

Research Article

Multiobjective Simulation-Based Optimization Based on Artificial Immune Systems for a Distribution Center

Chris S. K. Leung  and Henry Y. K. Lau

Department of Industrial & Manufacturing Systems Engineering, The University of Hong Kong, Pokfulam, Hong Kong

Correspondence should be addressed to Chris S. K. Leung; lskchris@yahoo.com

Received 10 October 2017; Revised 27 March 2018; Accepted 19 April 2018; Published 21 May 2018

Academic Editor: Efren Mezura-Montes

Copyright © 2018 Chris S. K. Leung and Henry Y. K. Lau. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Competitive market factors, such as more stringent government regulations, larger number of competitors, and shorter product life cycle, in recent years have created more significant pressure on the management in all supply chain parties. To this end, the ability of analyzing and evaluating systems and related operations involving the deployment of complex multiobjective material handling systems is vital for distribution practitioners. In this respect, simulation modeling techniques together with optimization have emerged as a very useful tool to facilitate the effective analysis of these complex operations and systems. In this paper, we apply a multiobjective simulation-based optimization framework consisting of a hybrid immune-inspired algorithm named Suppression-controlled Multiobjective Immune Algorithm (SCMIA) and a simulation model for solving a real-life multiobjective optimization problem. The results show that the framework is able to solve large scale problems with a large number of parameters, operators, and equipment involved.

1. Introduction

Simulation modeling is indeed a powerful industrial engineering technique for studying the functioning, performance, and operation of complex systems. As such, it becomes an extremely useful tool for stakeholders and decision-makers in various industries and domains including multiobjective optimization. By changing input data and operating parameters of a system being studied with simulation, predictions about the system's behaviors can be obtained through computer-aided simulation for helping management make the right decisions. Unlike a mathematical model, simulation can handle a variety of complex factors that are commonly found in real world. In addition, simulation is a cost-effective means for existing process redesign and new system design because alternative solutions can be studied and evaluated for correctness and feasibility before actual implementation. More importantly, the accuracy of the performance measures of the complex systems obtained from simulation models is normally higher than that of analytical methods because analytical methods in general involve making unrealistic

assumptions for the systems or problems under investigation [1].

In real world, many problems no matter whether they are in the domain of engineering, finance, business, or science can be formulated into different forms of optimization problems. These problems are characterized by the requirement of finding the best possible solution that fulfills certain criteria under certain constraints. Most of the real-world optimization problems normally involve multiple objectives rather than one single objective, in which some objectives conflict with others. Solving this kind of problems is never an easy task because objectives of such problems are often found to be noncommensurable and conflicting. Very often, there is no single best solution to the multiobjectives optimization problems, but rather a set of optimal solutions that exists among the objectives. However, using simulation modeling alone cannot provide us with optimal solutions to these optimization problems. Therefore, an optimization algorithm is needed to guide the search process to the optimal solutions. Over the past decades, different algorithms have been developed for solving multiobjective optimization

problems. However, some algorithms, such as Simulated Annealing (SA) [2] and Tabu Search (TS) [3–5], do not easily solve multiobjective optimization problems since they are initially developed for solving single objective optimization problems without having the ability to generate a set of optimal trade-off solutions. One of the possible approaches for them to solve the multiobjective optimization problems is to transform the multiple-objective problems into the single objective problems by emphasizing one particular optimal solution at each run [6]. Some other algorithms, such as Genetic Algorithm (GA) [7], Evolutionary Strategy (ES) [8], and Artificial Immune Systems (AIS) [9] which are inspired by natural processes, have been proved to be the effective algorithms in both single objective and multiobjective optimization contexts. Among these algorithms, AIS based on the concepts of biological immune system have received special attention among the research community because the biological immune system provides a rich source of stimulation and inspiration to the research community with their interesting characteristics: distributed nature, self-organization, memory, and learning capabilities. Motivated by its great potential for solving multiple-objective optimization problems, the study reported in this paper is to demonstrate how a real-life multiobjective simulation-based optimization problem in logistics industry where material handling system (MHS) is involved can be solved by a simulation-based optimization framework comprising a simulation tool together with an AIS-based algorithm.

This paper proceeds as follows: Section 2 starts by giving a brief overview of material handling and multiobjective optimization. Section 3 introduces the multiobjective optimization framework for simulation-based optimization. Section 4 evaluates the performance of the framework through a real-life case study on a MHS by benchmarking with some well-known optimization algorithms. Finally, concluding remarks are presented in Section 5.

2. Background

2.1. Importance of Material Handling in Distribution Industry. Material handling, which is the total management of material concerns in an operation, is a vital element of industrial processes. Material handling involves a variety of operations including the movement, storage, protection, and control of materials, products, and wastes throughout the processes of manufacturing, distribution, and disposal. Having efficient MHS is of great importance to various industries to maintain and facilitate a continuous flow of material through the workplace and guarantees that required materials are available when needed. It is especially important to logistics and manufacturing industries as it accounts for a large percentage of the operation in these industries. In the manufacturing sector, the time spent on different kinds of product handling and transportation can be as much as the time used on the value-added processes. Banks et al. [10] claimed that the time of material handling accounts for about 85% of the total manufacturing time. In addition to time, the money used on material handling activities is equally high because the material handling equipment and systems require

large capital investment in terms of system design, installation, operations, maintenance, and so forth. A number of studies conducted in different industries show that the cost of material handling alone is about 20–25% of the total manufacturing cost [11].

There are different kinds of material handling equipment and systems available that range from simple hand truck and pallet rack to complex conveyor system and Automated Storage and Retrieval System (AS/RS). A typical MHS is composed of different smaller components closely working together, thus making business activities more efficient and cost-effective. Over the past decade, MHS and its function have undergone a big change because of new advances in technologies, such as the development and applications of automation techniques and robotics, by which a large number of manual handling jobs are replaced by machines. Since the entire production or distribution process is automated, MHS has to respond just-in-time to the requirements of different processing activities. These new technologies, today, increasingly become prevalent in different industries as they help ease drudgery for manual labors and some of the mechanized or automated handling jobs are physically impossible to be done by workers.

Norman [12] claimed that equipment capacity, speed, and arrangement are the most critical considerations when modeling and optimizing MHS. Capacity under this context is the maximum quantity of products handled by the equipment. Speed is the average operating speed of the equipment, which may include acceleration, deceleration, and speeds of various equipment components. Configuration is the layout and structure of the MHS or its moving paths.

2.2. Optimization of MHS via Simulation. In the literature, there are a number of studies that dedicatedly contribute to the optimization of MHS via simulation. For example, Ebbesen et al. [13] studied the baggage handling system at airports. They developed an approach to optimize the design of conveyor systems, that is, the design of tracks suited for baggage handling systems with the use of a time domain simulation model of the entire system. Sergueyevich et al. [14] proposed a simulation model of the overhead monorail conveyor system coupled with statistical methods for analyzing and solving the multiobjective optimization problem regarding the manufacturing process. The aim was to determine the optimum speed of conveyor, lengths of queues, time in system, capacity of machines, and so forth under certain limitations. Elahi et al. [15] studied the General Motors paint shop conveyor system by developing a simulation model. The model works firmly with a decision optimizer incorporating integer linear programming model and dynamic programming model at critical points such as the beginning and end of buffer conveyors in the system in order to regroup batches of different color cars. Leung and Lau [16] proposed a simulation-based optimization framework that combines the processes of optimization and simulation for solving typical linear optimization problems related to logistics and production operation. The framework integrates an AIS-based algorithm with a simulation tool

for the evaluation of optimal system parameters and to reveal the performance of systems. Subulan and Cakmakci [17] made use of ARENA simulation program and Taguchi experimental design method to build a solution model for effectively designing material handling–transfer systems and optimizing the performance of automation technologies in automobile industry. Chang et al. [18] proposed a framework that integrates simulation optimization and data envelopment analysis techniques to find out the optimal vehicle fleet size for a multiobjective problem in automated materials handling systems. Lin and Huang [19] extended the optimal computing budget allocation by adding Genetic Algorithm together with the help of a simulation model for optimizing the vehicle allocation for the automated material handling system in semiconductor industry.

2.3. Multiobjective Simulation-Based Optimization Approaches. In practice, optimization problems involve several objectives that often conflict with each other and must be simultaneously optimized so that a possibly uncountable set of trade-off solutions rather than a single optimal point is found with respect to the contradicting objectives. Therefore, the aim of these problems is to find out the global trade-off solutions that effectively spread over the Pareto front. No solution from the Pareto front is worse than any other solution because it is better in at least one objective. These problems are normally termed as multiobjective problems, which were first studied in an economic context and then extended to the fields of science and engineering [20]. Since multiobjective optimization problems involve several objectives, the view towards optimum has changed, hence changing the aim from finding a single solution to obtaining a set of compromised solutions. Today, the notion of optimum for a multiobjective optimization problem is frequently called Pareto optimum, which was first proposed by Edgeworth [21] and then generalized by Pareto [22, 23].

As is known, the notion of optimality for multiobjective optimization problems is different from that of single objective optimization problems because the aim of multiobjective optimization problems is to find a set of optimal trade-off solutions rather than a single optimal solution. Thus, in the absence of preference information on the objectives, the concept of Pareto optimality is adopted in this study for solving multiobjective optimization problems [24] and several definitions about Pareto optimality [20, 25, 26] are considered and stated below.

Definition 1 (Pareto optimality). A point $X^* \in \Omega$ in the search space is said to be Pareto optimal/nondominated with respect to Ω if and only if there are no other solutions $X \in \Omega$ for which $\vec{f}(X)$ dominates $\vec{f}(X^*)$. In other words, X^* is Pareto optimal with regard to the whole decision variable space if it cannot be improved in any one objective without resulting in a simultaneous degradation in other objectives.

Definition 2 (Pareto dominance). Consider, without loss of generality, two decision vectors X_a and $X_b \in \mathbf{X}$ of a minimization problem. Then, a vector of decision variables X_a is said to dominate another vector X_b , which is denoted

by $X_a < X_b$, if and only if X_a is partially less than X_b . That is, the following two conditions must be satisfied:

- (1) X_a is not worse than X_b in all objectives

$$f_i(X_a) \leq f_i(X_b), \quad \forall i = 1, 2, \dots, r \quad (1)$$

- (2) X_a is better than X_b in at least one objective

$$f_i(X_a) < f_i(X_b), \quad \exists i = 1, 2, \dots, r, \quad (2)$$

where i is the number of objectives.

However, when any of these conditions are violated, the two solutions X_a and X_b are said to be indifferent to each other instead of dominating the other or being dominated by the other. Based on the above relations, Pareto optimal set, Pareto front, and nondominated set are defined below.

Definition 3 (Pareto optimal set). Pareto optimal set of solutions is a collection of all Pareto optimal solutions, which is defined as

$$\text{PS} := \{\mathbf{X} \in \Omega \mid \neg \exists X^* \in \Omega, X^* < \mathbf{X}\}. \quad (3)$$

Pareto optimal solutions are those solutions in the decision variable space whose corresponding objective vector elements cannot be all simultaneously improved [27]. The solutions inside the Pareto optimal set may have no apparent relationship except their membership in the set. Pareto optimal solutions are classified as such based on their values being evaluated through whatever means.

Definition 4 (Pareto front). A surface or line containing all nondominated solutions is called Pareto front, which is represented by

$$\text{PF} := \{\vec{f}(\mathbf{X}) \mid \mathbf{X} \in \text{PS}\}. \quad (4)$$

According to the literature, finding an analytical expression of the Pareto front is a very difficult task. Therefore, a common approach for Pareto front generation is to find out the points within Ω and their corresponding value $f(X)$, $X \in \Omega$. When this procedure is repeated a sufficient number of times, the nondominated points and hence the Pareto front are most likely to be found in the objective space [20].

Definition 5 (nondominated set). Of a solution set S , a nondominated set of solutions S^* comprises solutions that are not dominated by other solutions in the set S . It is worth mentioning that while Pareto optimal solutions are always nondominated solutions, nondominated solutions may include both non-Pareto optimal solutions and Pareto optimal solutions, thus revealing that the true Pareto optimal solutions could hardly be represented by the nondominated solutions obtained from running an optimizer. Thus, the idea, stated by van Veldhuizen and Lamont [28], about the true Pareto front PF_{true} distinguishing it from the final set of nondominated solutions found by an algorithm is called known Pareto front PF_{known} .

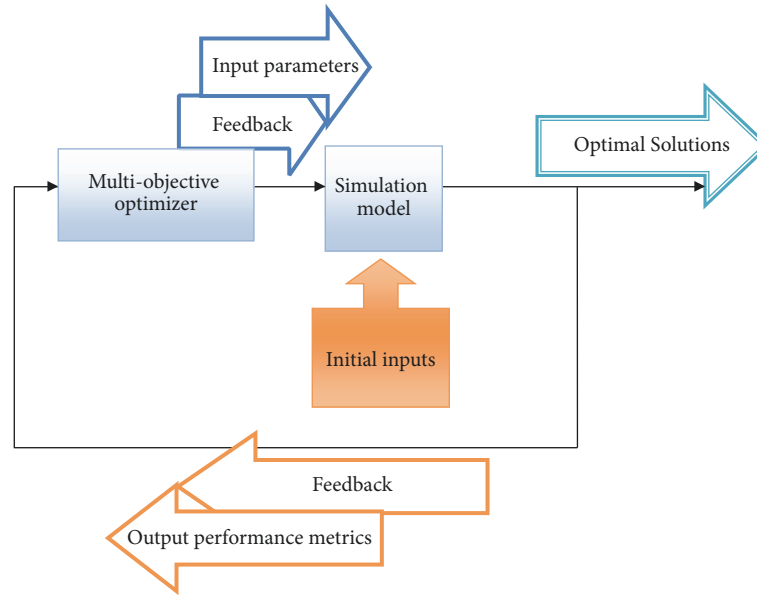


FIGURE 1: The framework for multiobjective simulation-based optimization (source: modified from Leung and Lau's work [16]).

Modern optimization approaches are very often population-based and evolutionary in nature. In such methods, the search for the global optima essentially comprises an iterative process that replaces the candidate solutions in the population by newly generated ones with an aim of achieving continuous improvement in the performance of the best candidate solutions through the help of mechanisms that guide the search to find a set of nondominated solutions. The use of the modern optimization approaches, especially population-based evolutionary algorithms, to solving the complex multiobjective optimization problems has been motivated mainly because of the following critical reasons. First, population-based evolutionary algorithms can recognize the specificity of multiobjective optimization problems by working simultaneously on all objectives and finally generating a group of optimal trade-off solutions, thus forming a uniformly distributed Pareto front. Second, as the name implies, the population-based approaches can deal with a population of candidate solutions simultaneously, allowing the generation of several elements of the Pareto optimal set in a single run of an optimizer instead of performing many separate runs when using classical mathematical programming methods [29]. This allows the decision-makers to simply pick the one that best fits the problem at hand, thus preventing the need to reconfigure and to rerun the optimization tool for finding other alternative Pareto optimal solutions. Third, their capability of maintaining diversity among the candidate solutions in the population is important to prevent the search from premature convergence to a specific region of the solution space, thus allowing a better exploration of the solution space and minimizing the susceptibility of the search to the presence of poor local optima in the optimization problems [30]. The final reason is that the population-based approaches are less susceptible to the continuity or shape of the Pareto front as

they can deal with concave and discontinuous Pareto fronts without difficulty [27, 31]. Nevertheless, on the other hand, the weaknesses of these population-based methods are that they do not guarantee the identification of optimal trade-off solutions and the solutions obtained are likely to be stuck at some “good” approximations [32].

During the past few decades, a large number of publications have been done in population-based evolutionary algorithms and proved to be effective for solving multiobjective optimization problems since the first multiobjective evolutionary algorithm has been developed by Schaffer [33]. These algorithms include PAES [34], NSGA-II [35], PESA [36], SPEA2 [37], PESA-II [38], micro-GA [39], micro-GA2 [40], omni-aiNet [41], NNIA [42], omni-AIOS [43], MTLBO [44], and SCMIA [45].

3. A Multiobjective Simulation-Based Optimization Framework

3.1. Mechanisms of the Multiobjective Simulation-Based Optimization Framework. The optimization framework adopted in this study is developed by taking advantage of the idea of separation between the optimization method and the simulation model. For this reason, the optimization framework can remain the same or require only minor modifications such as changing range of parameters, data type of decision variables, and number of decision variables, to optimize the simulation model that incorporates new requirements. This framework in fact is a modified version of Leung and Lau's work [16], which incorporates a multiobjective optimization algorithm instead of a single objective algorithm (Figure 1). The multiobjective algorithm adopted in the framework is Suppression-controlled Multiobjective Immune Algorithm (SCMIA) proposed by Leung and Lau [45]. Concepts of the algorithm are briefly discussed in the next section.

TABLE 1: Mapping between the biological immune system and SCMIA.

Biological Immune System	SCMIA
Antigen (Ag)	Objective function (simulation model in simulation study) to be optimized
Antibody (Ab)	Candidate solution (a set of decision variables) to be optimized
Ag-Ab affinity	Fitness value of each candidate solution evaluated based on Pareto dominance
Ab-Ab affinity	Crowding-distance proposed by Deb et al. [6] working as a measure of population diversity
Immune suppression	Mechanism to control the number of nearby candidate solutions based on similarity among candidate solutions in both the objective space and decision variable space
Memory cell	Current best nondominated solution

The framework consists of two critical components, namely, (1) the simulation model and (2) the immunity-inspired multiobjective optimization algorithm. A problem or a system to be optimized is represented by a discrete-event simulation model, which is developed separately from the optimization algorithm. The simulation model under study takes inputs (solutions) produced from the optimization algorithm (except the initial inputs which are entered by user). Based on these inputs, the simulation is executed automatically with various parameter settings. In other words, the simulation of the system with the parameters under consideration is iteratively conducted in an automatic manner. And then the simulation model generates several output performance metrics as feedback on the quality of solutions obtained from the optimization algorithm. During the process of simulation-based optimization, it is worth emphasizing that the general guidelines of discrete-event simulation should be followed, such as performing multiple replications of the simulation with the same input parameters but different random number streams to minimize the variability of the simulation outputs.

In essence, the AIS-based multiobjective optimization algorithm is a search algorithm. The simulation results (output performance metrics) direct the algorithm to search for a new set of input parameters that takes the system towards its optimal setting with respect to certain criteria. Hence a feedback mechanism is incorporated to direct the search for optimal solution in a controlled manner. In other words, the generation of a new set of input parameters for the simulation model is based on the simulation results of past evaluations. As such, the input parameters and the output performance metrics shown in the figure in fact serve as the feedback to the simulation model and the optimization algorithm, respectively. This iterative process repeats until prespecified termination criteria are met. The criteria could include the following: for example, no improving solutions can be found or the predefined number of iterations is reached.

In this framework, the optimization algorithm was implemented with Excel VBA, whereas the simulation model was developed by using the FlexSim simulation tool [46].

3.2. Multiobjective Simulation-Based Optimization Algorithm. The fundamental of the hybrid AIS-based optimization algorithm [45] adopted in the framework was inspired from

mechanisms of biological immune system and biological evolution. It was developed by hybridizing the clonal selection principle and immune network theory with the idea from Genetic Algorithm (GA). The algorithm makes use of the Pareto dominance for fitness assignment and some common AIS-based algorithm's features for guiding the search process, such as clonal selection and expansion, affinity maturation, antibody concentration, metadynamics, and immune memory. The interesting feature of this algorithm is the introduction of an innovative suppression operator, which is used to help eliminate similar antibodies, hence significantly minimizing the number of unnecessary searches and increasing the population diversity. The similarity among antibodies is determined in terms of both the objective space and the decision variable space to ensure that only similar antibodies are eliminated in the suppression operation. Moreover, a modified crossover operator originated from the biological evolution was also developed to help further enhance the diversity of the clone population and the convergence of the algorithm because some good genes from the elite parents can be passed to the offspring for facilitating the search of optimal solutions, otherwise it may take a longer time to converge towards the Pareto front [27] especially in simulation-based optimization context. The mapping between the biological immune system and the proposed artificial one is given in Table 1.

The algorithm comprises five immune operators: cloning operator, hypermutation operator, suppression operator, selection and receptor editing operator, and memory updating operator, and one genetic operator: crossover operator. Each of them takes responsibility for different tasks for the purpose of finding uniformly distributed Pareto front. The block diagram showing the computational steps for SCMIA is presented in Figure 2 and the details of these operators are provided as below.

Nondominated Neighbor-Based Selection. Based on the idea proposed by Gong et al. [42], only nondominated antibodies are selected to form an active parent population $\mathbf{A}(t)$ (where t is the iteration counter) with the size of N_A for undergoing cloning, crossover, and hypermutation. This process mimics the biological immune system in which only those B-cells (antibodies) that are capable of binding with foreign antigens will undergo clonal expansion and then hypermutation.

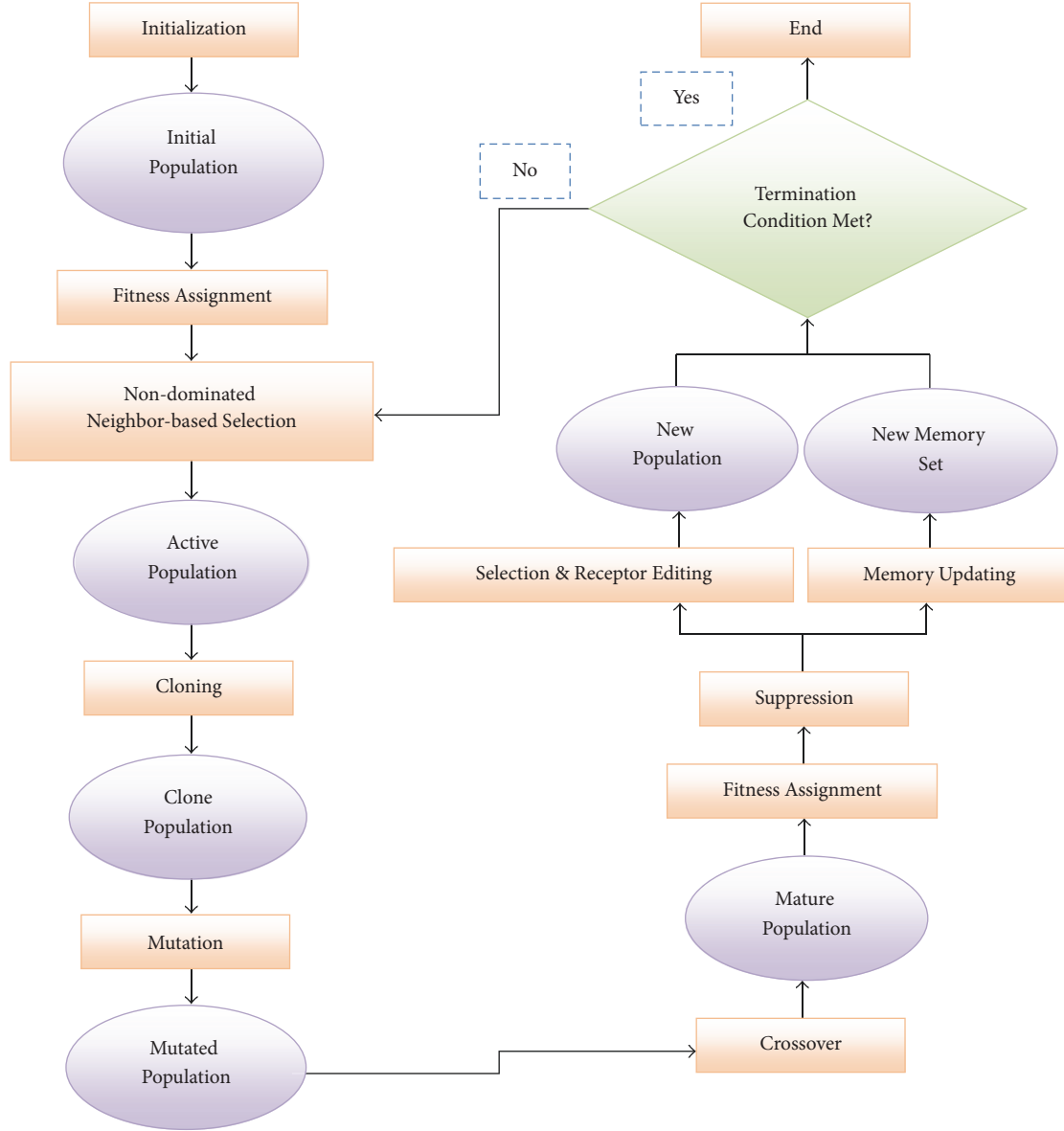


FIGURE 2: Computational steps for SCMIA.

However, if the number of nondominated antibodies is larger than N_A , an antibody density measure called crowding-distance [6] analogous with Ab-Ab affinity in biological immune system is employed for selecting the nondominated antibodies. The crowding-distance or Ab-Ab affinity for a nondominated antibody is computed as follows:

$$d(ab_i^*) = \sum_{j=1}^r \frac{|ab_{i+1}^*(f_j) - ab_{i-1}^*(f_j)|}{ab(f_j^{\max}) - ab(f_j^{\min})}, \quad (5)$$

where $d(ab_i^*)$ is the crowding-distance of the i th nondominated antibody ab_i^* , r is the number of objectives, $ab(f_j^{\max})$ and $ab(f_j^{\min})$ are the maximum and minimum fitness values of the j th objective, and $ab_{i+1}^*(f_j)$ and $ab_{i-1}^*(f_j)$ are the fitness values of the nearest neighboring antibodies from both sides.

Cloning Operator. It enlarges the population by generating a number of copies of each antibody in the active parent population $\mathbf{A}(t)$ and the number of copies is directly proportional to its Ab-Ab affinity, thus forming a clone population, which is denoted as $\mathbf{C}(t)$. Hence the size of the population now is $N_A + N_c$ and N_c is obtained by

$$\begin{aligned} c_i &= \text{round}(c_{\max} \times d(ab_i^*)), \\ N_c &= \sum c_i, \end{aligned} \quad (6)$$

where N_c is the total number of copies produced, c_i is the clone size for the antibody $ab_i^* \in \mathbf{A}(t)$, c_{\max} is the predefined maximum clone size of each antibody, and $\text{round}()$ is an operator for rounding its argument to the closest integer. Thus, the higher the Ab-Ab affinity an antibody has, the more the number of copies it generates.

Hypermutation Operator. It brings multipoint mutations to the pool of clones, hoping for producing better offspring. The mutations are dependent on the Ab-Ab affinity of their parents for the purpose of preventing the crowding of antibodies and maintaining the population diversity. The mechanism is that if the Ab-Ab affinity of an active parent is lower, the mutation rate for the clones of the parent should be higher to obtain mutated children in sparse locations in order to diversify the population. The clones are mutated proportionally as follows:

$$\alpha = e^{-p \times d(ab^*)} \quad (7)$$

$$\mathbf{C}^m(t) = \mathbf{C}(t) + \alpha \times R,$$

where α represents the mutation rate inversely proportional to the Ab-Ab affinity $d(ab^*)$, p is an exponential coefficient controlling the decay of α , $R \in [-1, 1]$ is a m -dimensional random vector obtained with uniform distribution, and $\mathbf{C}^m(t)$ is the mutated clone population.

Crossover Operator. It works on the mutated clone population to generate a mature clone population denoted as $\mathbf{C}^c(t)$ with the size of N_c . It is a modified single point crossover operator through which offspring is generated by randomly selecting a single crossover point on a clone and then swapping its content beyond that point with that of a parent antibody randomly selected from $\mathbf{A}(t)$. This operator is introduced to control the diversity of the clone population and the convergence of the algorithm. That being said, the diversity can be enhanced while the quick convergence can be ensured because some good genes from the active parent can be passed to the offspring.

Suppression Operator. It works on the whole population including the parent cells and mutated clones $\mathbf{A}(t) \cup \mathbf{C}^c(t)$ to eliminate similar individuals so as to avoid a particular search space being over exploited and acquire the uniformly distributed Pareto front based on the idea of immune network theory [47] such that B-cells are stimulated and suppressed by not only non-self-antigens but the interacted B-cells. Different from other AIS-based multiobjective optimization algorithms, the similarity among antibodies in this algorithm is determined in terms of both the objective space and the decision variable space so as to determine whether to retain or discard individual antibodies. Therefore, the suppression operation in this algorithm has two phases: in first phase, the suppression will be applied to all antibodies and the similarity between two antibodies is defined as follows:

$$dO(ab_a, ab_b)_j = |ab_a(f_j) - ab_b(f_j)| \leq \delta_j, \quad (8)$$

where $dO(ab_a, ab_b)_j$ is the distance between antibodies ab_a and ab_b in terms of j th objective and δ_j refers to the threshold value for j th objective.

In this phase, if the distances for all objectives between two antibodies are smaller than the thresholds, the two antibodies are said to be similar and hence the cell with poorer Pareto fitness pf will be suppressed and eliminated from the population.

In second phase, the suppression will only be applied to the similarity between nondominated cells and dominated cells and the similarity between two antibodies is defined as follows:

$$dV(ab_a, ab_b) = \sqrt{\sum_{j=1}^m [ab_a(x_j) - ab_b(x_j)]^2} \leq \varepsilon, \quad (9)$$

where $dV(ab_a, ab_b)$ is the Euclidean distance in decision variable space between the two antibodies ab_a (dominated cell) and ab_b (nondominated cell) $\in \mathbf{A}(t) \cup \mathbf{C}^c(t)$, m is the length of each antibody, and ε refers to the threshold value for the decision variable space.

In this phase, if the distance between two cells is smaller than the threshold in decision variable space, the two cells are said to be similar and hence the dominated cell will be suppressed and eliminated from the population. Eventually, surviving populations $\mathbf{A}^s(t) \cup \mathbf{C}^s(t)$ are obtained and then enter into the selection and receptor editing process and memory updating process simultaneously.

To enhance the population diversity and facilitate the search of uniformly distributed nondominated solutions, the threshold values for the decision variable space and the objective space are adaptive values that are dynamically calculated based on the maximum and minimum values found at each iteration. The adaptive threshold values are computed according to the following equations:

$$\varepsilon = \frac{ab(x^{\max}) - ab(x^{\min})}{n}, \quad (10)$$

$$\delta_j = \frac{ab(f_j^{\max}) - ab(f_j^{\min})}{n},$$

where $ab(x^{\max})$ and $ab(x^{\min})$ are the maximum and minimum distance values in the decision variable space, $ab(f_j^{\max})$ and $ab(f_j^{\min})$ are the maximum and minimum fitness values of the j th objective, and n is the number of antibodies in the population under investigation.

Selection and Receptor Editing Operator. It works like a director to guide the search towards the promising regions by selecting all nondominated antibodies with respect to the Pareto fitness pf from the populations $\mathbf{A}^s(t) \cup \mathbf{C}^s(t)$ to form a new population $\mathbf{Ab}(t + 1)$ with the size of N for the next generation $t + 1$. If the new population $\mathbf{Ab}(t + 1)$ is not full, dominated antibodies with a better Pareto fitness pf are selected and some genes of these antibodies are then randomly selected to be replaced by randomly generated genes. These restructured antibodies are finally added to the new population until the population is full in order to further enhance the population diversity. This process actually mimics the process of receptor editing in the biological immune system. However, if the number of nondominated antibodies found exceeds the limit of the population size N , only N nondominated antibodies with higher Ab-Ab affinity $d(ab)$ are selected.

Memory Updating Operator. It works as an elitist mechanism for helping preserve the best solutions that represent the Pareto front found over the search process. The memory set is updated at each iteration through the following procedure:

- (1) If the size of $\mathbf{P}(t)$ is equal to N_m , there is no memory archiving procedure to proceed.
- (2) If the size of $\mathbf{P}(t)$ is smaller than N_m , an archiving procedure is triggered to copy some of the dominated antibodies that have the best Pareto fitness pf and Ab-Ab affinity $d(ab)$ to $\mathbf{P}(t)$ for filling the vacant space up.
- (3) If the size of $\mathbf{P}(t)$ is larger than N_m , an archiving procedure is triggered to compress $\mathbf{P}(t)$ based on the antibody similarity (the measure used in the suppression operator) until its size becomes N_m .

After performing the final memory updating procedure at the last iteration, the memory set $\mathbf{P}(t)$ obtained represents the resulting solution set of the algorithm.

The algorithm is conducted by applying these heuristic and stochastic operators on the antibody population for balancing both the local and global search capabilities. For the detailed discussion of the hybrid multiobjective algorithm, one can refer to [45].

4. Multiobjective Simulation-Based Optimization Study

In this study, a set of experiments based on a real-life multiobjective optimization problem was performed to evaluate the performance and capability of the optimization framework and algorithm. In order to provide a better view on the performance of each method, both the graphical presentations of the simulation results and the statistical results of performance metrics on the problem under investigation were analyzed. All these experiments were conducted using a computer with Xeon E5-2620 2 GHz CPU with 2 GB RAM.

4.1. Performance Metrics. In this simulation-based optimization study, two performance metrics, namely, error ratio (ER) [48] and spacing (S) [49], are adopted to examine the quality of solution set in terms of the optimality and diversity.

Error Ratio (ER). The PF_{true} was originally used to compute this metric. However, since the PF_{true} can hardly be found for the real-life problem under investigation in this paper, we use the reference Pareto front “ PF_{ref} ”, that is, the best approximation of the true Pareto front for measuring the optimality of each solution set. The PF_{ref} is found by using all of the algorithms used in this study. To achieve this, a set of nondominated solutions, that is, a Pareto front for each algorithm, is firstly generated by running a very large number of iterations, e.g., 100 over 20 trials, and then all the fronts obtained by all the trials of all the compared algorithms were

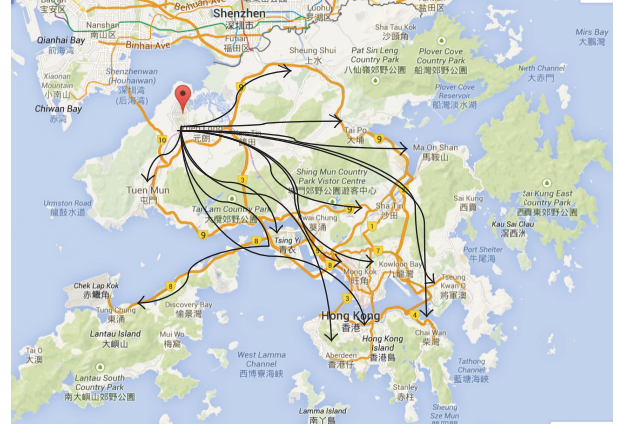


FIGURE 3: The distribution flow of S.F. Express's imported goods in Hong Kong.

merged together to form a reference Pareto front. Hence, this metric is mathematically defined as

$$\text{ER} := \frac{\sum_{i=1}^n e_i}{n}, \quad e_i = \begin{cases} 0, & x_i \in \text{PF}_{\text{ref}}, \\ 1, & x_i \notin \text{PF}_{\text{ref}}, \end{cases} \quad (11)$$

where n is the number of solutions in the PF_{known} found by the algorithm being evaluated, $e_i = 0$ if solution i belongs to the PF_{ref} , and $e_i = 1$ otherwise. $\text{ER} = 0$ indicates that all the generated solutions belong to the PF_{ref} .

Spacing (S). This metric is used for determining the diversity of the generated solutions. That being said, it shows how well the solutions are distributed in the PF_{known} . This metric is mathematically defined as

$$S := \sqrt{\frac{\sum_{i=1}^n (\bar{d} - d_i)^2}{n-1}}, \quad (12)$$

$$d_i = \min_j \left(\sum_{k=1}^m |f_k^i(x) - f_k^j(x)| \right), \quad i, j = 1, \dots, n,$$

where \bar{d} is the mean of all d_i , k is the number of objective functions, and n is the number of solutions in the PF_{known} . $S = 0$ indicates that all the nondominated solutions found are equally spaced and uniformly distributed in objective space.

4.2. Experimental Setup. In this section, a real-life multiobjective optimization problem, that is, the distribution operation of the material handling system (MHS) implemented at the distribution center (DC) of S.F. Express (Hong Kong) Limited, was studied through the simulation-based optimization approach. S.F. Express launched its service in Hong Kong in 1993. At present, its service network is covering almost all areas of Hong Kong, which is mainly served by the DC located in Tin Shui Wai (TSW) in northern New Territories (Figure 3). Its service stores are located in 30 areas of Hong Kong [50].



FIGURE 4: Inbound docks connecting to the conveyor system.

The DC has two major conveyor systems: one for handling the exported goods and another for imported goods. In this study, we focus on the physical goods flow at the DC, where the items are imported from China and then distributed to all parts of Hong Kong. At the DC, most of the items received at inbound docks (Figure 4) are directly transferred to outbound docks (Figure 5) and then shipped to the final destinations with little material handling in between, such as deconsolidation and sortation. As such, the expensive physical distribution functions of putaway, inventory holding, and order picking from the DC are eliminated. This approach is called cross-docking. To implement cross-docking effectively and efficiently, timely distribution of freight and better synchronization of all inbound and outbound shipments are required by making use of the information system infrastructure and advanced automated MHS in the supply chain, such as automated conveyor system, Warehouse Management System (WMS), and real-time material identification and tracking system (e.g., barcode).

4.2.1. Current Physical Layout and Labor Deployment of the MHS. The MHS is a circular shaped automated conveyor system, which comprises a number of interconnected 5-, 10-, and 20-meter long straight modular conveyor units and 2.5-meter radius curved conveyor units. The layout of the MHS is depicted in Figure 7. Each conveyor unit has a programmable logic controller to control the movement (speed, direction, acceleration, etc.) of the items being put on it and to communicate with the central computer. The conveyor system has 4 entrances connecting to 4 inbound docks and 16 exits connecting 30 outbound docks (each outbound dock serves trucks distributing parcels to one destination) (Figure 6).

At each conveyor entrance, 7 workers are deployed to deconsolidate the incoming bulky consolidated parcels uploaded from the big inbound truck (16 tonnes) for facilitating the subsequent sortation process. To enable the distribution process to go well and items to be accurately sorted according to customer requirements, 4 workers are assigned to each conveyor exit serving 2 destinations (except the 2 exits close to the entrances serving 1 destination that requires 2 workers) and equipped a barcode reader for confirming the destination of each small parcel and helping to load the parcel to the small outbound truck (5.5 tonnes).

4.2.2. Operation of the MHS. Each step of the operation is performed sequentially in the simulation process and the operation is given as follows:

- (1) When the operation starts, each of the incoming bulky consolidated parcels is unloaded by the workers from the 4 inbound trucks to the staging area of the 4 inbound docks.
- (2) And then each bulky consolidated parcel is deconsolidated into multiple small parcels and transported to the conveyor entrances by the workers.
- (3) After the parcels arrive at the conveyor, they are transported to the different exits of the conveyor system according to their destinations.
- (4) When the parcels reach the corresponding exits, they are moved to the outbound docks by the workers and ready for delivery.

4.2.3. Assumptions. To focus our study on the behavior of the MHS, certain real-world factors were simplified. Thus, the following assumptions were made:

- (1) Every day, the DC handles around 3 to 4 batches of parcels. Since the operation is similar in each batch, only one batch in the simulation model is simulated and studied. There are 400 bulky consolidated parcels in each batch coming from 4 supply sources (each contributes 100 bulky parcels) in which the processing time (cycle time) in one batch and the workers' utilization are studied.
- (2) The time for unloading bulky parcels follows a uniform distribution.
- (3) The time for deconsolidation of the bulky parcels follows a normal distribution in which each bulky parcel is separated into 10 small parcels. Therefore, there are 4000 small parcels in total being handled by the MHS.
- (4) The system only processes two types of parcels (namely, bulky consolidated parcels and small parcels) because they are packed similarly in terms of volume and weight.
- (5) The demand for each destination is similar which follows a uniform distribution, that is, the parcels are uniformly distributed among the 30 destinations.

4.2.4. Simulation Model. The simulation model with specific system configuration and behavior is implemented with FlexSim (Figure 7). Details of model components and initial model settings (Table 2) are as follows:

- (i) Entities are the 400 bulky consolidated parcels and the 4000 small parcels deconsolidated from the bulky parcels, which are processed through the system causing changes in the system state over time.
- (ii) Activities are the deconsolidation of bulky consolidated parcels unloaded from each incoming truck (represented by a source in the simulation model), the

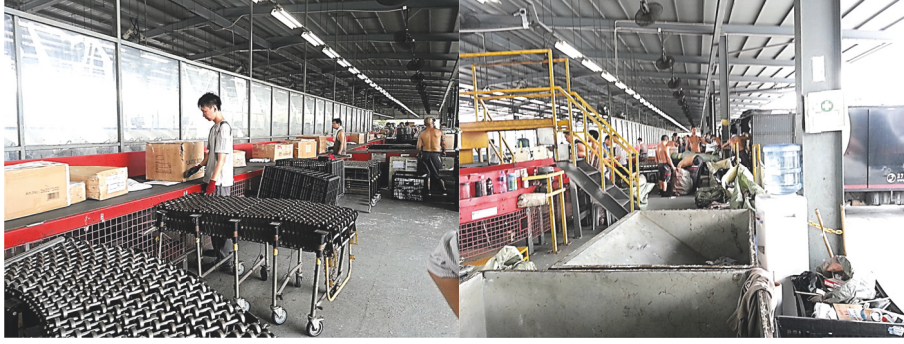


FIGURE 5: Outbound docks connecting to the conveyor system.



FIGURE 6: The docks and trucks at the DC.

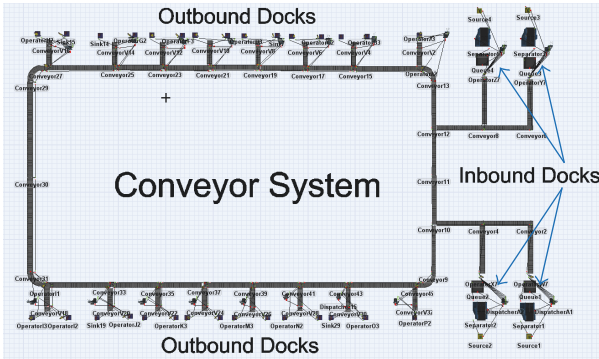


FIGURE 7: The simulation model of the MHS implemented with FlexSim.

picking of small parcels from the circular conveyor to each outgoing truck (represented by a sink in the simulation model) according to their destinations.

- (iii) Item attribute is the product type associated with its destination (one product type corresponds to one specific destination).

4.2.5. Parameter Setting. Several parameters, including number of generations, initial population size, active population size, maximum clone size, and mutation factor are studied through sensitivity analysis so as to examine the individual parameters' impact on the overall performance of the system and to determine the value for each parameter. Given with the results of the sensitivity analysis, the parameters of SCMIA were set as follows.

TABLE 2: Initial model settings.

Item	Value
Conveyor speed	2.5 m/sec (limit: 1–2.5 m/sec)
Conveyor spacing	1 parcel
Number of workers deployed for each conveyor entrance	7 workers (limit: 1–9 workers)
Number of workers deployed for each conveyor exit (serving 1 destination)	2 workers (limit: 1–4 workers)
Number of workers deployed for each conveyor exit (serving 2 destinations)	4 workers (limit: 1–6 workers)
Handling Capacity of Worker	1 parcel
Arrival pattern for each source	Uniform distribution with a min. of 5 sec and a max. of 10 sec
Processing time of deconsolidation process	Normal distribution with a mean of 30 sec and a standard deviation of 2 sec
Demand for each destination	Uniform distribution with a min. of 220 units and a max. of 280 units

The above model parameters are set based on the real system settings and observation.

SCMIA. Initial population size is $N = 30$; size of active population, $N_A = 12$; size of the memory population, $N_m = 30$; maximum number of clones for each cell, $\text{max_clone} = 6$; exponential distribution coefficient, $\rho = 0.05$; number of simulation replications per fitness evaluation, $\text{replication} = 10$.

To allow a fair comparison among the algorithms compared, the parameters of the benchmarking algorithms were set with similar values and the values suggested by the authors.

MISA. Initial population size is $N = 30$; size of clone population, $N_c = 180$; size of the memory population, $N_m = 30$; number of grid subdivisions, $\text{subd_size} = 25$; initial mutation rate, $\omega = 0.6$ (it decreases linearly over time until reaching the rate of $1/m$, where m is the number of decision

variables); number of simulation replications per fitness evaluation, replication = 10.

NNIA. Size of dominant population is $N_p = 30$; size of active population is $N_A = 8$; size of clone population is $N_c = 30$; crossover probability is $p_c = 0.9$; mutation probability is $p_m = 1/m$; distribution indexes for crossover and mutation operators are n_c and $n_m = 20$ (crossover probability, $p_c = 1$; mutation probability, $p_m = 1/m$); number of simulation replications per fitness evaluation is replication = 10.

NSGA-II. Initial population size is $N = 30$; crossover probability is $p_c = 0.9$; mutation probability is $p_m = 1/m$; distribution indexes for crossover and mutation operators are n_c and $n_m = 20$; number of simulation replications per fitness evaluation is replication = 10.

SPEA2. Initial population size is $N = 30$; archive size, $N_m = 30$; crossover probability, $p_c = 1$; mutation probability, $p_m = 0.006$; number of simulation replications per fitness evaluation, replication = 10.

Antibody Definition. An antibody ab (a set of decision variables) that has the direct impact on the system's performance in terms of cycle time (CT) and workers' utilization (WU) is defined as follows: x_1 is taken to be the conveyor speed, x_2 is the number of workers deployed for each conveyor entrance, x_3 is the number of workers of workers deployed for each conveyor exit (serving 2 destinations), and x_4 is the number of workers deployed for each conveyor exit (serving 1 destination). Since the objective of the study is to optimize the performance of the MHS by minimizing the system cycle time and maximizing the workers' utilization, the optimization problem can be given by

$$\underset{ab \in \Omega}{\text{Optimize}} \vec{f}(ab) = E[CT, WU, \omega] \quad (13)$$

subject to

$$\begin{aligned} 1 &\leq x_1 \leq 2.5, \\ 1 &\leq x_2 \leq 9, \\ 1 &\leq x_3 \leq 6, \\ 1 &\leq x_4 \leq 4, \end{aligned} \quad (14)$$

where a set of objective functions $\vec{f}(ab)$ to be optimized in objective space are the expected values of the random output variables $[CT, WU, \omega]$ that are obtained from running the simulation model, ω is a sample path (i.e., the sequence of random numbers used in a simulation run), and (14) define a set of physical constraints.

4.2.6. Experimental Results and Analysis. We conducted two experiments to evaluate the performance of the SCMIA and the simulation-based optimization framework, that is, (1) to compare the results of integrating simulation and

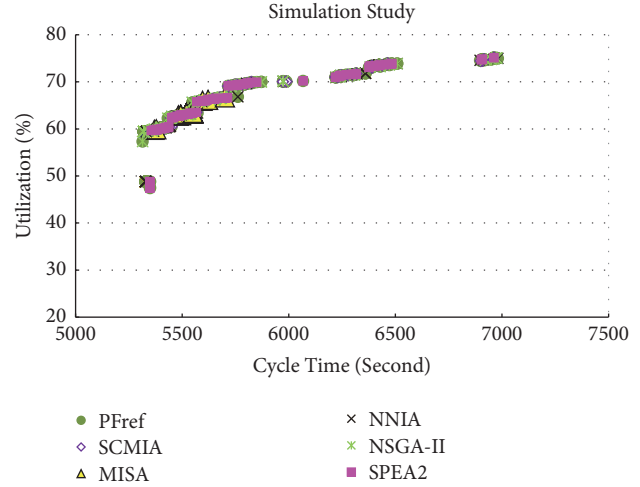


FIGURE 8: Graphical comparisons of the known Pareto fronts generated by the five algorithms.

optimization with the results without using any optimization algorithm and (2) to benchmark SCMIA against two immune-inspired algorithms, MISA [20] and NNIA [42], and two other evolutionary algorithms, NSGA-II [35] and SPEA2 [37], under the framework. The first experiment was conducted to evaluate the effectiveness of optimization algorithms when being employed in the simulation study while the second one was used to emphasize the comparison among the algorithms under investigation with respect to the optimality and the diversity using the same simulation-based optimization framework. All algorithms were run for 30 generations over 30 trials to obtain the average performance of each algorithm on the same condition.

(1) *Simulation without Optimization versus Simulation with Optimization.* The results shown in Table 3 are the optimized results obtained by making use of all optimization algorithms studied in this research.

The table shows that the cycle time of the whole distribution system at the DC is reduced by about 12–16% and the workers' utilization is increased by 38–49% when optimization algorithms are deployed in the simulation process. This proves that the use of the optimizers can enhance the performance in the system's cycle time and the workers' utilization. However, the higher the utilization achieved, the longer the cycle time spent, and vice versa. When comparing SCMIA with other benchmark algorithms, SCMIA is able to produce comparable results in both the cycle time and workers' utilization.

(2) *Performance Comparison between SCMIA and Other Benchmark Algorithms.* The performance of the algorithms studied in this study was compared by using the graphical representation together with the application of the metrics mentioned in Section 4.1. The comparison between the known Pareto fronts shown in Figure 8 suggests that the overall Pareto front patterns generated by the five algorithms are largely similar in which the cycle time ranges from around

TABLE 3: Performance comparison between the results obtained from the simulation alone and that from the simulation-based optimization approach by using different optimization algorithms (the best results are bolded).

	Cycle Time (the improvement in % compared with the one without optimization)	Workers' Utilization (the improvement in % compared with the one without optimization)
Simulation without optimization	6788.64 sec	45.04%
Simulation-based optimization with SCMIA	5763.86 sec (15.10%)	65.46% (45.34%)
Simulation-based optimization with MISA	5754.88 sec (15.23%)	63.45% (40.87%)
Simulation-based optimization with NNIA	5672.87 sec (16.44%)	62.00% (37.66%)
Simulation-based optimization with NSGA-II	5680.23 sec (16.33%)	62.90% (39.65%)
Simulation-based optimization with SPEA2	5948.53 sec (12.38%)	67.10% (48.76%)

TABLE 4: Spacing and error ratio values generated by the five algorithms (the best results are bolded).

	SCMIA	MISA	NNIA Mean (Standard Deviation)	NSGA-II	SPEA2
Error Ratio (ER)	0.71 (0.10)	0.85 (0.16)	0.78 (0.18)	0.76 (0.14)	0.74 (0.06)
Spacing (S)	38.74 (10.97)	58.02 (55.82)	49.23 (26.10)	51.85 (17.58)	39.32 (37.16)

TABLE 5: Computational times of the five algorithms.

	SCMIA	MISA	NNIA	NSGA-II	SPEA2
Time (hours)	33.75	104.19	24.46	24.54	23.75

5300 sec to 7000 sec and the workers' utilization ranges from around 50% to 75%. From Figure 8, it is shown that the higher the utilization achieved, the longer the cycle time spent, and vice versa. This implies that increasing the utilization does not mean that the efficiency of the system can be increased. Therefore, according to the figure, although the optimal solution with 75% utilization while spending almost 7000 sec and another case achieving less than 50% utilization while spending only around 5300 sec both fall within the reference Pareto front PF_{ref} ; the latter case is preferred in practice because the cycle time is much shorter and hence the efficiency and productivity of the system are much higher. The utilization becoming lower in the latter case is mainly due to the increased values in the decision variables, namely, conveyor speed and number of workers. This implies that in order to further enhance both objectives at the same time for the DC, the company may need to do something other than just changing the system parameters, such as redesigning the layout of the material handling systems particularly the conveyor system.

The performance regarding the optimality and diversity of SCMIA in this multiobjective optimization problem was then examined by applying the metrics, namely, spacing and error ratio as shown in Table 4.

In this experiment, we compared the results of the mean and standard deviation of the two metrics over 30 trials obtained by SCMIA with that of the other benchmark

algorithms. From the results shown in Table 4, we found that SCMIA generally is able to provide the best results in terms of the diversity and optimality because it generates the lowest values in the metrics of ER (0.71) and S (38.74) and the latter metric is significantly lower than most of the other algorithms. This implies that the generated front is very close to the PF_{ref} . In terms of the stability, SCMIA is also the best one among these five algorithms in error ratio and spacing because it has much lower standard deviations (0.10) and (10.97), respectively, than other algorithms except SPEA2, implying that SCMIA is able to provide a relatively consistent result for each trial.

(3) *Comparison of Computational Time.* The computational times of the investigated algorithms are provided in Table 5, which are the mean times of 30 trials of each algorithm running for 30 generations. It can be observed that the computational time of MISA is the longest (104.19 hours) because MISA spends very much time in solution evaluation through running the complex simulation model due to the largest number of candidate solutions being generated and evaluated at each iteration. This is very time-consuming. SCMIA comes second (33.75 hours), although the performance in terms of computational time is much better than MISA. The reason is that SCMIA also generates a large number of solutions at each iteration, but the suppression operator adopted in the algorithm helps reduce the number of similar solutions and

hence reduce the number of solution evaluations. On the contrary, NNIA, NSGA-II, and SPEA2 consume less and similar computational resources largely due to the much smaller number of solution evaluations being conducted at each iteration.

5. Conclusion

This study applies a multiobjective simulation-based optimization framework incorporating a hybrid immune optimization algorithm SCMIA for the evaluation of the optimality of the distribution system with respect to two criteria: system cycle time and workers' utilization through simulation modeling. Based on the results of the simulation-based optimization study, the following conclusions and implications can be drawn regarding the performance of the framework and algorithm.

SCMIA generally performs better than other benchmark algorithms especially in the diversity aspect. This is largely attributed to the operators employed in the algorithm. For example, the selection operator incorporates the crowding-distance as a measure to select nondominated antibodies for undergoing the subsequent evolutionary processes so that the antibodies in less crowded regions will have a higher priority to be selected. The cloning operator and hypermutation operator are based on the same measure to generate a number of copies for exploring the solution space and bringing variation to the clone population, respectively, where less crowded individuals are given more chances for cloning and hypermutation in order to hopefully produce better offspring and increase population diversity. The crossover operator helps further enhance the diversity of the clone population and the convergence of the algorithm because some good genes from the active parent can be passed to the offspring. The suppression operator helps reduce antibody redundancy by eliminating similar individuals, hence significantly minimizing the number of unnecessary searches and increasing the population diversity. The memory updating operator takes account of the antibody similarity in terms of both the objective space and the decision variable space to formulate the memory population. As a result, SCMIA is able to generate a well-distributed set of solutions while it is a good approximation to the reference Pareto front. Other than the optimality and diversity, the stability of each algorithm can also be observed based on the standard deviations of the metrics shown in Table 4. It can be seen that the standard deviation of spacing obtained by SCMIA is much smaller than the other four algorithms. In a word, SCMIA is the most stable one among the five algorithms in terms of population diversity in simulation-based optimization.

The results overall demonstrate the ability of the multiobjective simulation-based optimization framework to serve as a decision support tool for helping management to effectively and efficiently find near optimal system operating parameters such as the speed of machines, the number of workers, or any other decision variables of interest in order to fulfill different objectives including system's cycle time and workers' utilization. As a result, significant savings in money, energy,

and so forth, are achieved through the effective and efficient deployment of material handling systems and well-coordinated activities based on the optimized results. The research also sheds light into important issues of logistics system operation and management based on the case study, such as manpower allocation and design capacity of facilities.

Based on the findings of the current undertaking, it is worthwhile to extend the framework to tackle other complex problems involving many objectives to be solved in an efficient and effective manner in the future. Future research could also extend this approach to solve other real-world complex business problems other than the problems associated with material handling systems so as to establish the practical value of the framework in the simulation-based optimization context.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

- [1] S. L. Rosen, *Automated Simulation Optimization of Systems with Multiple Performance Measures Through Preference Modeling*, Pennsylvania State University, State College, Pa, USA, 2003.
- [2] S. Kirkpatrick, J. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *American Association for the Advancement of Science: Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [3] F. Glover, "Heuristics for integer programming using surrogate constraints," *Decision Sciences*, vol. 8, no. 1, pp. 156–166, 1977.
- [4] F. Glover, "Tabu search—part I," *ORSA Journal on Computing*, vol. 1, no. 3, pp. 190–206, 1989.
- [5] F. Glover, "Tabu search—part II," *ORSA Journal on Computing*, vol. 2, no. 1, pp. 4–32, 1990.
- [6] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [7] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.
- [8] J. D. Knowles and D. W. Corne, "Approximating the non-dominated front using the pareto archived evolution strategy," *Evolutionary Computation*, vol. 8, no. 2, pp. 149–172, 2000.
- [9] L. N. de Castro and J. Timmis, *Artificial Immune Systems: A New Computational Intelligence Approach*, Springer, London, UK, 2002.
- [10] J. Banks, I. John, S. Carson, B. L. Nelson, and D. M. Nicol, *Discrete-Event System Simulation*, Prentice Hall, 4th edition, 2010.
- [11] Operations Management, "Importance and scope of material handling," March 2007, <https://www.citeman.com/1413-importance-and-scope-of-material-handling.html>.
- [12] V. B. Norman, *Simulation of Automated Material Handling And Storage Systems*, Auerbach, Princeton, NJ, USA, 1984.
- [13] M. K. Ebbesen, M. R. Hansen, and N. L. Pedersen, "Design optimization of conveyor systems," in *III European Conference on Computational Mechanics*, C. A. Mota Soares, J. A. C. Martins, H. C. Rodrigues, J. A. C. Ambrósio, C. A. B. Pina, and C. M. Mota Soares, Eds., pp. 721–721, Springer, Netherlands, 2006.

- [14] S. V. Sergueyevich, M. G. O. Rosales, J. M. García, L. A. Z. Quintana, and G. R. P. López, "Chain conveyor system simulation and optimization," in *Proceedings of the 17th IASTED International Conference on Modelling and Simulation*, pp. 172–177, Canada, May 2006.
- [15] M. M. L. Elahi, G. V. Záruba, J. Rosenberger, and K. Rajpurohit, *Modeling and Simulation of a General Motors Conveyor System Using a Custom Decision Optimizer*, University of Texas, Arlington, Va, USA, 2009.
- [16] C. S. K. Leung and H. Lau, "An optimization framework for modeling and simulation of dynamic systems based on AIS," in *Proceedings of the International Federation of Automatic Control World Congress*, Italy.
- [17] K. Subulan and M. Cakmakci, "A feasibility study using simulation-based optimization and Taguchi experimental design method for material handling-transfer system in the automobile industry," *The International Journal of Advanced Manufacturing Technology*, vol. 59, no. 5-8, pp. 433–443, 2012.
- [18] K.-H. Chang, A.-L. Chang, and C.-Y. Kuo, "A simulation-based framework for multi-objective vehicle fleet sizing of automated material handling systems: An empirical study," *Journal of Simulation*, vol. 8, no. 4, pp. 271–280, 2014.
- [19] J. T. Lin and C.-J. Huang, "Simulation-based evolution algorithm for automated material handling system in a semiconductor fabrication plant," in *Proceedings of the 4th International Asia Conference on Industrial Engineering and Management Innovation, IEMI 2013*, pp. 1035–1046, twN, July 2013.
- [20] C. A. C. Coello and N. C. Cortés, "Solving multiobjective optimization problems using an artificial immune system," *Genetic Programming and Evolvable Machines*, vol. 6, no. 2, pp. 163–190, 2005.
- [21] F. Y. Edgeworth, "Mathematical psychics," *Mind*, vol. 6, pp. 581–583, 1881.
- [22] V. Pareto, *Cours d'Économie Politique*, vol. 1, F. Rouge, Lausanne, Switzerland, 1896.
- [23] V. Pareto, *Cours d'Économie Politique*, vol. 2, F. Rouge, Lausanne, Switzerland, 1897.
- [24] C. M. Fonseca and Fleming P. J., "Genetic algorithms for multiobjective optimization: formulation discussion and generalization," in *Proceedings of the 5th International Conference on Genetic Algorithms (ICGA '93)*, pp. 416–423, 1993.
- [25] Y. Hu and T. Chen, "Multi-objective optimization algorithm based on clonal selection," in *Proceedings of the Second International Conference on Genetic and Evolutionary Computing*, pp. 265–268, 2008.
- [26] J. Gao and J. Wang, "WBMOAIS: A novel artificial immune system for multiobjective optimization," *Computers & Operations Research*, vol. 37, no. 1, pp. 50–61, 2010.
- [27] C. A. C. Coello, G. B. Lamont, and D. A. van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, vol. 5, Springer, New York, NY, USA, 2nd edition, 2007.
- [28] D. A. van Veldhuizen and G. B. Lamont, "On measuring multi-objective evolutionary algorithm performance," in *Proceedings of the 2000 Congress on Evolutionary Computation*, pp. 204–211, La Jolla, Calif, USA, July 2000.
- [29] K. Miettinen, *Nonlinear Multiobjective Optimization*, Kluwer Academic Publishers, Norwell, Mass, USA, 1999.
- [30] G. P. Coelho, F. O. De Franca, and F. J. V. Zuben, "A concentration-based artificial immune network for combinatorial optimization," in *Proceedings of the IEEE Congress of Evolutionary Computation (CEC '11)*, pp. 1242–1249, New Orleans, La, USA, June 2011.
- [31] K. Deb, *Multiobjective Optimization Using Evolutionary Algorithms*, John Wiley & Sons Inc., Chichester, UK, 2001.
- [32] E. Y. C. Wong, H. S. C. Yeung, and H. Y. K. Lau, "Immunity-based hybrid evolutionary algorithm for multi-objective optimization in global container repositioning," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 6, pp. 842–854, 2009.
- [33] J. D. Schaffer, "Multiple Objective Optimization with Vector Evaluated Genetic Algorithms," in *Proceedings of the 1st International Conference on Genetic Algorithms*, pp. 93–100, 1985.
- [34] J. Knowles and D. Corne, "The pareto archived evolution strategy: a new baseline algorithm for Pareto multiobjective optimisation," in *Proceedings of the Congress on Evolutionary Computation (CEC '99)*, vol. 1, pp. 98–105, July 1999.
- [35] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, pp. 849–858, 2000.
- [36] D. W. Corne, J. D. Knowles, and M. J. Oates, "The Pareto-envelope based selection algorithm for multiobjective optimization," in *Parallel Problem Solving from Nature PPSN VI*, vol. 1917 of *Lecture Notes in Computer Science*, pp. 869–878, Springer, New York, NY, USA, 2000.
- [37] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," in *Computer Engineering and Communication Networks Lab (TIK)*, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 2001.
- [38] D. W. Corne, N. R. Jerram, J. Knowles, and M. J. Oates, "PESA-II: Regionbased selection in evolutionary multiobjective optimization," in *Proceeding of the Genetic and Evolutionary Computation Conference*, pp. 283–290, San Francisco, Calif, USA, 2001.
- [39] C. A. Coello Coello and G. T. Pulido, "A micro-genetic algorithm for multiobjective optimization," in *Evolutionary multi-criterion optimization (Zurich, 2001)*, vol. 1993 of *Lecture Notes in Comput. Sci.*, pp. 126–140, Springer, Berlin, 2001.
- [40] G. Toscano Pulido and A. Coello Coello, "The micro genetic algorithm 2: towards online adaptation in evolutionary multi-objective optimization," in *Proceedings of Evolutionary Multi-Criterion Optimization: Second International Conference, EMO 2003, Faro, Portugal, April 8–11, 2003*, C. Fonseca, P. Fleming, E. Zitzler, L. Thiele, and K. Deb, Eds., vol. 2632, pp. 252–266, Springer, Berlin, Germany, 2003.
- [41] G. P. Coelho and F. J. Von Zuben, *omni-aiNet: An Immune-Inspired Approach for Omni Optimization*, 2006.
- [42] M. Gong, L. Jiao, H. Du, and L. Bo, "Multiobjective immune algorithm with nondominated neighbor-based selection," *Evolutionary Computation*, vol. 16, no. 2, pp. 225–255, 2008.
- [43] Z. Zhang, "Artificial immune optimization system solving constrained omni-optimization," *Evolutionary Intelligence*, vol. 4, no. 4, pp. 203–218, 2011.
- [44] T. Niknam, R. Azizpanah-Abarghoee, and M. Rasoul Nari-mani, "A new multi objective optimization approach based on TLBO for location of automatic voltage regulators in distribution systems," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 8, pp. 1577–1588, 2012.
- [45] C. S. Leung and H. Y. Lau, "A Hybrid Multi-objective Immune Algorithm for Numerical Optimization," in *Proceedings of the*

8th International Conference on Evolutionary Computation Theory and Applications, pp. 105–114, Porto, Portugal, November 2016.

- [46] Flexsim Software Products Inc, <https://www.flexsim.com/>, 2017.
- [47] N. K. Jerne, “Towards a network theory of the immune system,” *Annales D’Immunologie*, vol. 125, no. C, pp. 373–389, 1974.
- [48] D. A. Van Veldhuizen, *Multiobjective Evolutionary Algorithms: Classifications, Analyses, And New Innovations*, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, USA, 1999.
- [49] J. Schott, *Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization*, Massachusetts Institute of Technology, Cambridge, Mass, USA, 1995.
- [50] S.F. Express (Hong Kong) Limited, <http://www.sf-express.com/hk/tc/>, 2018.

