Resource Scheduling Problem in Distribution Center

C. Hamontree¹, S. Prompakdee² and J. Koiwanit³
Department of Industrial Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand

Corresponding author: Chaowalit.ha@kmitl.ac.th

Abstract. Due to the rapid growth of the retail shop trend recently, the Distribution Center (DC) or warehouse had become to be a vital part in the supply chain in order to store goods from multiple suppliers and deliver to multiple retailers. Cross docking is a relatively the logistics technique used in the DC in order to move materials from inbound trailers to outbound trailers as quickly as possible. From the following problem, an ineffectual DC management leads to the resources utilization problem such as machine, forklift, industrial trucks, etc. The main objective of this research is to study and solve the scheduling problem of the usage of the forklift in the Cross docking of DC and to manage the inbound and outbound docks in order to minimize Tardiness jobs, and also to maximize the efficiency of machine utilization. This study has provided the scheduling, comparison from genetic algorithm and heuristic method. As the result came out we found that the forklift scheduling by the genetic algorithm method is given the better management result better than heuristic method.

1. Introduction
Materials handling is a central concern in storage space decisions and is mostly a cost-consuming activity. The objectives for materials handling are therefore cost focused and attempt to reduce handling costs while increasing space utilization. Materials handling can be improved through good load utilization, space layout and equipment choices. Typically, storage and order picking operations take up the bulk of handling activity in a warehouse. These operations include stock locating, stock arrangement, product sequencing, order splitting and item batching, all of which are labour intensive and expensive. Further, warehouses need to be configured to handle equipment and storage facilities to accommodate these activities and an inventory management system has to be in place. Cross-docking attempts to lessen or eliminate these burdens [6].

2. Research Theories

2.1. Cross Docking Problem
There are four major functions in a traditional warehouse receiving, storage, order picking, and shipping. The middle two functions are typically the most costly storage because of inventory holding costs; and order picking because it is labour intensive. Cross-docking is a logistics technique to transfer shipments directly from incoming to outgoing locations without storage in between. In a traditional DC, the warehouse maintains stock until a customer orders, then the product is picked, packed, and shipped. When replenishments arrive at the DC, they are stored until a customer is identified. In a cross-docking
model, the customer is known before the product gets to the warehouse and there is no need to move it to storage [6]. In this case, cross-docking is very much like Just-In-Time (JIT) manufacturing, which is most viable when demand has low variance and there is high enough volume to justify frequent setups.

![Figure 1. Typical Cross-Docking Flow](image)

2.2. Scheduling Problem

It has been specified as “when labour, equipment and facilities are needed to produce a product or provide a service” [5]. Scheduling work on machines in a large manufacturing system is an extremely challenging task. A lot of research on this topic has illustrated poor performance in scheduling had brought numerous problems, such as failures to expedite jobs required by downstream work centers, long throughput time, high level of work in process inventory, excessive overtime and other operational costs and so on [3], [4]. Therefore, scheduling is one of the core activities in cross-docking environment.

2.2.1. Machine Scheduling Problem. The study of earliness and tardiness penalties in scheduling models is a relatively recent area of inquiry. For many years, scheduling research focused on single performance measures, referred to as regular measures, which are non-decreasing in job completion times. Most of the literature deals with such regular measures as mean flow time, mean lateness, the percentage of jobs tardy, and mean tardiness. The mean tardiness criterion, in particular, has been a standard way of measuring conformance to due dates, although it ignores the consequences of jobs completing early. However, this emphasis has changed with the current interest in Just-In-Time (JIT) production, which espouses the notion that earliness, as well as tardiness, should be discouraged. In a JIT scheduling environment, jobs that complete early must be held in finished goods inventory until their due date, while jobs that complete after their due dates may cause a customer to shut down operations. Therefore, an ideal schedule is one in which all jobs finish exactly on their assigned due dates. This can be translated to a scheduling objective in several ways. The most obvious objective is to minimize the deviation of job completion times around these due dates.

2.2.2. Genetic Algorithm. Genetic Algorithm (GA) was developed by John Holland in 1975. GA mimics the natural selection and natural genetic selection process concurrent with artificial system for optimising problems. This algorithm is applied from Darwin’s principle of “Survival of fittest” [2]. Goldberg [1] stated that GA is different from other search methods, which are the superior pros of GA towards other methods. GA operates on a population of solutions rather than a single solution. GA works by encoding the parameter set of the problem into finite-length string. The strings, which represent the solution of the search problems are denoted as chromosomes. The alphabet is denoted as genes. The value of genes is referred as alleles. Implementing natural selection to scheduling problem, chromosomes represent the sequence of scheduling whilst genes are the position of the jobs in the schedule.
2.3. **Simple Genetic Algorithm**

The concept of GA starts by transforming potential solution of the problem into individual chromosomes. An initial set of chromosomes is called a population. Each individual chromosome has their own characteristics or fitness, which appraise through the GA operators: selection, reproduction (crossover), and mutation.

![Simple Genetic Algorithm cycle](image)

The procedure starts from generating an initial population. Then, individuals are evaluated. If the individual is unsatisfied, the GA operation is preceded by starting from selection, recombination, and mutation respectively. The selection chooses a parent chromosome from the population. Secondly, the crossover produces new parent by recombining parts from parents, which is called offspring. Lastly, the mutation produces new parent by making changes within the offspring. The new population is chosen from better fitness, which is kept from each GA operator by operating through continuous iterations; called generations. Once having several generations, the algorithm converges to the best parent, which represents an optimal or suboptimal solution to the problem.

### 3. Research Theories

#### 3.1. Problem Parameter Setting

The cross-docking scheduling problem described in the previous section can be modelled as a machine-scheduling problem. Each incoming container and each outgoing container is a job to be processed by teams that we think of as machines, where only a limited number are available. Further, machines handling incoming cargo can be thought to be parallel and likewise for machines handling outgoing cargo. This is because teams are able to operate simultaneously. Each incoming container can be thought of as a job that has a release time after which it can be processed, a due date and a processing time, assumed to be known beforehand. Each outgoing container has a due date, a processing time, and a number of source containers that feed it. Here, use ‘container’ as a generic packing form to include containers, pallets, and other packing used in DC. Use the terms ‘team’ and ‘machine’ and ‘container’
and ‘job’ interchangeably. The number of breakdown and build-up processing teams is also known together with penalties for earliness and lateness. An experiment is developed to study the suitability of the Genetic Algorithm (GA) solution for the cross-docking scheduling problem. The instance parameter to apply in this problem is shown in Table 1.

| Table 1. Setting up the optimization parameters |
|-----------------------------------------------|
| Parameter          | Value                      |
| Number Generation  | 10,000 and 50,000          |
| Experimental Population | 500 and 1000              |
| Crossover rate     | 0.4, 0.6 and 0.9           |
| Mutation rate      | 0.05, 0.10 and 0.25        |

### 3.2. Setting Parameters

The trials number, population rate, crossover rate, and mutation rate are predefined automatically in the program. However, this value can be adjusted depending on the experiments. In this research, the trials number is 10,000 and 50,000. The population sizes are 500 and 1000. The crossover rates are 0.4, 0.6 and 0.9. The mutation rates are 0.05, 0.1, and 0.25.

Figure 3. Model definition and setting parameters (Adjustable cell ranges)

### 4. Experimental Results

We have provided two methods for solving the cross-dock scheduling problem by using Heuristic and GA method. Both of these are appropriate for the problem. As cross-dock scheduling problem is NP-hard, so it will take a long time for an integer programming solver such as MPL (Not applied in this case) to reach an optimal solution. A part from heuristic methods, this problem has been solved by using Genetic Algorithm (GA). There are various parameters that should be considered in order to evaluate the performance of GA. The experiment is divided into two cases and three performance measurements are considered including of makespan, tardiness time, and machine utilization, which are described below.

**Case I:** One-Phase Single-Machine at inbound area

**Case II:** Two-Phase Parallel-Machine at inbound and outbound area

**Parameter Notation Definition:**

| Parameter | Description |
|-----------|-------------|
| A         | Release time for incoming container i |
| B         | Processing time required to break down incoming container i |
| C         | Due date for incoming container i |
| D         | Number of source of outgoing container j |
| E         | Processing time required to build up outgoing container j |
| F         | Due date for outgoing container j |

---

4
Table 2. Instance Data Set Case II-IV

| Case | Incoming Area | Outgoing Area |
|------|---------------|---------------|
|      | No. Container | No. Forklift  | No. Container | No. Forklift |
| 1    | 15            | 5             | -            | -            |
| 2    | 15            | 5             | 14           | 1            |

Table 3. Data Set Case I-IV

| Number of Containers |
|----------------------|
| 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| A  | 0  | 0  | 0  | 0  | 10 | 10 | 20 | 30 | 10 | 10 | 10 | 10 | 15 | 15 |
| B  | 15 | 15 | 15 | 15 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 15 | 15 |
| C  | 70 | 70 | 70 | 70 | 80 | 90 | 70 | 50 | 60 | 90 | 90 | 90 | 90 | 70 |
| D  | 2  | 4  | 3  | 1  | 5  | 6  | 8  | 3  | 9  | 12 | 13 | 12 | 6  | 12 |
| E  | 20 | 20 | 20 | 20 | 20 | 20 | 10 | 10 | 15 | 25 | 10 | 10 | 10 | 10 |
| F  | 80 | 80 | 80 | 90 | 90 | 80 | 110| 110| 70 | 80 | 110| 110| 110| 110|

Case I: One-Phase Single-Machine Model at Inbound Area (Based Case)

In this case, we will investigate the one phase and single machine problem, which means only one machine at inbound area and only consider the breakdown operations and omit the build-up operations in outbound area, see in Figure 4.

![Figure 4. The experimental result of one-phase single-machine model at inbound area](image1)

Case II: Two-Phase Parallel-Machine Model at Inbound and Outbound Area

![Figure 5. The experimental result of two-phase parallel-machine at inbound and outbound area](image2)
Case II: Two-Phase Parallel-Machine at Inbound and Outbound Area

This case is exactly the cross-docking scheduling model, which is two-phase parallel-machine scheduling problem. Two machines work at inbound area to consider the breakdown operations and the build-up operations in outbound area, see in Figure 5.

| Inbound Area | Complete time |
|--------------|--------------|
| Forklift     | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 1            | 40 | 55 | 70 | 90 | 110 | 150 | 195 | - | - |
| 2            | 20 | 30 | 50 | 65 | 80 | 110 | 145 | 195 | - |

| Outbound Area | Complete time |
|---------------|--------------|
| Forklift      | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 1             | 50 | 60 | 70 | 90 | 100 | 155 | 165 | - | - |
| 2             | 40 | 60 | 75 | 85 | 100 | 155 | 165 | - | - |

Figure 6. The experimental result of two-phase parallel-machine at inbound and outbound area

| Table 4. Comparison of the performance criteria |
|-----------------------------------------------|
| Method            | Performance Criteria (Case 2) |
|                  | Makespan (min) | Total Tardiness (min) | Utilization (%) |
|                  | Inbound  | Outbound | Inbound  | Outbound | Inbound | Outbound |
| GA                | 182     | 155      | 100      | 175      | 100     | 42.4     |
| Heuristics        | 195     | 165      | 115      | 180      | 90      | 36.1     |
| Improved (%)      | 6.67    | 6.06     | 13.04    | 2.78     | 11.11   | 17.45    |

5. Conclusion

The experiment is finding the optimal solution for solving the forklift scheduling problem in cross-docking environment by using GA and heuristic method. There are five cases applied in this case study where the GA algorithm has shown that it is suitable method and given high performance better than heuristic method for solving the cross-dock scheduling problem. Moreover, it can be increased the machine utilization. This leads to improve the forklift scheduling and resource utilization.

References

[1] Goldberg, D.E, Genetic algorithms in search, optimization, and machine learning. Boston: Addison-Wesley Pub. Co., 2004.
[2] Holland, John H. Adaptation in Natural and Artificial Systems. Ann Arbor: University of Michigan Press, 1975.
[3] Jacobs, Larry W., and Lauer, Joachim . DSS for job shop machine scheduling. Industrial Management & Data Systems, 1994.
[4] Melnyk, Steven A., Vickery, Shawnee K., and Carter, Phillip L. Scheduling, sequencing, and dispatching: Alternative perspectives. Production and Inventory Management, 1986.
[5] Roberta S. Russell and Bernard W. Taylor- 4th ed., international ed. Upper Saddle River, N.J.: Prentice Hall, 2003.
[6] Y Li, A Lim and B Rodrigues, Crossdocking – JIT scheduling with time windows; Journal of the Operational Research Society, December, Volume 55, Number 12, 2004