s)	
10c1-1	655.29	21.153	655.29	5.37	0.00%	655.29	6.527	0.00%
10c1-2	1010.24	21.044	1010.24	5.716	0.00%	1010.24	4.873	0.00%
10c1-3	689.41	4.415	689.41	5.621	0.00%	689.41	4.902	0.00%
10c1-4	1007.74	75.333	1007.74	5.429	0.00%	1018.46	4.523	-1.05%
10c1-5	577.68	56.394	577.69	4.985	0.00%	577.69	5.895	0.00%
10c1-6	1070.85	66.737	1070.85	4.932	0.00%	1070.85	4.405	0.00%
10c1-7	818.68	45.739	818.68	5.521	0.00%	818.68	5.452	0.00%
10c1-8	828.7	74.911	828.7	4.745	0.00%	828.7	4.404	0.00%
10c1-9	875.92	33.15	875.92	5.628	0.00%	875.92	4.932	0.00%
10c1-10	902.63	48.173	902.63	5.097	0.00%	902.63	4.38	0.00%
10c2-1	733.13	29.577	733.13	4.817	0.00%	733.13	4.358	0.00%
10c2-2	1010.24	103.522	1010.24	4.996	0.00%	1010.24	4.493	0.00%
10c2-3	730.95	193.597	730.95	5.233	0.00%	730.95	4.401	0.00%
10c2-4	1031.01	70.387	1031.01	6.095	0.00%	1039.61	4.436	-0.83%
10c2-5	614.18	80.512	614.18	6.348	0.00%	614.18	5.14	0.00%
10c2-6	1069.25	70.934	1069.25	4.986	0.00%	1069.25	5.08	0.00%
10c2-7	827.9	52.744	827.9	4.918	0.00%	827.9	4.8	0.00%
10c2-8	828.68	69.889	828.68	4.729	0.00%	828.68	4.473	0.00%
10c2-9	887.24	48.719	887.24	5.317	0.00%	887.24	4.684	0.00%
10c2-10	923.81	92.041	923.81	5.032	0.00%	923.81	4.554	0.00%
10r1-1	669.04	127.187	669.04	4.926	0.00%	669.04	4.349	0.00%
10r1-2	749.48	81.604	749.48	4.69	0.00%	749.48	4.294	0.00%
10r1-3	725.28	62.618	725.28	4.831	0.00%	725.28	4.732	0.00%
10r1-4	879.55	113.912	879.55	4.741	0.00%	879.55	4.48	0.00%
10r1-5	828.07	74.116	828.07	4.706	0.00%	828.07	4.695	0.00%
10r1-6	627.22	80.902	627.22	5.093	0.00%	627.22	4.66	0.00%
10r1-7	935.62	64.912	935.62	4.713	0.00%	935.62	4.343	0.00%
10r1-8	670.55	42.526	670.55	4.758	0.00%	670.55	4.443	0.00%
10r1-9	921.71	28.72	921.71	4.86	0.00%	921.71	4.447	0.00%
10r1-10	581.01	111.884	581.01	4.945	0.00%	581.01	4.552	0.00%
10rc1-1	1086.51	55.63	1086.51	4.884	0.00%	1086.51	4.409	0.00%
10rc1-2	1068.58	74.319	1068.58	5.546	0.00%	1068.58	4.34	0.00%
10rc1-3	935.76	48.454	935.76	4.676	0.00%	935.76	4.27	0.00%
10rc1-4	1094.44	73.571	1094.44	4.96	0.00%	1094.44	4.621	0.00%
10rc1-5	814.7	44.335	814.7	4.773	0.00%	814.7	4.257	0.00%
10rc1-6	824.19	99.966	824.19	4.905	0.00%	824.19	4.297	0.00%
10rc1-7	729.97	74.663	729.97	4.838	0.00%	729.97	4.395	0.00%
10rc1-8	891.16	105.394	891.16	4.694	0.00%	891.16	4.251	0.00%
10rc1-9	771.99	66.721	771.99	4.894	0.00%	771.99	4.495	0.00%
10rc1-10	764.61	57.112	764.61	4.866	0.00%	764.61	4.466	0.00%
15c1-1	1176.86	402.514	1176.86	14.01	0.00%	1176.86	10.614	0.00%
15c1-2	1178.26	476.411	1178.26	18.184	0.00%	1178.26	13.09	0.00%
15c1-3	1263.94	1040.74	1263.94	10.909	0.00%	1263.94	7.581	0.00%
15c1-4	1352.7	919.158	1352.7	13.764	0.00%	1352.7	10.553	0.00%
15c1-5	1322.21	766.917	1322.21	11.521	0.00%	1322.21	10.542	0.00%

15c1-6	1196.9	335.137	1196.9	9.146	0.00%	1196.9	8.604	0.00%
15c2-1	1247.61	146.469	1247.61	10.839	0.00%	1247.61	9.252	0.00%
15c2-2	1201.81	313.484	1201.81	10.088	0.00%	1201.81	8.849	0.00%
15c2-3	1285.65	642.178	1285.65	8.669	0.00%	1285.65	8.897	0.00%
15c2-4	1361.5	1290.17	1361.5	13.357	0.00%	1361.5	10.467	0.00%
15c2-5	1335.26	730.817	1335.26	12.638	0.00%	1335.26	10.041	0.00%
15c2-6	1218.27	367.663	1218.27	10.605	0.00%	1218.27	10.065	0.00%
15r1-1	1090.72	415.883	1090.72	12.683	0.00%	1090.72	8.792	0.00%
15r1-2	1062.4	322.704	1062.4	9.357	0.00%	1064.55	7.254	-0.20%
15r1-3	1122.82	221.786	1122.82	9.047	0.00%	1122.82	7.437	0.00%
15r1-4	1141.49	409.924	1141.49	9.473	0.00%	1142.66	9.201	-0.10%
15r1-5	1296.11	419.456	1296.11	8.983	0.00%	1296.11	8.953	0.00%
15r1-6	1235.82	260.741	1235.82	10.114	0.00%	1235.82	8.481	0.00%
15rc1-1	1530.67	399.206	1530.67	9.832	0.00%	1530.67	11.871	0.00%
15rc1-2	1519.88	180.914	1519.88	8.836	0.00%	1519.88	7.79	0.00%
15rc1-3	1412.25	302.486	1412.25	9.669	0.00%	1412.25	8.384	0.00%
15rc1-4	1226.56	252.503	1226.56	8.112	0.00%	1226.56	6.956	0.00%
15rc1-5	1064.1	350.706	1064.1	7.612	0.00%	1064.1	6.576	0.00%
15rc1-6	1294.56	71.355	1294.56	7.93	0.00%	1294.56	6.746	0.00%

ACKNOWLEDGMENTS

This research is partially supported by the Ministry of Science and Technology of the Republic of China (Taiwan) under grant MOST 106-2410-H-011-002-MY3.

REFERENCES

Bartz-Beielstein, T., Chmielewski, A., Janas, M., Naujoks, B., & Scheffermann, R. (2006) Optimizing door assignment in LTL-terminals by evolutionary multiobjective algorithms. *IEEE Congress on Evolutionary Computation.*

Berghman, L., Briand, C., Leus, R., & Lopez, P. (2015) The truck scheduling problem at crossdocking terminals-exclusive versus mixed mode. *4th International Conference on Operations Research and Enterprise Systems (ICORES 2015).*

Bodnar, P., de Koster, R., & Azadeh, K. (2015) Scheduling trucks in a cross-dock with mixed service mode dock doors. *Transportation Science*, **51**, 112-131.

Boysen, N., & Fliedner, M. (2010) Cross dock scheduling: Classification, literature review and research agenda. *Omega*, **38**, 413-422.

Bozer, Y. A., & Carlo, H. J. (2008) Optimizing inbound and outbound door assignments in less-than-truckload crossdocks. *IIE Transactions*, **40**, 1007-1018.

Dondo, R., & Cerdá, J. (2015) The heterogeneous vehicle routing and truck scheduling problem in a multi-door cross-dock system. *Computers & Chemical Engineering*, **76**, 42-62.

Enderer, F., Contardo, C., & Contreras, I. (2017) Integrating dock-door assignment and vehicle routing with cross-docking. *Computers & Operations Research*, **88**, 30-43.

Guignard, M., Hahn, P. M., Pessoa, A. A., & da Silva, D. C. (2012) Algorithms for the cross-dock door assignment problem. *Proceedings of the Fourth International Workshop on Model-Based Metaheuristics.*

Ladier, A.-L., & Alpan, G. (2016) Cross-docking operations: Current research versus industry practice. *Omega*, **62**, 145-162.

Mousavi, S. M., Tavakkoli-Moghaddam, R., & Jolai, F. (2013) A possibilistic programming approach for the location problem of multiple cross-docks and vehicle routing scheduling under uncertainty. *Engineering Optimization*, **45**, 1223-1249.

Shakeri, M., Low, M. Y. H., & Lee, E. W. (2010a) Development of a novel resource-constrained crossdocking model for the truck scheduling problem. *IEEE 15th Conference on Emerging Technologies and Factory Automation (ETFA 2010), Bilbao, 2010, pp. 1-9.*

Shakeri, M., Low, M. Y. H., & Li, Z. (2008) A generic model for crossdock truck scheduling and truck-to-door assignment problems. *6th IEEE International Conference on Industrial Informatics, Daejeon, 2008, pp. 857-864.*

Shakeri, M., Low, M. Y. H., Li, Z., & Lee, E. W. (2010b) Two efficient constructive heuristics for scheduling trucks at crossdocking terminals. *Proceedings of 2010 IEEE International Conference on Service Operations and Logistics, and Informatics, Qingdao, Shandong, 2010, pp. 177-182.*

Solomon, M. M. (1987) Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, **35**, 254-265.

Stalk, G., Evans, P., & Shulman, L. E. (1992) Competing on capabilities: the new rules of corporate. *Harvard Business Review*, **5**, 35-45.