

Multi-objective Genetic Algorithms for Scheduling Material Handling Equipment at Automated Air Cargo Terminals

Henry, Y.K.Lau

Dept. of Industrial and Manufacturing Systems Engineering
The University of Hong Kong
Hong Kong, PRC
Hyklau@hkucc.hku.hk

Ying Zhao

Dept. of Industrial and Manufacturing Systems Engineering
The University of Hong Kong
Hong Kong, PRC
Zhao-ying@hkusua.hku.hk

Abstract— In order to improve the productivities of a typical cargo handling system, it is important to reduce the waiting time of stacker cranes (SCs) and the total traveling time of automated guided vehicles (AGVs) through efficient scheduling of SCs and AGVs, which are cooperating tightly to perform cargo handling operations in an optimal way. In this paper, we develop and investigate the application of the multi-objective genetic algorithm (MOGA) to solve such scheduling problem with the objectives of minimizing the AGV total traveling time and the total delay time of the SC. The results of the experiments demonstrated that MOGA produces better solutions than the single objective genetic algorithms.

I. INTRODUCTION

For air cargo handling in a highly automated air cargo terminal, the ULD (Unit Load Device) is the main loading unit commonly deployed. In the process of air cargo handling, two main processes that involve the movement of ULD take place. First, the incoming ULD's are broken down and in the breakdown process ULDs are ungrouped into individual cargo. Then, outgoing ULDs are built up and in the buildup process cargos are grouped into ULDs. However not all cargos and ULDs pass through these processes. In the breaking down and building-up processes, Automated Guided Vehicles (AGVs) are deployed to transport the ULDs from the conveyor system to the break down workstations or from the build-up workstation to the conveyor system. For the cargos that are built up on a ULD, the conveyor system then transports these ULDs to specific container storage system compartments with Stacker Cranes (SCs) for temporarily storage. For those ULDs that need to be broken down, the SCs will transport them from the specific container storage system compartments to the conveyor system, where the AGVs will transport them to the workstations for breaking down. In these cargo handling processes, in order to achieve high efficiency in cargo handling, it is essential to reduce the waiting time of SCs and the total traveling time of AGVs through the efficient scheduling of SCs and AGVs, which are cooperating tightly to perform the cargo handling operations in an optimal way.

Most existing literature in the field of automated cargo handling only considered single objective or constraint. Meersmans and Wagelmans [1] considered the problem of

integrated scheduling of various types of handling equipment at an automated container terminal with the objective of minimizing the makespan of the schedule. Kim and Kim [2] discussed how to route straddle carriers during the loading operation of export containers in port container terminals to minimize the total travel distance of straddle carriers in the yard. Kim and Kim [3] discussed how to optimally route transfer cranes in a container yard during loading operations of export containers at port terminals to minimize the total container handling time of a transfer crane. Though, some researchers study the multi-objective problems, they combine the multiple objectives into one single objective. For example, Kim and Park [4] combine the objectives of minimizing the makespan and the total completion time linearly with fixed weights to schedule the cranes. Kim and Bae [5] set different priorities to the objective in minimizing the delay time of cranes when they dealt with the bi-objective AGV dispatching problems. These are the ways to solve multi-objective problems, and allow a simpler optimization algorithm to be used, but the obtained solution largely depends on the weight vector used in the transformation process. Moreover, a decision-maker may be interested in knowing alternate solutions.

Genetic algorithms have been mainly applied to single-objective optimization problems. Some studies have attempted to design multi-objective genetic algorithms since Schaffer's work. Schaffer [6] proposed the Vector Evaluated Genetic Algorithm (VEGA) for finding Pareto optimal solutions of multi-objective optimization problems. The point of multi-objective optimization problems is how to find all possible tradeoffs among multiple objective functions that are usually conflicting. Since it is difficult to choose a single solution for a multi-objective optimization problem without iterative interaction with the decision maker, one general approach is to show the set of Pareto optimal solutions to the decision maker. Then the decision maker can choose any one of the Pareto optimal solutions. To find out all the Pareto optimal solutions using genetic algorithms, the variety of individuals should be kept in each generation. Recently, Horn *et al.* [7] proposed the Niche Pareto Genetic Algorithm by incorporating the concept of Pareto domination in the selection procedure and applying a niching pressure to spread the population out along

Pareto fronts. Murata *et al.* [8] proposed the multi-objective genetic algorithm (MOGA) to search the Pareto optimal solutions of bi-objective and tri-objective scheduling problems.

In this paper, the MOGA is adapted and applied to key equipment, particularly, SCs and AGVs scheduling problems with two objectives: to minimize the total delay time of SCs, and to minimize the AGV total traveling time in a typical air cargo handling system. Experiment results show that the MOGA approach outperforms the single objective genetic algorithms.

II. SCHEDULING PROBLEMS OF EQUIPMENT

In this paper, all the operation times for handling equipment to perform tasks are assumed to be deterministic, which include the operation time of SC, the travel time of an AGV, and the transfer time between equipment. The task sequence lists, which are constructed for load balance of each SC, are made to specify the sequence of discharging and loading operations of individual ULDs, that is, this paper only addresses the problem of solving a static scheduling problem where AGVs are assigned to cover a known set of tasks.

We define an event, E_{ik} , to be the beginning of load or unload of i^{th} ULD from or to the I/O conveyor by the SC k , $k \in K, i \in m_k$, where K is the set of SCs involved in the ULD operation, and m_k is the number of tasks for the ULD operation assigned to SC k . Each event is relevant to one SC task. As the pertinent information on cargo operations is assumed to be known at the beginning of planning horizon, the earliest possible time of event E_{ik} (P_{ik}) can be calculated under the condition that there is no delay of SC's operations. The time of event E_{ik} is denoted by R_{ik} that is increased when the SC's operation corresponding to event E_{ik} is delayed and that can be replaced with P_{ik} when delay of the operation by a SC is not allowed. One of the objectives is the minimization of all delays of SC's operations that according to the following:

$$\text{Minimize } f_1 = \sum_{k \in K, i \in m_k} (R_{ik} - P_{ik})^+ \quad (1)$$

The event that corresponds to the beginning of loads or unloads of a ULD from or to the I/O conveyors by the AGV k is denoted as e_{ik} , particular, e_{0k} represents the initial event of AGV k . The pure travel time from the location where the event e_{ik} occurs to the location where the event e_{jk} occurs is denoted by t_{ijk} . The location where the event e_{0k} occurs is the initial position of AGV k . The total AGV traveling time is calculated by:

$$\text{Minimize } f_2 = \sum_{k \in V} \sum_{i, j \in T_k} t_{ijk} \quad (2)$$

with V being the set of AGVs and T_k being the set of e_{ik} that the AGV would go through, $k \in V$.

Due to the tight cooperation between equipment, a mapping function to assign the SC loading and discharging events to AGV loading and unloading events is developed:

$$\Gamma : E_{jg} \rightarrow e_{ik} \quad (3)$$

with E_{jg} being the j^{th} event in the sequence for SC g and e_{ik} being the i^{th} event in the sequence for a AGV k .

III. MULTI-OBJECTIVE GENETIC ALGORITHM

Combing multiple objectives linearly with fixed weights is the common method to simultaneously consider these objectives, which transforms multiple objectives into single objective. However, this approach overlooks and loses the diversity of objectives. To overcome this shortcoming, Pareto optimal solutions are applied to show the diversity among these multiple objectives. The definition of Pareto optimal solutions can be referred to Tamaki *et al.* [9]. The MOGA, which aims to find the Pareto optimal solutions, is developed to solve the scheduling problem described in the section 2.

A. Representation and Initialization

There are two essential issues to be dealt with in the AGV task scheduling problems, which are:

- Task assignment to AGVs
- Sequence tasks for each AGV

A representation method presented by Cheng *et al.* [10] was adopted to encode these two things into a chromosome. The representation consists of a *task symbol* list and an *assignment symbol* list, in which integers are used to represent all possible permutation of tasks and asterisks * are used to designate the assignment of tasks to AGVs. Let us consider a simple example with 8 tasks and 4 AGVs. Suppose there is a schedule that the task sequences of AGV1 is (1, 6), AGV2 is (5, 2), AGV3 is (3, 8) and AGV4 is (7, 4). The chromosome can be represented as follows:

$$[16 * 52 * 38 * 74]$$

Generally, for an n task in AGV problem, a legal chromosome contains n task symbols and $m-1$ assignment symbols, resulting in a total size of $(n+m-1)$. As the initial population is randomly generated, not all the individuals are flexible. To make all the individuals flexible, one repair function is developed to make each chromosome contain all integers from one to the number of tasks exactly once. Due to the assignment of SC tasks to different AGVs, and each task relevant to one event, a mapping of the event E_{jg} of SCs' to the corresponding event e_{ik} of AGVs' is done using the function in (3) ($j \rightarrow i, g \rightarrow k$). Therefore, the task sequences of SCs' should be taken into account when the flexible individual chromosome was formed. Another repair function is developed to make the task sequence for each AGV correspond to the task sequences of SCs'.

B. Genetic Operators

There are four genetic operators commonly used in the genetic algorithms: selection, crossover, mutation and

evaluation. Usually, the crossover and mutation are used as main genetic operators and the performance of a genetic system depends, to a great extent, on the performance of the crossover and mutation operators used. We adopt a different approach for designing the crossover and mutation operation. As mentioned, the essential issue of the AGV scheduling problem is the combination and permutation of the tasks and AGVs. Both crossover and mutation operators are designed to handle the task partition and task permutation. For the MOGA developed in this paper, there is a special characteristic feature of selection operation, where the Murata's idea is adopted.

Selection: In the selection procedure, a weighted sum of multiple objective functions is used to combine them into a scalar fitness function. The weights attached to the multiple objective functions are not constant but randomly specified for each selection. Therefore, the direction of search is not constant. The variable weights can be specified as follows for the equipment scheduling problem considered in this paper.

$$f = w_1 f_1 + w_2 f_2, \quad (4)$$

with

$$w_1 = (i-1)/(N_{\text{selection}}-1) \quad i=1,2,\dots,N_{\text{selection}}, \quad (5)$$

$$w_2 = 1 - w_1, \quad (6)$$

where $N_{\text{selection}}$ is the number of selections in each generation. The weighting scheme in (4)-(6) means that different weights are used for selecting each pair of parents. Because $N_{\text{selection}}$ pairs of parents are to be selected in each generation, $N_{\text{selection}}$ pairs of different weights are specified in (5)-(6), in which the values of w_1 and w_2 are evenly distributed over the closed interval [0, 1].

The weighted sum f in (4) is used for determining the selection probability of each individual. Because the weights w_i 's are not constant but variable; the selection probability of each individual is also variable even in a single generation, which realizes various search direction in the MOGA.

Crossover: The sub-schedule means a complete schedule for one AGV, we intend to use such crossover to maintain the building blocks in the offspring in much the same manner as Cheng *et al.* [10] described. The crossover takes two parents and creates a single offspring by propagating the overall partitioning structure and a sub-schedule into offspring from one parent and then completing the offspring with remaining tasks derived from another parent. It performs with main three steps:

- 1) Obtains asterisk positions from one parent.
- 2) Obtains a randomly selected sub-schedule from the same parent.
- 3) Obtains remaining tasks from the other parent making a left-to-right scan.

With the two parents, and one of sub-schedule of parent 1, which is selected to be propagated into offspring, the operation of crossover is illustrated by Fig. 1. According to Fig. 1 we can see that the proposed crossover can adjust task partition and task order simultaneously.

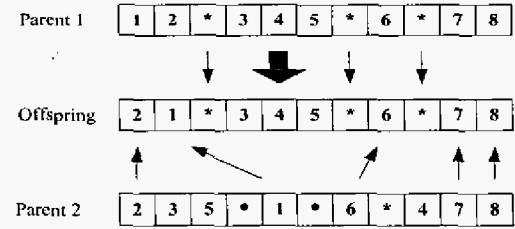


Fig. 1 Illustration of crossover operation

Mutation: This paper adopts the swapping mutation operation, which selects two random positions and then swaps their genes. The randomly swapped genes may be either task or asterisk, of which possible combinations result in four basic types of mutation.

- 1) If both genes are tasks, two cases may occur. One case is that two selected tasks are assigned to the same AGV. In such case the mutation alters the task order for the AGV. Another case is that two tasks are assigned to different AGV. In such case the mutation alters both task order and task assignment to AGVs for the chromosome.
- 2) If one gene is asterisk and another is task, the mutation alters task order and task assignment to AGVs for the chromosome.
- 3) If both genes are asterisk, the mutation performs a trivial operation. This kind of mutation is prohibited.

The second one is the only genetic operation which can alter the position of asterisks shown in Fig. 2. Without this type of mutation, the position of asterisks will never change throughout all evolutionary process and therefore genetic search is confined by the initial positions of asterisks. So this type of mutation plays a vital role in the genetic search.

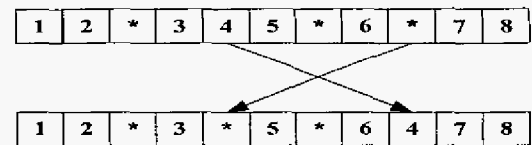


Fig. 2 Illustration of the fourth type of mutation operation

Evaluation and Elite strategy: The original concept of fitness is the larger the better because solutions with larger fitness tend to propagate to the next generation. In this paper, minimization of objective is considered. Hence it contradicts the original idea of fitness. Many authors have done the transformation to reverse the minimization to maximization. However, in this paper no transformations are performed. Therefore, the fitness is the smaller the better. Moreover, for the infeasible individual solutions, the penalty function is developed to make their fitness to be 0, and therefore they will not be propagated to the next generation.

During the process of evolution, a tentative set of Pareto optimal solutions is stored and updated at every generation,

and each of them is used as elite individual to randomly replace the individual of the old population. As the number of Pareto optimal solutions found in each generation is varied, this elite preserve strategy has an effect on keeping the variety in each generation in our MOGA.

Algorithm: The steps of our MOGA are described in the following:

- 1) **Initialization:** Generate an initial population containing N_{pop} individuals where N_{pop} is the number of individuals in each population
- 2) **Evaluation:** Calculate the values of the objective functions for the generated individuals. Update a tentative set of Pareto optimal solutions.
- 3) **Elitist strategy:** Add all the Pareto optimal solutions to the current population, and randomly remove the same number of individuals from the N_{pop} individuals generated by the above operations.
- 4) **Selection:** Calculate the fitness value f of each individual by using the weights defined as (4)-(6). Select a pair of individuals from the current population using the roulette wheel selection strategy.
- 5) **Crossover:** For each selected pair, apply the crossover operation to generate an offspring with the crossover probability p_c . As a single offspring is generated from a pair of parents in the crossover operation, the $N_{selection} = N_{pop}$.
- 6) **Mutation:** For each individual generated by the crossover operation, apply the swapping mutation with a pre-specified mutation probability p_m .
- 7) **Termination:** If the number of population does not reach the pre-specified maximum generation size, return to Step 2.
- 8) **User selection:** The MOGA produces the final set of Pareto optimal solutions for the decision maker. A single solution is selected by the decision maker's preference.

IV. SIMULATION RESULTS

Fig. 3 illustrates the layout of a cargo handling system, which was used in the numerical experiment in this study. It is assumed that each aisle has an input and an output buffer (conveyor) with three storage locations. It is also assumed that the adjoining conveyors have the same load and unload points (L/U points), and the L/U points are located in front of the workstations, based on the layout design of the guide path and the dynamics of the AGVs, the traveling time between every combination of the L/U points is evaluated and is given in Table 1.

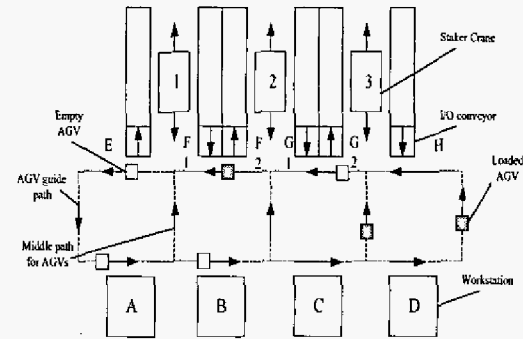


Fig.3 An abstract model of a zone of operation in an automated cargo handling system

TABLE I. TRAVEL TIME BETWEEN L/U POINTS IN THE CARGO HANDLING SYSTEM (IN SEC)

From To	A	B	C	D	E	F	G	H
A	-	15	30	45	65	80	95	110
B	50	-	15	30	80	65	80	95
C	65	50	-	15	95	80	95	80
D	80	65	50	-	110	95	80	65
E	35	50	65	80	-	50	65	80
F	50	65	80	95	15	-	50	65
G	65	80	95	110	30	15	-	50
H	80	95	110	125	45	30	15	-

TABLE II. THE SCS' TASK SEQUENCES

Task Sequence	ID of SC	Task Type*	ID of Workstation	Earliest Possible Time	AGV Event Time
1	1	L	B	0	-15
2	1	D	D	35	50
3	1	L	B	40	25
4	1	D	D	80	95
5	2	L	C	0	-15
6	2	D	B	55	70
7	2	L	C	60	45
8	2	D	C	85	100
9	3	D	B	25	40
10	3	L	B	30	15
11	3	D	D	75	90
12	3	L	C	80	65

* L: loading, D: discharging

The task sequence defines the sequence of discharging and loading operations of individual ULDs. Table 2 shows a sequence list that defines the operation of the three SCS' to perform cargo handling operations based on the dual-cycle strategy.

Because the MOGA adopts probabilistic search methods, its performance cannot be evaluated with a single trial. Thus, we applied the MOGA to each test problem 10 times. In each test problem, only the non-dominated solutions are compared.

Based on the above primary computational data, the performance of the MOGA is studied by varying the parameters of the probability of crossover and mutation operations. A fixed generation size of 500 and a population size N_{pop} of 30 are used. A set of randomly generated combination of probability of crossover and mutation is given in below:

(0.2, 0.8), (0.3, 0.7), (0.4, 0.6), (0.1, 0.9), (0.6, 0.4) and (1.0, 1.0)

The Pareto optimal solutions found with these combinations are shown in Fig. 4, where the horizontal and vertical axes are the AGV traveling time and the total delay time of SC. As the objectives of the scheduling problem are to minimize the AGV traveling time and to minimize the total delay time of SC, the solutions with minimum two objectives may be preferred by the decision maker. From Fig. 4 we can see that the combination of (0.3, 0.7) produces the relative better results, as most of its Pareto optimal solutions are non-dominated ones.

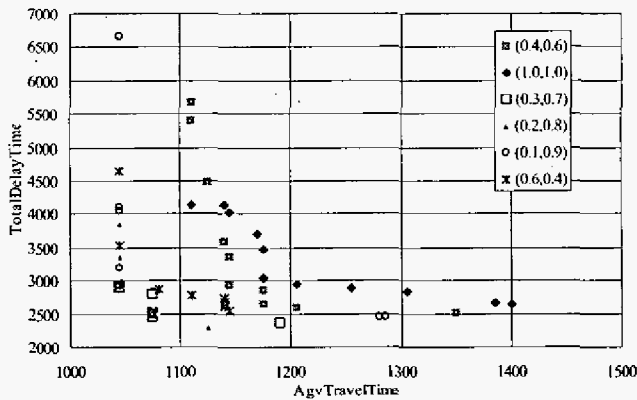


Fig. 4 Results with different combination of probability of crossover and mutation

Following this study, the influence of the parameter of population size to the performance of the MOGA is then investigated. A fixed generation size of 500 with p_m and p_c set to 0.3, 0.7 respectively are used. By varying the population size from 10 to 60, simulation studies are undertaken. Fig. 5 shows the non-dominated results with 10 random runs for each parameter setting. From the results we can see that when the solution size is between 20 and 40, the best optimal solution is obtained. When the solution size is larger than 40, the performance of the algorithm has not been improved significantly.

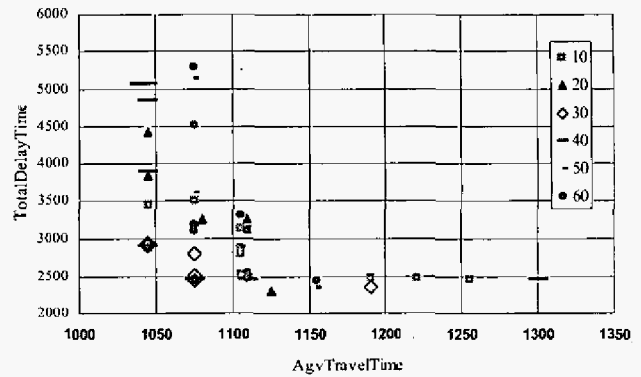


Fig. 5 Results with different population size

The proposed MOGA is further compared with the single objective genetic algorithm. The basic settings of parameters of the genetic algorithm are the same: with a generation size of 500 and the population size of 30. The values of p_m and p_c in MOGA are set to 0.3, 0.7 respectively, while in the single objective GA, they vary from 0.0 to 1.0. Simulation studies of 10 runs for each value are performed. A set of Pareto optimal solutions are obtained after 110 simulation runs, which are performed in the same manner as for the MOGA, with the results indicated by the symbol ■ in Fig. 6. From Fig. 6, we can see that the set of non-dominated solutions obtained by MOGA is superior to the set of non-dominated solutions obtained using the single objective genetic algorithm. This is because many solutions denoted by ■ are dominated by solutions denoted by ●. This demonstrates the superior performance of the proposed MOGA.

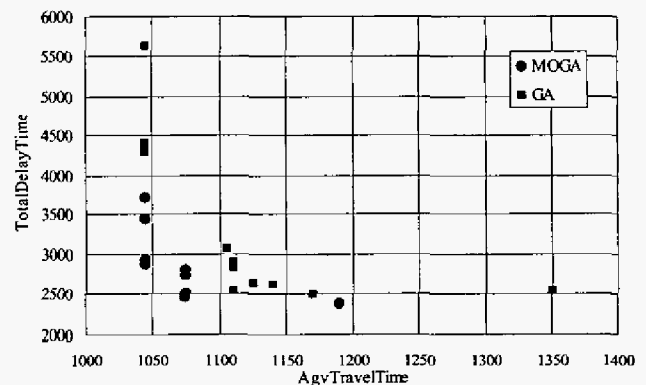


Fig. 6 Comparison on MOGA and single objective GA

V. CONCLUSIONS

In this paper, we addressed the key equipment scheduling problem in a typical air cargo handling system with the multiple objectives of minimizing the AGV traveling distance and minimizing the total delay time of SC. The paper describes in detail the design and the implementation of the multi-objective genetic algorithm. Simulation studies are

performed on a typical configuration of a typical air cargo handling terminal and experimental results that demonstrated the performance of the proposed MOGA are shown and analyzed. It is shown that MOGA produces better optimal solutions than the single objective genetic algorithm. The proposed MOGA can be easily applied to other material handling equipment scheduling problems with multi-objective.

REFERENCES

- [1] P.J.M. Meersmans, and Wagelmans, A.P.M. Wagelmans, "Effective algorithms for integrated scheduling of handling equipment at automated container terminals," *Econometric Institute Report 226/Erasmus University Rotterdam*, Econometric Institute, 2001.
- [2] K.H. Kim, and K.Y. Kim, "Routing straddle carriers for the loading operation of containers using a beam search algorithm," *Computers and Industrial Engineering*, vol. 36, pp.109-136, 1999a.
- [3] K.H. Kim, and K.Y. Kim, "An optimal routing algorithm for a transfer crane in port container terminals," *Transportation Science*, vol 36, pp.109-136, 1999b.
- [4] K.H. Kim, and Y.-M. Kim, "A crane scheduling method for port container terminals," *European Journal of Operational Research*, vol. 156, n. 3, pp. 752-768, 2004.
- [5] K.H. Kim, and J.W. Bae, "A dispatching method for automated guided vehicles to minimize delays of containership operations," *International Journal of Management Science*, vol. 5, n. 1, pp. 1-25, 1999.
- [6] J.D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithm," *Proc. 1st ICGA*, pp.93-100, 1985.
- [7] J. Horn, N. Nafpliotis, and D.E. Goldberg, "A niched Pareto genetic algorithm for multiobjective optimization," *Proc. 1st CIEC*, pp.82-87, 1994.
- [8] T. Murata, H. Ishibuchi, and H. Tanaka, "Multi-objective genetic algorithm and its applications to flowshop scheduling," *Computers ind. Engng*, vol. 30, n. 4, pp..957-968, 1996.
- [9] H. Tamaki, H. Kita, and H. Kobayashi, "Multi-objective optimization by genetic algorithms: A review," *Proceedings of IEEE International Conference on Evolutionary Computation*, pp.517-522, 1996.
- [10] R. Cheng, and M. Gen, "Parallel machine scheduling problems using memetic algorithms," *Computers ind. Engng*, vol. 33, n. 3-4, pp.761-764, 1997.