Research Article

Modeling and Optimization of the Multiobjective Stochastic Joint Replenishment and Delivery Problem under Supply Chain Environment

Lin Wang,¹ Hui Qu,¹ Shan Liu,² and Cai-xia Dun¹

¹ School of Management, Huazhong University of Science and Technology, Wuhan 430074, China

² Economics and Management School, Wuhan University, Wuhan 430072, China

Correspondence should be addressed to Shan Liu; 279266384@qq.com

Received 21 July 2013; Accepted 18 September 2013

Academic Editors: S. Choi and S. Comai

Copyright © 2013 Lin Wang et al. 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.

As a practical inventory and transportation problem, it is important to synthesize several objectives for the joint replenishment and delivery (JRD) decision. In this paper, a new multiobjective stochastic JRD (MSJRD) of the one-warehouse and n -retailer systems considering the balance of service level and total cost simultaneously is proposed. The goal of this problem is to decide the reasonable replenishment interval, safety stock factor, and traveling routing. Secondly, two approaches are designed to handle this complex multi-objective optimization problem. Linear programming (LP) approach converts the multi-objective to single objective, while a multi-objective evolution algorithm (MOEA) solves a multi-objective problem directly. Thirdly, three intelligent optimization algorithms, differential evolution algorithm (DE), hybrid DE (HDE), and genetic algorithm (GA), are utilized in LP-based and MOEA-based approaches. Results of the MSJRD with LP-based and MOEA-based approaches are compared by a contrastive numerical example. To analyse the nondominated solution of MOEA, a metric is also used to measure the distribution of the last generation solution. Results show that HDE outperforms DE and GA whenever LP or MOEA is adopted.

1. Introduction

The joint replenishment problem (JRP) is a practical inventory problem of a group of products that can be jointly ordered from a single supplier (Goyal [1]; Wang et al. [2]), which can help save the ordering costs and inventory holding costs. According to the characteristic of demand, the existing study of JRPs can be divided into two categories: (1) constant demand; (2) stochastic or dynamic demand. An extensive literature review is available in Khouja and Goyal [3] and Narayanan et al. [4]. Many scholars also discussed more realistic JRPs (J.-M. Chen and T.-H. Chen [5]; Axsäter et al. [6]; Hsu [7]; Abdul-Jalbar et al. [8]).

Many companies have realized that a joint replenishment and delivery scheduling (JRD) policy can result in considerable cost savings. But the literature on the JRDs under supply chain environment is limited. A stochastic JRD of the one-warehouse, n -retailer system has been formulated (Qu et al. [9]). Wang et al. [10] studied the same JRD but

reduced the decision variables using specific mathematical method and provided a new differential evolution algorithm. Sindhuchoo et al. [11] studied the coordinated inventory and transportation decisions with the vehicle capacity limitation in an inbound commodity collection system. Chan et al. [12] addressed issues in scheduling of the multi-item, multibuyer, and single supplier system. Cha et al. [13] handled the JRD of the one-warehouse and n -retailer system in which the warehouse supplies items from the supplier and delivers them to retailers. Moon et al. [14] modified the model of [13] by utilizing a consolidated freight delivery policies. Wang et al. [2] extended the JRD model of Cha et al. [13] under fuzzy environment and used the widely used signed distance method to ranking fuzzy numbers. Wang et al. [15] studied the JRD with deterministic demand and fuzzy cost using the graded mean integration representation and centroid approaches to defuzzify the total costs. A common limitation in all the literature studies mentioned above is that they only consider a single objective.

However, managers are usually faced with complex multiobjective optimization problems (MOPs) in reality. For the JRD policy, it is necessary to decrease the total cost while improving the service level. Although there are several papers that studied multiobjective inventory models (Roy and Maiti [16]; Rong et al. [17]; Islam [18]; Wee et al. [19]), no study on the multiobjective JRD can be found.

For MOPs, direct comparison among the solutions is very difficult because of the different measurements between each contradicted target. In this study, total cost and service level are obviously two contradictory targets: reducing total cost may result in the decline of service level and vice versa. So we should coordinate two targets. Different from a single-objective optimization problem which has unique optimal solution, an MOP has a set of optimal solutions called Pareto optimal solutions. Due to the characteristics of MOPs, they are much more complex and the key is to find an effective method to obtain Pareto optimal solutions. Unfortunately, the classical JRPs and JRDs are already NP hard problems (Arkin et al. [20]), and the multiobjective makes the JRDs become much more difficult to handle.

Many linear or nonlinear weighted methods (Rong et al. [17]; Islam [18]; Wee et al. [19]; Roy and Maiti [16]) were used to convert the multiobjective to a single one in the existing studies. These methods undoubtedly provide one easy way to deal with the multiobjective JRD model. However, these approaches do not solve the MOPs intrinsically, since the solutions for MOPs are multiple rather than one. On the other hand, multiobjective optimization methods based on Pareto-based MOEAs are widely used, such as multiobjective genetic algorithm (MOGA) (Aiello et al. [21]), nondominated sorting genetic algorithm (NSGA) (Lin [22]), and strength Pareto evolutionary algorithm (SPEA) (Zitzler and Thiele [23]; Sheng et al. [24]).

In recent years, several MOEAs based on Pareto differential evolution (DE) were utilized to solve MOPs. Santana-Quintero and Coello [25] presented a DE-based multiobjective algorithm using a secondary population and the concept of ϵ -dominance. The performance of the proposed algorithm was also compared with NSGA-II and ϵ -MOEA. Qian et al. [26] proposed a memetic algorithm based on DE (MODEMA) for multiobjective job shop scheduling problems. Qian and Li [27] proposed an adaptive DE (ADEA) and the results of five test functions showed that the ADEA was very efficient to find out the true Pareto front. However, the majority of the above studies focused on the effectiveness verification of the algorithms and always analyzed standard testing functions and ignored the practical applications of them in MOPs. Due to the existing unpredictable and uncertain factors of inventory management, it is difficult to convert shortage rate/quantity to shortage cost. Therefore, discussing shortage rate/quantity independently is meaningful.

The aim of this study is to propose a new multiobjective stochastic JRD (MSJRD) model including two minimum objectives, that is, the total cost and shortage quantity. The main difference between this study and [10] is that two objectives are handled simultaneously. Moreover, effective approaches are provided to handle this MSJRD. Having the successful applications in engineering management, as well

as the effectiveness of DE in solving MOPs and JRPs/JRDs (Wang et al. [28–30]), linear programming (LP) and MOEA-based approaches using DE are provided. Results of an example show that the proposed hybrid DE is more effective than the original DE and GA whatever LP or MOEA method is used.

The rest of this paper is organized as follows. Section 2 describes the proposed multiobjective stochastic JRD model. Section 3 introduces the hybrid DE. Section 4 presents two approaches to solve the MSJRD. Section 5 contains numerical examples and results. Section 6 discusses conclusions and provides future research directions.

2. Formulation of the Proposed MSJRD Model

Consider an enterprise has a center warehouse in a proper place and several suppliers in decentralized locations. The center warehouse jointly replenishes items from its suppliers according to the market demand or historical data. Then, the center warehouse will collect replenished items from suppliers. In this situation, the center warehouse can jointly determine the replenishment and distribution policy to obtain the optimal decision.

Referring to the model of Qu et al. [9], the differences between the JRD policy and typical JRP can be concluded as follows: (a) the JRP supposes deterministic demand, while our model considers stochastic demand and allows for shortage; (b) the JRP just discusses the replenishment policy, while our model analyses not only replenishment but also transportation decision; (c) the JRP is a single objective model, while our model is a multiobjective model.

For the MSJRD, reducing the related total cost as well as decreasing shortage quantity should be considered simultaneously. Therefore, the first target is to minimize total cost which consists of replenishment cost, inventory holding cost, and distribution cost. Minimizing the shortage quantity is the second target.

2.1. The First Target: Total Cost. The total cost includes inventory holding cost, replenishment cost, and distribution cost. Distribution cost involves the stopover cost at suppliers and cost related to distance. The infinite capacity of vehicle assumption makes the distribution problem a traveling salesman problem (TSP).

Inventory cost can be given as

$$C_H = \sum_{i=1}^n h_i \left[\frac{1}{2} D_i k_i T + z_i \sigma_i \sqrt{(L + k_i T)} \right]. \quad (1)$$

In (1), the first term is deterministic inventory and the second one is safety stock. The following replenishment cost is the same as the classical JRP:

$$C_S = \frac{S}{T} + \sum_{i=1}^n \frac{s_i}{k_i T} = \frac{1}{T} \left(S + \sum_{i=1}^n \frac{s_i}{k_i} \right). \quad (2)$$

When k_i are given, taking the least common multiple of k_i we can get integer M . It means that the replenishment and

distribution behavior will be repeated every M period. For example, if $k_i = (4, 2, 1, 1)$, then $M = 4$. That is to say, in period 1, all items should be replenished; in period 2, items 3 and 4 need to be replenished; in period 3, items 2, 3, and 4 should be replenished; in period 4, items 3 and 4 should be replenished. It is obvious that in period 5 the situation is the same as in period 1. The cycle period is M . Therefore, a limited horizon with M periods is used to calculate annual distribution cost.

Jointly replenishment of items can take the advantage of scale economies not only in the replenishment but also in distribution process. Four items specifically should be replenished in period 1 and it is advisable for the center warehouse to traverse suppliers that supply these items at one time. The shortest path in each period is considered so distribution cost can be obtained by solving a travelling salesman problem (TSP). The distribution cost is

$$C_T = \frac{\sum_{j=1}^M (\sum_{p=1}^P x_{pj} F_p + cd(j))}{MT}, \quad (3)$$

where $x_{pj} = \begin{cases} 1, & \text{if supplier } p \text{ is visited} \\ 0, & \text{otherwise.} \end{cases}$

Therefore, the first target can be summarized as follows:

$$\begin{aligned} F_1(T, k_i, z_i) &= C_H + C_S + C_T \\ &= \sum_{i=1}^n h_i \left[\frac{1}{2} D_i k_i T + z_i \sigma_i \sqrt{(L + k_i T)} \right] \\ &\quad + \frac{1}{T} \left(S + \sum_{i=1}^n \frac{s_i}{k_i} \right) \\ &\quad + \frac{\sum_{j=1}^M (\sum_{p=1}^P x_{pj} F_p + cd(j))}{MT}. \end{aligned} \quad (4)$$

2.2. The Second Target: Stock-Out Quantity. In this study, the demand is assumed to follow normal distribution at interval of unit time, which is widely used in the literature (see Qu et al. [9]; Eynan and Kropp [31, 32]). This assumption means, that given k_i and T for item i , the demand will follow normal distribution over the interval of length $L + k_i T$ with the expectation $E = D_i(L + k_i T)$, variance $\text{Var} = \sigma_i^2(L + k_i T)$ and probability density function $f(x_i, L + k_i T)$.

With a periodic replenishment policy, stockout could occur any time during replenishment intervals as long as the real demand exceeds maximum inventory level R_i . The total annual stock-out quantity is

$$\begin{aligned} F_2(T, k_i, z_i) &= \frac{\sum_{i=1}^n \left(\int_{R_i}^{\infty} (x_i - R_i) f(x_i, L + k_i T) dx_i \right)}{k_i T} \\ &= \sum_{i=1}^n \frac{\sigma_i}{k_i T} \sqrt{k_i T + L} \int_{z_i}^{\infty} (y - z_i) f(y) dy \end{aligned}$$

$$\begin{aligned} &= \sum_{i=1}^n \frac{\sigma_i}{k_i T} \sqrt{k_i T + L} \\ &\quad \times \left(\int_{z_i}^{\infty} y f(y) dy - z_i \int_{z_i}^{\infty} f(y) dy \right) \\ &= \sum_{i=1}^n \frac{\sigma_i}{k_i T} \sqrt{k_i T + L} (f(z_i) - z_i [1 - F(z_i)]), \end{aligned} \quad (5)$$

where $R_i = D_i(k_i T + L) + z_i \sigma_i \sqrt{(k_i T + L)}$.

2.3. The MSJRD Model. The whole multiobjective model can be written as

$$\begin{aligned} \text{Min } F_1(T, k_i, z_i) &= \sum_{i=1}^n h_i \left[\frac{1}{2} D_i k_i T + z_i \sigma_i \sqrt{(L + k_i T)} \right] \\ &\quad + \frac{1}{T} \left(S + g(k_i) + \sum_{i=1}^n \frac{s_i}{k_i} \right), \\ \text{Min } F_2(T, k_i, z_i) &= \sum_{i=1}^n \frac{\sigma_i}{k_i T} \sqrt{k_i T + L} \\ &\quad \times (f(z_i) - z_i [1 - F(z_i)]), \end{aligned} \quad (6)$$

where $g(k_i) = \sum_{j=1}^M (\sum_{p=1}^P x_{pj} F_p + cd(j)) / M$.

The goal of this multiobjective model is to find out the optimal k_i , z_i , and T to simultaneously minimize the total cost and stock-out quantity and thus to achieve Pareto solutions in which two objectives can be balanced. Two targets have different units of measurements and it is usually difficult to convert the shortage quantity to stock-out cost. In addition, they are often in conflict with each other, that is, decreasing shortage quantity may result in cost increasing.

MOPs are much more complex but closer to reality. Several traditional mathematic methods are used for solving multiobjective models, such as linear programming, goal programming, and analytic hierarchy process. However, they are successful only in small scale problems. Mathematic methods are too complex and too time consuming to solve large scale problems. In the following, we provide two common approaches based on an HDE to deal with the proposed MSJRD. Then a numerical example and comparative study between the proposed LP and MOEA are presented.

3. The Hybrid Differential Evolution Algorithm (HDE)

3.1. The Classical DE. DE has been described as an effective and robust method to optimize some well-known nonlinear, nondifferentiable, and nonconvex functions. Due to its easy implementation, quick convergence, and robustness, DE has turned to be one of the best evolutionary algorithms in a variety of fields (Wang et al. [33]; Cui et al. [34]). DE contains three operations: mutation, crossover, and selection.

3.1.1. Mutation. The mutation operation creates a new vector by adding the weighted difference of two random vectors to a third one. For each target vector x_t^G ($t = 1, 2 \dots NP$), the mutated vector is created as follows:

$$v_t^{G+1} = x_{r_1}^G + F \times (x_{r_2}^G - x_{r_3}^G). \quad (7)$$

In (7), r_1 , r_2 , and r_3 , are three serial numbers of vectors, which are randomly generated with different values and none of them equals t . Three vectors $x_{r_1}^G$, $x_{r_2}^G$, and $x_{r_3}^G$ will be selected from the population for mutation operation when r_1 , r_2 , and r_3 are confirmed, F is a scaling factor and G is the current number of iteration.

3.1.2. Crossover. A trial vector is created by mixing the mutated vector with the target vector according to the following formula:

$$u_{tj}^{G+1} = \begin{cases} v_{tj}^{G+1}, & \text{if } randm(j) \leq CR \text{ or } j = randn(t) \\ x_{tj}^G, & \text{otherwise,} \end{cases} \quad (8)$$

where j represents the j th dimension; $randm(j)$ is randomly generated from 0 to 1; $randn(t) \in [1, 2, \dots, D]$ is a randomly selected integer to ensure the effect of mutated vector; CR is the crossover probability and it is very important for DE since the larger CR is, the more v_t^{G+1} contributes to u_t^{G+1} .

3.1.3. Selection. The selection operation is implemented by comparing the trial vector (obtained through mutation and crossover operations) with the corresponding target vector. For example, to minimize the function, the next generation is formed by

$$x_t^{G+1} = \begin{cases} u_t^{G+1}, & \text{if } f(u_t^{G+1}) < f(x_t^G), \\ x_t^G, & \text{otherwise,} \end{cases} \quad (9)$$

where the $f(\cdot)$ is the fitness function of DE.

Here, an example is given to illustrate the three operations mentioned previously. For the current number of iteration G and target vector x_1^G , suppose that random generated numbers r_1 , r_2 , and r_3 are 23, 40, and NP respectively, and we obtain the following:

Target vectors x^G :

Vector x_1 : 6 5 2 4 0.090

...

Vector x_{23} : 5 4 2 1 0.578

...

Vector x_{40} : 5 1 3 6 0.745

...

Vector x_{NP} : 4 2 5 7 0.024.

(10)

Mutation: if $F = 0.6$, the mutated vector v_1^{G+1} can be obtained by (7) as follows:

Mutated vectors v^{G+1} :

Vector v_1 : 5.6 3.4 0.8 0.4 1.010 (11)

...

Crossover: if $CR = 0.3$, $randn(t) = 3$ (here $t = 1$), and vector $randm = (0.1, 0.4, 0.5, 0.2, 0.6)$, the trial vector can be obtained by (8) as follows:

Trial vectors u^{G+1} :

Vector u_1 : 5.6 5 0.8 0.4 0.090 (12)

...

Selection: then target x_1 should be compared with u_1 . Since $f(u_1^{G+1}) < f(x_1^G)$, vector u_1 should be selected to the next generation as follows:

Next generation x^{G+1} :

Vector 1: 5.6 5 0.8 0.4 0.090 (13)

...

3.2. The Proposed Hybrid DE (HDE). The typical DE is simple and easy to be implemented. However, it is likely to be premature too early. One-to-one competing is one of the main reasons. Therefore, improvements including dynamic parameter adjusting, different mutation and crossover strategies, or hybrid algorithms are necessary to be adopted.

Similar to DE, a genetic algorithm (GA) contains crossover, mutation, and selection operations. The crossover operation of GA is quite complicated and its complexity may grow rapidly when the problem scale becomes larger. Fortunately, GA has several efficient selection operations such as roulette wheel selection, tournament selection, and truncation selection. In this study, an HDE that combines the advantages of DE and GA is proposed. The proposed HDE can simplify the evolutionary process, and it can overcome the limitation of one-to-one selection of DE and thus prevent premature convergence.

Actually, several scholars also proposed hybrid DEs based on DE and GA (Hrstka and Kučerová [35]; He et al. [36]; Lin [37]), but their mixing modes are quite different from ours. In the proposed HDE, the mutation and crossover operations are the same as in DE while the selection operation is from truncation selection of GA. That is to say, it will be reserved instead of comparing with the target vector when a trial vector is generated. When all trial and target vectors are determined, top NP vectors with better performance are selected to the next generation. The HDE-based procedure is shown in Figure 1.

4. Two Methods for Solving MSJRD

4.1. Linear Programming (LP) Approach for the MSJRD. This method is to summarize the weighted targets and thus converts the multiobjective model to a single one. Take into consideration that two targets have different measurements; it is necessary to standardize two targets beforehand.

4.1.1. Model Analysis Using Linear Programming Approach. Suppose that the weights of two objectives are w_1 and w_2 , and then the multiobjective problem can be described as

$$\begin{aligned} \text{Max} \quad & \lambda = w_1 u_1 + w_2 u_2 \\ \text{s.t.} \quad & \begin{cases} u_1 = \frac{F_1^{\max}(T, k_i, z_i) - F_1(T, k_i, z_i)}{F_1^{\max}(T, k_i, z_i) - F_1^{\min}(T, k_i, z_i)} \\ u_2 = \frac{F_2^{\max}(T, k_i, z_i) - F_2(T, k_i, z_i)}{F_2^{\max}(T, k_i, z_i) - F_2^{\min}(T, k_i, z_i)} \\ w_1 + w_2 = 1, \end{cases} \end{aligned} \quad (14)$$

$F_1^{\max}(T, k_i, z_i)$, $F_1^{\min}(T, k_i, z_i)$, $F_2^{\max}(T, k_i, z_i)$, and $F_2^{\min}(T, k_i, z_i)$ are the tolerant maximum total cost, minimum total cost, maximum stock-out quantity, and minimum stock-out quantity, respectively, which can be given in advance by decision makers (Wee et al. [18]).

Set $w'_1 = w_1/(F_1^{\max} - F_1^{\min})$, $w'_2 = w_2/(F_2^{\max} - F_2^{\min})$. Then, the objective function is changed to

$$\text{Max} \quad \lambda = F_c - (w'_1 F_1 + w'_2 F_2), \quad (15)$$

where $F_c = w_1 F_1^{\max}/(F_1^{\max} - F_1^{\min}) + w_2 F_2^{\max}/(F_2^{\max} - F_2^{\min})$ and

$$\begin{aligned} F_w &= w'_1 F_1 + w'_2 F_2 \\ &= w'_1 \left(\sum_{i=1}^n h_i \left[\frac{1}{2} D_i k_i T + z_i \sigma_i \sqrt{(L + k_i T)} \right] \right. \\ &\quad \left. + \frac{1}{T} \left[S + g(k_i) + \sum_{i=1}^n \frac{s_i}{k_i} \right] \right) \\ &\quad + w'_2 \left(\sum_{i=1}^n \frac{\sigma_i}{k_i T} \sqrt{k_i T + L} (f(z_i) - z_i [1 - F(z_i)]) \right). \end{aligned} \quad (16)$$

Let $\partial F_w / \partial z_i = 0$, that is, $w'_1 h_i \sigma_i \sqrt{k_i T + L} + w'_2 (\sigma_i / k_i T) \sqrt{k_i T + L} (df(z_i)/dz_i - [1 - F(z_i)] + z_i f(z_i)) = 0$.

Note that for standard normal distribution, $df(z_i)/dz_i = -z_i f(z_i)$, so that

$$1 - F[z_i(k_i T)] = \frac{w'_1 h_i}{w'_2} k_i T. \quad (17)$$

That is to say, when k_i and T is known, the optimal value of z_i must satisfy (17).

Taking the second derivation of $TC(T, k_i, z_i)$ with respect to z_i , we obtain $\partial^2 TC(T, k_i, z_i) / \partial z_i^2 = (w'_2 \sigma_i / k_i T) \sqrt{k_i T + L} f(z_i) > 0$, which means that the optimal z_i is derived from (17) and $z_i^*(k_i T) = F^{-1}(1 - (w'_1 h_i / w'_2) k_i T)$.

Substituting $z_i^*(k_i T)$ into (16), the optimal value of F_w for given k_i and T is

$$\begin{aligned} F_w &= w'_1 \left(\sum_{i=1}^n \frac{D_i h_i k_i}{2} T + \frac{1}{T} \left[S + g(k_i) + \sum_{i=1}^n \frac{s_i}{k_i} \right] \right) \\ &\quad + w'_2 \left(\sum_{i=1}^n \frac{\sigma_i}{k_i T} \sqrt{k_i T + L} f(z_i^*) \right). \end{aligned} \quad (18)$$

Finally, the linear programming model can be written as

$$\begin{aligned} \text{Max} \quad & \lambda(T, k_i) \\ &= F_c - w'_1 \left(\sum_{i=1}^n \frac{D_i h_i k_i}{2} T + \frac{1}{T} \left[S + g(k_i) + \sum_{i=1}^n \frac{s_i}{k_i} \right] \right) \\ &\quad - w'_2 \left(\sum_{i=1}^n \frac{\sigma_i}{k_i T} \sqrt{k_i T + L} f(z_i^*) \right) \\ \text{s.t.} \quad & \begin{cases} F_c = \frac{w_1 F_1^{\max}}{F_1^{\max} - F_1^{\min}} + \frac{w_2 F_2^{\max}}{F_2^{\max} - F_2^{\min}}, \\ w'_1 = \frac{w_1}{F_1^{\max} - F_1^{\min}}, \\ w'_2 = \frac{w_2}{F_2^{\max} - F_2^{\min}}, \\ w_1 + w_2 = 1 \\ z_i^*(k_i T) = F^{-1} \left(1 - \frac{w'_1 h_i}{w'_2} k_i T \right). \end{cases} \end{aligned} \quad (19)$$

The goal is to determine the best k_i and T to maximize λ for the given w_1 and w_2 .

4.1.2. HDE-Based Procedures for MSJRD Using LP Approach. When k_i are determined, the optimal delivery cost can be calculated by solving a TSP. $z_i^*(k_i T)$ can be calculated by $z_i^*(k_i T) = F^{-1}(1 - (w'_1 h_i / w'_2) k_i T)$ when k_i and T are known. Change k_i and T with the following steps until maximum λ is obtained.

Step 1. Initialization: set related parameters (CR, F, and NP) for HDE. Set the lower bound (k_i^{LB}) and the upper bound (k_i^{UB}) of k_i respectively. Note that k_i are integers so, k_i^{LB} is obviously 1. According to experience of [2, 10, 15], k_i^{UB} is set sufficiently large to guarantee that the optimal solution does not escape. In this study, it can be set to 100. T is randomly generated in the range of 0 and 1. Combining k_i and T we get the t th individual $x_t = (k_i, T)$. Create initial population randomly.

Step 2. For a given w_1 , calculate the objective function. When x_t is determined, $z_i^*(k_i T)$ can be derived accordingly. x_t and $z_i^*(k_i T)$ jointly determine λ .

FIGURE 1: Flow chart of HDE.

Step 3. Differential operations: while stopping criterion is not met, implement mutation and crossover for each individual. After that, the number of population is two times the original one.

Step 4. Genetic operations: select the individuals according to λ . Those with larger λ will be chosen to the next generation. Then the number of population is the same as the original one.

Step 5. When the number of iteration reaches a predefined maximum number, output the optimal results; otherwise, repeat Steps 2–4.

4.2. Multiobjective Evolution Algorithm (MOEA) Approach for the MSJRD. In this section, a brief introduction of MOP is given. Then, an HDE-based procedure to handle the MSJRD using noninferior and crowding distance is designed.

4.2.1. Some Definitions of MOP

Definition 1 (multiobjective optimization problems (MOP)).

$$\begin{aligned} \text{Min } F(x) &= f_1(x), f_2(x), \dots, f_k(x) \\ \text{Subject to: } g_i(x) &\leq 0, \quad i = 1, 2, \dots, m. \end{aligned} \quad (20)$$

A general MOP consists of n decision variables, k objective functions, and m constraints. In Definition 1, x refers to the decision space and $g_i(x)$ are constraints of MOP.

Definition 2 (Pareto optimal solution). The optimal solution of MOP is often referred to as the Pareto optimal solution. Let vector a belong to x and suppose that x^* is a subset of x . If there does not exist any vector in x^* that is better than a , then a is called noninferior solution (or Pareto optimal solution) of x^* . Moreover, if vector a is the noninferior solution of x , then vector a is the Pareto optimal of the MOP.

Step 1. For each $x_i \in I$, initialize distance $D_{(i)} = 0$
 Step 2. Calculate the crowding distance of x_i
 Step 2.1. Sorting I for each objective m , $I = \text{sort}(I, m)$
 $D(I_{(1)}) = D(I_{(1)}) + [2(I_{(2),m} - I_{(1),m})]^2$
 $D(I_I) = D(I_I) + [2(I_{I,m} - I_{I-1,m})]^2$
 For $i = 2, \dots, |I| - 1$
 $D(I_{(i)}) = D(I_{(i)}) + [(I_{(i+1),m} - I_{(i-1),m})]^2$
 Step 2.2. $D_{(i)} = \text{sqr}t(D_{(i)})$
 where $|I|$ is the number of individuals contained in I ; $I_{(i),m}$ is the objective value of individual i .

ALGORITHM 1: Steps of calculating crowding distance.

4.2.2. HDE-Based Procedures for MSJRD Using MOEA Approach. There exist many difficulties when applying DE to solve an MOP compared with single objective problem. The main challenges for solving MOP are as follows: how to generate offspring and how to keep Pareto solutions uniformly distributed. The classical DE is not suitable for an MOP since many good solutions may be abandoned due to its one-to-one competing mechanism. This will also be confirmed by a numerical example.

Therefore, we also use an HDE which uses truncation selection to choose next generation based on front rank and crowding distance adopted by Qian and li [27]. The steps of calculating crowding distance are presented in Algorithm 1.

In this algorithm, the low front rank corresponds to the high quality of a solution. As to the those individuals with the same front rank, the larger crowding distance means better distribution. Therefore, individuals with lower front rank and larger crowding distance are selected to the next generation.

The first target can be divided into an inventory problem and a delivery problem. When all k_i are determined, the optimal delivery cost can be calculated by solving a TSP. In addition, for a stochastic JRP with normal distributed demand, when k_i , z_i , and T are known, the stochastic JRP can then be solved. With the same value of k_i , z_i , and T in the second target, we can obtain the corresponding value of the second target. Then change k_i , z_i , and T with the following steps until the termination condition is satisfied. The steps of HDE-based approach are described as follows.

Step 1. Initialization: set related parameters (CR, F, and NP) for the HDE. Set the lower bound and the upper bound of k_i , respectively; that is, $k_i^{\text{LB}} = 1$ and $k_{ij}^{\text{UB}} = 100$. z_i is randomly generated in the range of 0 and 3, which can cover 99.7% of the demand. T is randomly generated in the range of 0 and 1. Combining k_i , z_i , and T we get the t th individual $x_t = (k_i, z_i, T)$. Create initial population randomly.

Step 2. Calculate the objective function, that is, the total cost and the total shortage quantity of all items.

Step 3. Calculate the Pareto front and crowding distance of each individual.

Step 4. Differential operations: while stopping criterion is not met, implement mutation and crossover for each individual.

TABLE 1: Supply relationship between items and suppliers.

	Supplier 1	Supplier 2	Supplier 3
Item 1	1	0	0
Item 2	0	1	0
Item 3	0	0	1
Item 4	0	0	1
F_p	40	50	60

TABLE 2: Parameters of items.

	Item 1	Item 2	Item 3	Item 4
D (unit/year)	600	900	1200	1000
σ^2 (unit ² /year)	800	600	700	500
L (year)	0.02	0.02	0.02	0.02
h (\$/unit/year)	5.6	21	42	15
s (\$/order)	25	14	20	30
π (\$/unit)	28	35	40	30

The other two parameters are as follows: $S = 100$ and $c = 0.5$.

After that, the number of population is two times the original one.

Step 5. Genetic operations: select the individuals according to the front rank and crowding distance. Then the number of population is the same as the original one.

Step 6. When the number of iteration reaches a predefined maximum number, output the optimal results; otherwise, repeat Steps 2–5.

5. Contrastive Example and Results Analysis

5.1. Basic Data of Numerical Example. The data come from Qu et al. [9]. Table 1 describes the items to be replenished and the center warehouse correspondingly. Tables 2 and 3 are the related parameters of items and distances between suppliers and warehouse, respectively.

In the following, two approaches named LP and MOEA are compared. The comparison contains two aspects: the Pareto solutions and some specific solutions obtained by each method. In the meanwhile, three algorithms used in each

TABLE 3: Distances between suppliers and warehouse.

	Warehouse	Supplier 1	Supplier 2	Supplier 3
Warehouse	0	11	9	7
Supplier 1	11	0	5	8
Supplier 2	9	5	0	10
Supplier 3	7	8	10	0

TABLE 4: Parameters of the algorithms.

Parameter	Value	Algorithms
Number of population, NP	200	HDE, DE, and GA
Maximum number of iteration, GenM	300	HDE, DE, and GA
Mutation factor, F	0.6	HDE and DE
Crossover rate, CR	0.3	HDE and DE
Probability of mutation, P_m	0.1	GA
Probability of crossover, P_c	0.9	GA

TABLE 5: Results for LP with HDE, DE, and GA ($w_1 = 0.56$).

	HDE	DE	GA
k	2, 1, 1, 1	2, 1, 1, 1	2, 1, 1, 1
z	1.67, 1.35, 0.93, 1.53	1.67, 1.35, 0.93, 1.53	1.70, 1.38, 0.96, 1.56
T	0.0824	0.0824	0.0785
F_1	8468.21	8468.21	8487.52
F_2	17.44	17.44	16.91
λ	0.7553	0.7553	0.7536

method are compared with each other. Table 4 reports related parameters of HDE, DE, and GA.

For LP-based approach, we directly set $F_1^{\max}(T, k_i, z_i) = 10500$, $F_1^{\min}(T, k_i, z_i) = 7500$, $F_2^{\max}(T, k_i, z_i) = 120$, and $F_2^{\min}(T, k_i, z_i) = 0$ according to the advice of the decision makers. This approach is also widely used by other scholars (Wee et al. [18]).

5.2. Comparisons for LP-Based and MOEA-Based Solutions.

In this section, the above numerical example is handled using LP and MOEA. For LP, the weight of each objective must be assigned firstly. In order to compare with MOEA, the objectives can be converted to single index by setting the total cost and total shortage quantity with the same weights for MOEA when the Pareto solutions are obtained. The best results for LP when $w_1 = 0.56$ are presented in Table 5. As to MOEA, the highest index after converting is shown in Table 6.

Table 5 shows that HDE and DE are better than GA for LP; Table 6 implies that HDE is better than GA and GA is better than DE for MOEA. In order to further verify the conclusion, we obtained for different w_1 , w_1 is set from 0.1 to 0.9 and the results are reported in Table 7.

Set the total cost and total shortage quantity with the same weights w_1 and w_2 , respectively, for MOEA. Then, solutions with the highest weighted objective from the obtained Pareto solutions are shown in Table 8.

TABLE 6: Results for MOEA with HDE, DE, and GA ($w_1 = 0.56$).

	HDE	DE	GA
k	2, 1, 1, 1	3, 1, 1, 1	2, 1, 1, 1
z	1.59, 1.35, 0.96, 1.49	1.49, 1.64, 1.11, 1.67	1.55, 1.37, 0.99, 1.46
T	0.0721	0.0736	0.0721
F_1	8466.57	8587.38	8475.24
F_2	17.70	12.95	17.36
λ	0.7547	0.7495	0.7543

Table 7 shows that values of each target F_1 and F_2 change correspondently when w_1 varies, which means that different settings of the weight will result in different decisions. Moreover, when the weight of the first objective (total cost) equals 0.1, the solution is the best. When the difference between w_1 and w_2 becomes smaller, the solutions become worse. Table 8 implies a similar conclusion for HDE, DE, and GA.

From the comparisons for specific solutions, the following conclusion can be easily drawn: (1) HDE is better than DE or GA no matter whether LP or MOEA is adopted: HDE and DE are more suitable than GA when LP is used; HDE and GA are better than DE when MOEA is used. (2) Different weights for objectives will influence the solutions, for the conflicted objectives, and the assigned weights with large ratios (i.e., $w_1 : w_2 \geq 3 : 1$) may result in better solutions.

5.3. Nondominated Solution Analysis of MOEA. For the MOP, there have several metrics to evaluate the quality of the nondominated solutions (Robič and Filipič [38]). However, the implementation of most metrics needs a prerequisite; that is, the true Pareto front must be known. In this study, it is impossible to find the true Pareto front because the MSJRD is a practical problem. So we adopt the metric (*Spacing, SP*) used by Esparcia-Alcázar et al. [39, 40] to measure the distribution of solutions on the Pareto front by evaluating the variance of neighboring solutions. The lower value of SP means that better nondominated solution is obtained.

SP measures the relative distances between the members of Pareto front as

$$SP = \sqrt{\frac{\sum_{i=1}^n (\bar{d} - d_i)^2}{(n-1)}}, \quad (21)$$

where n is the number of the first nondominated solutions found. The distance d_i is given by

$$d_i = \min_j (|f_1^i(x) - f_1^j(x)| + |f_2^i(x) - f_2^j(x)|), \quad (22)$$

$$i, j = 1, \dots, n,$$

where $f_N^k(\cdot)$ is the fitness of point k on objective N and $\bar{d}$ is the mean of all d_i . Table 9 shows the mean and variance of SP by 10 runs using MOEA.

Table 9 shows that the mean and variance of SP obtained by HDE is the lowest, and corresponding values obtained by DE are biggest. That is to say, HDE is better than DE and

TABLE 7: Results for LP with HDE, DE, and GA (w_1 varies).

w_1	HDE			DE			GA		
	F_1	F_2	λ	F_1	F_2	λ	F_1	F_2	λ
0.1	9253.01	1.02	0.9339	9256.99	1.01	0.9339	9252.98	1.02	0.9339
0.2	9031.90	2.55	0.8809	9040.17	2.48	0.8808	9117.39	2.40	0.8762
0.3	8867.45	4.74	0.8356	8867.69	4.76	0.8354	8873.75	4.67	0.8354
0.4	8718.56	7.96	0.7977	8716.80	8.06	0.7975	8760.01	7.59	0.7941
0.5	8567.08	12.96	0.7682	8574.14	12.77	0.7678	8591.70	12.51	0.7659
0.6	8396.13	21.43	0.7493	8400.40	21.34	0.7488	8409.68	20.93	0.7483
0.7	8176.59	38.11	0.7468	8183.03	37.86	0.7460	8172.24	38.64	0.7465
0.8	7890.98	72.37	0.7751	7892.54	72.87	0.7739	7894.30	72.40	0.7742
0.9	7674.80	123.83	0.8444	7674.80	123.83	0.8444	7656.65	130.97	0.8439

TABLE 8: Results for MOEA with HDE, DE, and GA (w_1 varies).

w_1	HDE			DE			GA		
	F_1	F_2	λ	F_1	F_2	λ	F_1	F_2	λ
0.1	9253.73	1.03	0.9338	9285.77	0.89	0.9337	9225.35	1.17	0.9337
0.2	9046.73	2.42	0.8807	9017.73	2.72	0.8806	9031.13	2.59	0.8807
0.3	8886.03	4.48	0.8352	8878.41	4.77	0.8343	8848.34	5.17	0.8350
0.4	8720.04	8.02	0.7972	8768.47	7.00	0.7959	8739.46	7.55	0.7970
0.5	8545.97	13.94	0.7676	8587.38	12.95	0.7648	8617.66	11.16	0.7672
0.6	8423.72	20.00	0.7486	8587.38	12.95	0.7394	8359.45	23.93	0.7483
0.7	8207.26	35.76	0.7456	8348.16	32.22	0.7215	8229.71	33.76	0.7453
0.8	7913.56	69.71	0.7735	8348.16	32.22	0.7201	7892.98	73.03	0.7735
0.9	7685.28	121.57	0.8431	8348.16	32.22	0.7187	7711.52	117.17	0.8389

TABLE 9: Statistical analysis of SP by 10 runs.

	Mean	Variance
HDE	8.75	1.93
DE	17.48	10.48
GA	12.15	3.96

GA for the MOEA method. The conclusion is consistent with Section 5.2. At the same time, it verifies that the conversion using weights for the MSJRD is feasible.

In order to have a better understanding of the solutions' distribution of the last generation, the entire nondominated fronts found by HDE, DE, and GA are presented from Figure 2 to Figure 4.

Figures 2, 3, and 4 show that HDE and GA are capable of obtaining Pareto solutions, while the effectiveness and distribution of solutions are much worse for DE. The main reason is the one-to-one competing of DE. When trial individual dominates target individual, the trial vector (otherwise the target vector) remains to the next generation. This mechanism is different from the selection operation of the HDE, where all the target and trial individuals are kept to be chosen according to the front rank and crowding distance. So, DE is not suitable for solving this MSJRD.

FIGURE 2: Nondominated solutions of the final population obtained by HDE.

6. Conclusions and Future Research

In this study, a new multiobjective JRD model with stochastic demand is proposed which takes into account the service level while making the replenishment and delivery decisions. Then, two approaches to solve this complex optimization problem are designed using an improved HDE. The main contributions are as follows.

- (1) Considering the difficulty to estimate the shortage cost in reality, the shortage quantity is utilized as

FIGURE 3: Nondominated solutions of the final population obtained by DE.

FIGURE 4: Nondominated solutions of the final population obtained by GA.

another standard to evaluate the rationality of decisions besides the total cost. To our best knowledge, this is the first time to propose a practical multiobjective stochastic JRD model.

- (2) The MOEA is adopted to solve the proposed multiobjective JRD model. The results of the numerical example and Pareto solution analysis show the feasibility of the MOEA to handle the proposed MSJRD. It enriches the application field of the MOEA.
- (3) The comparison of two approaches for the MSJRD verifies that LP and MOEA are suitable for solving this MSJRD problem. Furthermore, results show that the proposed HDE is more effective than DE and GA whatever LP or MOEA method is used. This illustrates that DE is likely to combine with other algorithms so as to provide a more effective way to solve complex problems.

The future research on the multiobjective JRD problem should consider more realistic assumptions such as uncertain costs, freight consolidation, and budget constraint.

Abbreviations

D_i : Annual demand rate of item i
 S : Major ordering cost
 s_i : Minor ordering cost of item i

h_i : Annual inventory holding cost of item i
 T : Basic cycle time (decision variable)
 k_i : Multiplier of T (decision variable)
 T_i : Cycle time of item i (decision variable)
 R_i : Maximum inventory level of item i during replenishment interval
 z_i : Safety stock factor of item i (decision variable)
 L : Lead time of items
 σ_i : Standard deviation of item i
 M : Least common multiple of k_i
 π_i : Stock-out cost per unit for item i
 c : Distribution cost per unit distance
 F_p : Stopover cost at supplier p
 $d(j)$: Shortest path needed for distribution in j th basic cycle.

Acknowledgments

This research is partially supported by the National Natural Science Foundation of China (70801030; 71101060; 71131004; 71371080) and Humanities and Social Sciences Foundation of the Chinese Ministry of Education (no. 11YJC630275).

References

- [1] S. K. Goyal, "Determination of optimum packaging frequency of items jointly replenished," *Management Science*, vol. 21, no. 4, pp. 436–443, 1974.
- [2] L. Wang, C. X. Dun, C. G. Lee, Q. L. Fu, and Y. R. Zeng, "Model and algorithm for fuzzy joint replenishment and delivery scheduling without explicit membership function," *International Journal of Advanced Manufacturing Technology*, vol. 66, no. 9–12, pp. 1907–1920, 2013.
- [3] M. Khouja and S. Goyal, "A review of the joint replenishment problem literature: 1989–2005," *European Journal of Operational Research*, vol. 186, no. 1, pp. 1–16, 2008.
- [4] A. Narayanan, P. Robinson, and F. Sahin, "Coordinated deterministic dynamic demand lot-sizing problem: a review of models and algorithms," *Omega*, vol. 37, no. 1, pp. 3–15, 2009.
- [5] J.-M. Chen and T.-H. Chen, "The multi-item replenishment problem in a two-echelon supply chain: the effect of centralization versus decentralization," *Computers & Operations Research*, vol. 32, no. 12, pp. 3191–3207, 2005.
- [6] S. Axsäter, J. Marklund, and E. A. Silver, "Heuristic methods for centralized control of one-warehouse, N-retailer inventory systems," *Manufacturing & Service Operations Management*, vol. 4, no. 1, pp. 75–97, 2002.
- [7] S.-L. Hsu, "Optimal joint replenishment decisions for a central factory with multiple satellite factories," *Expert Systems with Applications*, vol. 36, no. 2, pp. 2494–2502, 2009.
- [8] B. Abdul-Jalbar, A. Segerstedt, J. Sicilia, and A. Nilsson, "A new heuristic to solve the one-warehouse N-retailer problem," *Computers & Operations Research*, vol. 37, no. 2, pp. 265–272, 2010.
- [9] W. W. Qu, J. H. Bookbinder, and P. Iyogun, "An integrated inventory-transportation system with modified periodic policy for multiple products," *European Journal of Operational Research*, vol. 115, no. 2, pp. 254–269, 1999.

- [10] L. Wang, C. X. Dun, W. J. Bi, and Y. R. Zeng, "An effective and efficient differential evolution algorithm for the integrated stochastic joint replenishment and delivery model," *Knowledge-Based Systems*, vol. 36, pp. 104–114, 2012.
- [11] S. Sindhuchao, H. E. Romeijn, E. Akçali, and R. Boondiskulchok, "An integrated inventory-routing system for multi-item joint replenishment with limited vehicle capacity," *Journal of Global Optimization*, vol. 32, no. 1, pp. 93–118, 2005.
- [12] C. K. Chan, L. Y.-O. Li, C. T. Ng, B. K.-S. Cheung, and A. Langevin, "Scheduling of multi-buyer joint replenishments," *International Journal of Production Economics*, vol. 102, no. 1, pp. 132–142, 2006.
- [13] B. C. Cha, I. K. Moon, and J. H. Park, "The joint replenishment and delivery scheduling of the one-warehouse, n-retailer system," *Transportation Research E*, vol. 44, no. 5, pp. 720–730, 2008.
- [14] I. K. Moon, B. C. Cha, and C. U. Lee, "The joint replenishment and freight consolidation of a warehouse in a supply chain," *International Journal of Production Economics*, vol. 133, no. 1, pp. 344–350, 2011.
- [15] L. Wang, H. Qu, Y. H. Li, and J. He, "Modeling and optimization of stochastic joint replenishment and delivery scheduling problem with uncertain costs," *Discrete Dynamics in Nature and Society*, vol. 2013, Article ID 657465, 12 pages, 2013.
- [16] T. K. Roy and M. Maiti, "Multi-objective inventory models of deteriorating items with some constraints in a fuzzy environment," *Computers & Operations Research*, vol. 25, no. 12, pp. 1085–1095, 1998.
- [17] M. Rong, N. K. Mahapatra, and M. Maiti, "A multi-objective wholesaler-retailers inventory-distribution model with controllable lead-time based on probabilistic fuzzy set and triangular fuzzy number," *Applied Mathematical Modelling*, vol. 32, no. 12, pp. 2670–2685, 2008.
- [18] S. Islam, "Multi-objective marketing planning inventory model: a geometric programming approach," *Applied Mathematics and Computation*, vol. 205, no. 1, pp. 238–246, 2008.
- [19] H.-M. Wee, C.-C. Lo, and P.-H. Hsu, "A multi-objective joint replenishment inventory model of deteriorated items in a fuzzy environment," *European Journal of Operational Research*, vol. 197, no. 2, pp. 620–631, 2009.
- [20] E. Arkin, D. Joneja, and R. Roundy, "Computational complexity of uncapacitated multi-echelon production planning problems," *Operations Research Letters*, vol. 8, no. 2, pp. 61–66, 1989.
- [21] G. Aiello, G. La Scalia, and M. Enea, "A non dominated ranking multi-objective genetic algorithm and electre method for unequal area facility layout problems," *Expert Systems with Applications*, vol. 40, no. 12, pp. 4812–4819, 2013.
- [22] P. C. Lin, "Portfolio optimization and risk measurement based on non-dominated sorting genetic algorithm," *Journal of Industrial and Management Optimization*, vol. 8, no. 3, pp. 549–564, 2012.
- [23] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.
- [24] W. X. Sheng, Y. M. Liu, X. L. Meng, and T. S. Zhang, "An improved strength Pareto evolutionary algorithm 2 with application to the optimization of distributed generations," *Computers & Mathematics with Applications*, vol. 64, no. 5, pp. 944–955, 2012.
- [25] L. V. Santana-Quintero and C. A. C. Coello, "An algorithm based on differential evolution for multi-objective problems," *International Journal of Computational Intelligence Research*, vol. 1, no. 2, pp. 151–169, 2005.
- [26] B. Qian, L. Wang, D.-X. Huang, and X. Wang, "Scheduling multi-objective job shops using a memetic algorithm based on differential evolution," *International Journal of Advanced Manufacturing Technology*, vol. 35, no. 9-10, pp. 1014–1027, 2008.
- [27] W. Qian and A. Li, "Adaptive differential evolution algorithm for multi-objective optimization problems," *Applied Mathematics and Computation*, vol. 201, no. 1-2, pp. 431–440, 2008.
- [28] L. Wang, J. He, D. S. Wu, and Y.-R. Zeng, "A novel differential evolution algorithm for joint replenishment problem under interdependence and its application," *International Journal of Production Economics*, vol. 135, no. 1, pp. 190–198, 2012.
- [29] L. Wang, J. He, and Y.-R. Zeng, "A differential evolution algorithm for joint replenishment problem using direct grouping and its application," *Expert Systems*, vol. 29, no. 5, pp. 429–441, 2012.
- [30] L. Wang, Q. L. Fu, C. G. Lee, and Y. R. Zeng, "Model and algorithm of fuzzy joint replenishment problem under credibility measure on fuzzy goal," *Knowledge-Based Systems*, vol. 39, pp. 57–66, 2013.
- [31] A. Eynan and D. H. Kropp, "Periodic review and joint replenishment in stochastic demand environments," *IIE Transactions*, vol. 30, no. 11, pp. 1025–1033, 1998.
- [32] A. Eynan and D. H. Kropp, "Effective and simple EOQ-like solutions for stochastic demand periodic review systems," *European Journal of Operational Research*, vol. 180, no. 3, pp. 1135–1143, 2007.
- [33] L. Wang, Q.-L. Fu, and Y.-R. Zeng, "Continuous review inventory models with a mixture of backorders and lost sales under fuzzy demand and different decision situations," *Expert Systems with Applications*, vol. 39, no. 4, pp. 4181–4189, 2012.
- [34] L. G. Cui, L. Wang, and J. Deng, "RFID technology investment evaluation model for the stochastic joint replenishment and delivery problem," *Expert Systems with Applications*, 2013.
- [35] O. Hrstka and A. Kučerová, "Improvements of real coded genetic algorithms based on differential operators preventing premature convergence," *Advances in Engineering Software*, vol. 35, no. 3-4, pp. 237–246, 2004.
- [36] D. He, F. Wang, and Z. Mao, "A hybrid genetic algorithm approach based on differential evolution for economic dispatch with valve-point effect," *International Journal of Electrical Power and Energy Systems*, vol. 30, no. 1, pp. 31–38, 2008.
- [37] W.-Y. Lin, "A GA-DE hybrid evolutionary algorithm for path synthesis of four-bar linkage," *Mechanism and Machine Theory*, vol. 45, no. 8, pp. 1096–1107, 2010.
- [38] T. Robič and B. Filipič, "DEMO: differential evolution for multiobjective optimization," *Lecture Notes in Computer Science*, pp. 520–533.
- [39] A. I. Esparcia-Alcázar, J. J. Merelo, A. Martínez-García, P. García-Sánchez, E. Alfaro-Cid, and K. Sharman, "Comparing single and multiobjective evolutionary approaches to the inventory and transportation problem," CoRR, <http://arxiv.org/abs/0909.3384>.
- [40] A. I. Esparcia-Alcázar, E. Alfaro-Cid, P. García-Sánchez, A. Martínez-García, J. J. Merelo, and K. Sharman, "An evolutionary approach to integrated inventory and routing management in a real world case," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '10)*, pp. 1–7, Barcelona, Spain, July 2010.