International Journal of Multimedia and Ubiquitous Engineering
Vol.10, No.2 (2016), pp.1-8
http://dx.doi.org/10.14257/ijmue.2016.11.2.01

Joint Replenishment and Delivery Problem with Resource Constraint for Deteriorating Item

Chengyan Li, Jun Gao and Chuang Wang

School of Computer Science and Technology
Harbin University of Science and Technology, Harbin 150080, China
chengyan@hrbust.edu.cn

Abstract

The joint replenishment and delivery model with deterministic resource restriction for deteriorating item is developed. The model is formulated as cost minimization problem, including the ordering cost, the inventory holding cost, the transportation cost, the customer waiting cost and the deterioration cost. Differential evolution (DE) algorithm is proposed to solve the model. Numerical illustrations of the model and algorithm are presented and the sensitivity analysis with respect to deterioration rate of item is performed. The comparison between DE and genetic algorithm (GA) for solving the model are also made.

Keywords*: Supply chain management Joint replenishment and delivery problem Resource constraint deteriorating item Differential evolution*

1. Introduction

For the past few decades, the supply chain management (SCM) has received much attention from the business community. Many companies have realized that significant cost saving can be achieved by integrating inventory control and delivery policies throughout their supply chains, that is, the joint replenishment and delivery scheduling (JRD) policy. In JRD, cost can be saved when replenishment of several items are coordinated in a multi-item inventory system. Replenishment of a group of item takes place after a fixed interval of time, called the basic cycle time. Time between two consecutive orders of an item is assumed to be an integer multiple of the basic cycle time. The objective of the JRD is to minimize the total costs incurred per unit time.

Moon *et al*. [1] developed joint replenishment and consolidated freight delivery policies for a third party warehouse with deterministic demand rates in a supply chain. Cha *et al*. [2] dealt with the JRD model of the one-warehouse n-retailer system and suggested a flexible policy for a warehouse. Wang *et al*. [3] studied a joint replenishment and delivery scheduling in which a central warehouse serves n-retailers in the presence of vague operational conditions such as ordering cost and inventory holding cost, and the membership function is approximated using piecewise linear functions based on alpha level sets.

The deteriorating items are subject to a continuous loss in their masses or utility throughout their lifetime due to decay, damage, spoilage, and penalty of other reasons, and most of the physical goods undergo decay or deterioration over time [4]. Considering to this fact, controlling and maintaining the inventory of deteriorating items becomes a challenging problem for decision makers. Chaman developed two-warehouse supply chain model with power form stock-dependent demand under the assumption that the deterioration rate per unit items are different due to different preservation environments [5]. XU *et al*. considered production-inventory models for a deteriorating item in a single vendor-buyer system with constant production and demand rate, and developed Ant Colony algorithm to solve the problem [6]. Debasis *et al*. considered two-warehouse

ISSN: 1975-0080 IJMUE

inventory model for a deteriorating item with time-varying demand and fully backlogged shortages, and using GA with vary population size to solve the model [7].

The differential evolution (DE) algorithm is one of the latest evolutionary optimization methods proposed by Storn and Price [8] for complex continuous non-linear functions. DE is a stochastic population-based optimization method which uses mutation, crossover, and selection operators at each generation to move its population toward the global optimum. Over the past ten years, many research indicated that the DE algorithm can solve the problem more effectively. Kazemipoor *et al*. [9] considered the multiskilled project portfolio scheduling problem and presented an efficient metaheuristic algorithm based on DE, the comparison between the results of DE and Tabu search confirms the effectiveness of the DE algorithm. Wang *et al*. developed an approach based on DE to find a close to optimum for the basic JRP [10]. Das and Suganthan [11] surveyed the state-of-the-art of the differential evolution and its application.

The aim of this paper is to develop a practical constraint JRD model with deteriorating item based on the early work. Secondly, for solving the problem, a DE algorithm is applied. Thirdly, to illustrate the effectiveness and efficiency of the algorithms, extensive computational experiments are performed.

The rest of this paper is organized as follows. Section 2 introduces the mathematical model of constraint JRD with deteriorating item. Section 3 develops the DE algorithm and presents the procedure to solve the problem. Section 4 illustrates the procedure of proposed algorithms with a numerical example. Section 5 summarize the conclusions of the present work and provide directions for future research.

2. Mathematical Model of Joint Replenishment and Delivery Problem with Deteriorating Item

The constraint joint replenishment and delivery problem with deteriorating item (which is abbreviated as Deter-CJRD for the rest of the paper) is the multi-item inventory problem of coordinating the replenishment and delivery of a group of deteriorating items that may be jointly ordered from a single supplier under resource restrictions.

Some common assumptions usually made for the constrained joint replenishment problem:

- The demand rate of each item is deterministic and constant.
- The unit holding cost of each item is known and constant.
- The major ordering cost incurred for an order is known and constant.
- The minor ordering cost incurred for a specific ordered item is known and constant.
- The outbound transportations cost incurred for a specific delivered item is known and constant.
- The customer waiting cost is known and constant.
- No quantity discount.
- Stock replenishment is complete when it occurs.
- The budget constraint on the amount of an order is known and constant.
- The deterioration rate of item follows exponential distribution.

The following notation is defined:

i index of item, i=1,2,...,n
D_i demand rate of item i
S^W major ordering cost
s_i^W minor ordering cost when item i is included in a group replenishment
h_i^W inventory holding cost of item i, per unit per unit time
s_i^C outbound transportation cost for item i
w_i^C customer waiting cost for item i per unit, per unit time
T basic cycle time (decision variable)

k_i integer number that decides the replenishment schedule of item i (decision variable)

K $n\times1$ vector that consists of k_i, i=1,2,…,n

f_i integer number that determines the outbound delivery schedule of item i (decision variable)

F $n\times1$ vector that consists of f_i, i=1,2,…,n

TC total annual cost for all items, objective function that is to be minimized

b_i unit cost of item i

B limit on capital that can be invested

r_i unit deterioration cost of item i

θ_i deterioration rate of item i.

The initial inventory level is I_0 at time 0. From t=0 to T, the inventory level reduces, owing to both demand and deterioration, until it reaches zero level. At this time, shortage is accumulated and backlogged. At the end of the cycle, the inventory replenishes and again raises the level to I_0, as shown in Figure 1.

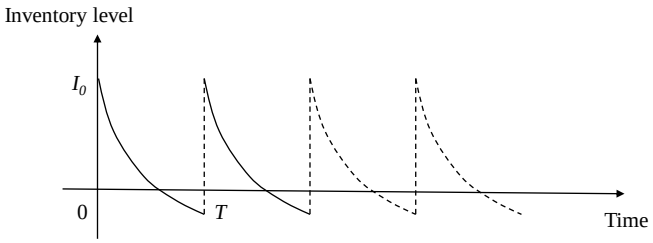

Figure 1. Graphical Representation of Inventory Level

The inventory level at time t is governed by the Eq. (1).

$$\frac{dI_i(t)}{dt} + \theta_i I_i(t) = -D_i, I_i(T) = 0 \tag{1}$$

Then

$$I_i(t) = \frac{D_i}{\theta_i}(e^{\theta_i(T-t)} - 1) \tag{2}$$

The inventory holding cost and deterioration cost of item i in basic replenishment cycle is

$$h_i^W \int_0^T I_i(t)dt = h_i^W \int_0^T \frac{D_i}{\theta_i}(e^{\theta_i(T-t)} - 1)dt = \frac{1}{2}(\frac{D_i h_i^W k_i(f_i - 1)T}{f_i} + D_i\theta_i r_i k_i T) \tag{3}$$

The model for constraint joint replenishment and delivery problem with deteriorating item is given by

(Deter-CJRD)

$$TC(T,K,F) = \frac{S^W + \sum_{i=1}^{n}\frac{s_i^W}{k_i}}{T} + \sum_{i=1}^{n}\frac{D_i h_i^W k_i(f_i-1)T}{2f_i} + \sum_{i=1}^{n}\frac{f_i s_i^C}{k_i T} + \sum_{i=1}^{n}\frac{k_i T D_i w_i^C}{2f_i} + \sum_{i=1}^{n} D_i\theta_i r_i k_i T \tag{4}$$

Subject to

$$\sum_{i=1}^{n} D_i k_i T b_i \le B \tag{5}$$

$$k_i \in Z^+, f_i \in Z^+, T \in R^+, i = 1,2,\ldots n \tag{6}$$

Eq. (4) is the total cost to be minimized, that is the sum of the ordering cost (major and minor), the inventory holding cost, the outbound transportation cost, the customer waiting cost and the deterioration cost. The resource constraint is the limit on capital that can be

invested, as given by Eq. (5). Eq. (6) indicates that k_i's and f_i's are positive integer number and T is positive real number.

To solve the unconstrained JRD, Moon *et al*. used Eq. (7) as the optimal T in their algorithm.

$$T^* = \left\lceil \left. 2(S^W + \sum_{i=1}^{n} \frac{s_i^W + f_i s_i^C}{k_i}) \middle/ \sum_{i=1}^{n} k_i D_i (h_i^W + \frac{w_i^C - h_i^W}{f_i}) \right\rceil^{1/2} \quad (7)$$

The optimality condition of k_i is:

$$k_i(k_i - 1) \le \frac{2(s_i^W + f_i s_i^C)}{T^2 D_i (h_i^W + \frac{w_i^C - h_i^W}{f_i})} \le k_i(k_i + 1) \quad (8)$$

Similarly, the optimality condition of f_i is :

$$f_i(f_i - 1) \le \frac{k_i^2 T^2 D_i (w_i^C - h_i^W)}{2 s_i^C} \le f_i(f_i + 1) \quad (9)$$

3. Differential Evolution Algorithm

In order to solve the constraint joint replenishment and delivery problem with deteriorating item, in this section, we present a differential evolution (DE) algorithm approach for the Deter-CJRD model. DE algorithm, differing from conventional evolutionary optimization methods, such as genetic algorithm (GA), relies on the mutation operations as the main operator. The DE algorithm introduces a novel mutation operation which is simple and effective. The mutation operation is based on the differences of randomly sampled pairs of solution in the population. Furthermore, the fitness of an offspring is one-to-one competed with that of the corresponding parent in DE algorithm.

3.1. Representation and Initialization

The DE algorithm for the Deter-CJRD is to find an optimal schedule which can get a close to minimum total cost. The appropriate representation of a solution plays an important role in the development of DE algorithm. In the Deter-CJRD model, the basic cycle T and n integer k_i's and f_i's have to be decided for solving the problem. In the DE, k_i's and f_i's are searched through the operations of DE and the basic cycle T is determined through the optimality condition of T. For the given k_i's and f_i's , the optimal T^* can be easily obtained by the Eq. (7). So, in our study, we use $2n$ random number representation for n k_i's and n f_i's, because it is very easy to decode our chromosome to a feasible solution. As shown in Figure 2, the decoding individual contains two parts. One is for the replenishment schedule and the other is for the outbound schedule of each item.

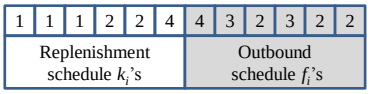

Figure 2. Decoding Individual

The initial population is created by assigning random integer values of the decision variable. Each individual is generated by Eq. (10).

$$x_{ij} = rand(), i = 1, 2, \ldots, POPSIZE, \; j = 1, 2, \ldots, D \quad (10)$$

Where *POPSIZE* is the number of individuals; D is the dimension of each individuals; *rand*() is a function which generates a uniform distribution rand number

in range $[k_i^{LB},k_i^{UB}]$ or $[f_i^{LB},f_i^{UB}]$. k_i^{LB} and k_i^{UB} are the lower and upper bound of k_i, respectively, and can be defined from the following equations.

$$k_i^{LB} = 1, k_i^{UB} = \left\lceil \frac{\sqrt{2(S^W + s_i^W)/D_i h_i^W}}{T_{\min}} \right\rceil \tag{11}$$

f_i^{LB} and f_i^{UB} is the lower and upper bound of f_i, respectively, and can be defined from the following equations.

$$f_i^{LB} = 1,\ f_i^{UB}(f_i^{UB} - 1) \le \frac{(k_i^{UB})^2 T_{\max}^2 D_i (w_i^C - h_i^W)}{2 s_i^C} \le f_i^{UB}(f_i^{UB} + 1) \tag{12}$$

$T_{\max}$ and $T_{\min}$ are defined as follows

$$T_{\max} = \left\lceil 2(S^W + \sum_{i=1}^{n} s_i^W) \Big/ \sum_{i=1}^{n} D_i h_i^W \right\rceil^{1/2}$$
$$T_{\min} = \min(2 s_i / D_i h_i^W) \tag{13}$$

For a given particular set of k_i's and f_i's, eqs. (4) and (5) can be written as

$$TC(T) = \frac{C_1}{T} + C_2 T\ , \tag{14}$$

Subject to

$$C_3 T \le B\ . \tag{15}$$

Where

$C_1 = S^W + \sum_{i=1}^{n} \frac{s_i^W}{k_i} + \sum_{i=1}^{n} \frac{f_i s_i^C}{k_i}$, $C_2 = \sum_{i=1}^{n} \frac{(f_i - 1) k_i D_i h_i^W}{2 f_i} + \sum_{i=1}^{n} \frac{k_i D_i w_i^C}{2 f_i} + \sum_{i=1}^{n} D_i \theta_i r_i k_i$ and

$C_3 = \sum_{i=1}^{n} D_i k_i b_i$ are constants.

We take the first order derivative with respect to T, and set it to zero. Then we obtain $T^0 = \sqrt{C_1 / C_2}$. And for a constraint, we obtain $T_1 = B / C_3$. According to Moon *et al*. [1], the optimal basic cycle time T follows $T^* = \min(T^0, T^1)$.

3.2. Mutation

For each target individual $X_i=\{x_{i1},x_{i2},\ldots,x_{iD}\}$, $i=1,2,\ldots,POPSIZE$, a mutant new individual V_i is generated according to

$$V_i = X_{r1} + F(X_{r2} - X_{r3}), r_1 \ne r_2 \ne r_3 \ne i \tag{16}$$

With randomly chosen integer indexes; $F \in [0,2]$ is a real number called mutation factor used to control the amplification of the differential variation.

3.3. Crossover

To complement the differential mutation search strategy, DE employs uniform crossover, which also known as binomial method, to enhance the potential diversity of the population. The crossover operator implements a discrete recombination of the trial individual V_i and the parent individual X_i to produce the offspring X_i^{new}.

The trial vector will be found using the following rules:

$$x_{ij}^{new} = \begin{cases} v_{ij} & \text{if } rand_j(0,1) \le CR \text{ or } j = rnb(i) \\ x_{ij} & \text{otherwise } j = 1,2,\ldots,D \end{cases} \tag{17}$$

Where D is the dimension of X_i; x_{ij} refers to the jth element of the individual; v_{ij} is similarly defined; $rand_j(0,1)$ is the jth evaluation of a uniform random number generator between [0,1]; $rnb(i)$ is a randomly chosen integer in the set $\{1,2,\ldots,D\}$ which ensures the trail vector gets at least one parameter from the mutated vector; $CR \in [0,1]$ is a crossover constant.

3.4. Selection

The selection operator is to determine whether the target (parent) or the new vector (offspring) survives to the next generation. If a new vector, $X_i^{\text{new}}, i=1,2,\ldots,POPSIZE$, has a smaller evaluation function value (total cost) than its garget vector, X_i , it is copied to the next generation; otherwise, it is the target vector that passes to the next generation. The selection process can be expressed as:

$$X_i = \begin{cases} X_i^{new} & \text{if } fitness(X_i^{new}) < fitness(X_i) \\ X_i & \text{otherwise} \end{cases} \quad (18)$$

Thus the next generation of population either gets better in terms of the fitness function or remains constants.

3.5. Stop Criterion

The termination condition is to stop if no improvement of fitness function is made in 50 generations.

4. Numerical Example

A numerical example will be employed to illustrate the proposed DE algorithm, we use the numerical example of Moon *et. al.*, [1] and Wang *et. al.*, [3]. The data for this example are given in Table 1. We also assume S^W=200 and B=25000. Notice that, there are resource restriction, b_i and B, in our example.

Table 1. Parameter Values for the Example

Item i	1	2	3	4	5	6
D_i	10000	5000	3000	1000	600	200
s_i^W	45	46	47	44	45	47
h_i^W	1	1	1	1	1	1
s_i^C	5	5	5	5	5	5
w_i^C	1.5	1.5	1.5	1.5	1.5	1.5
b_i	6.25	6.25	6.25	6.25	6.25	6.25
r_i	1	1	1	1	1	1

The DE algorithm is compared to the GA algorithm for the same numerical example. The results are reported in Table 2. It shows in Table 2 that DE obtains better solution than GA does.

Table 2. Computational Results under Different θ

θ	DE				GA			
	k_i	f_i	T^*	TC	k_i	f_i	T^*	TC
0.00	1,1,1,2,2,4	4,3,2 ,3,2,2	0.1881	4831.71	1,1,1,2,2,4	4,3,2,3,2,4	0.1818	4836.37
0.02	1,1,1,2,2,4	4,3,2 ,3,2,2	0.1881	4911.71	1,1,1,2,2,5	4,3,2 ,3,2,8	0.1802	4936.51
0.04	1,1,1,2,2,4	4,3,2 ,3,2,2	0.1881	4991.71	1,1,1,2,2,3	4,3,2 ,3,2,1	0.1830	4998.02
0.06	1,1,1,2,2,4	4,3,2 ,3,2,2	0.1881	5071.15	1,1,1,2,2,3	4,3,2 ,3,2,4	0.1800	5076.10
0.08	1,1,1,2,2,4	4,3,2 ,3,2,2	0.1881	5149.38	1,1,1,2,2,4	4,3,2 ,3,2,5	0.1775	5159.95
0.10	1,1,1,2,2,4	4,3,2 ,3,2,2	0.1881	5226.43	1,1,1,2,2,5	4,3,2 ,3,2,9	0.1743	5258.59

We compare the performance of two algorithms, DE and GA, for 1600 randomly generated Deter-CJRDs. The parameter values are all generated from a uniform distribution, D_i~U[500,5000], s_i^W~U[30,50], s_i^C~U[0.1s_i^W,0.3s_i^W], h_i^W~U[0.2,3.0] and w_i^C~U[1.2h_i^W,2.0h_i^W] respectively. Every b_i is considered as 6.25, i=1,2,…,n. Moreover, θ=0.02 is constant. Four different values of the number of items, n=10, 20, 30 and 50, and four different values of the major ordering cost, S^W=100, 200, 300 and 400 are considered. This results in 16 combinations of n and S^W, and for each combination, 100 problems with random parameter values are generated and solved. The termination condition is to stop if no improvement is made in 50 generations.

A summary of computational results is shown in Table 3.

Table 3. Comparison of DE and GA

n	S^W	Best solution problems		DE better than GA (%)		
		DE	GA	Num	Max	Avg.
10	100	100	100	80	5.2648	0.7359
	200	100	100	83	1.9952	0.5502
	300	100	100	86	3.3217	0.7789
	400	100	100	90	4.4830	0.7540
20	100	100	100	86	4.4388	1.3800
	200	100	100	80	4.6668	0.6221
	300	100	100	76	3.0064	0.6071
	400	100	100	71	2.1448	0.6581
30	100	100	100	82	3.2284	1.5014
	200	100	100	83	4.6256	1.2560
	300	100	100	81	4.4656	0.7110
	400	100	100	76	2.0580	0.6614
50	100	100	100	93	1.8431	0.9628
	200	100	100	98	3.9306	2.5099
	300	100	100	100	5.4950	3.3884
	400	100	100	99	6.9276	3.9919

As show in Table 3, both algorithms can solve the 1600 random generated problems. The DE is superior to GA algorithm, especially for n=50.

5. Conclusion

This paper focuses on the constrained joint replenishment and delivery problem with deteriorating item, and developing efficient algorithms to solving it. We introduced the DE algorithm to handle the problem. Using comprehensive computational experiments, we have shown the performance of the DE. Furthermore, DE is more suitable for solving

Deter-CJRD than the GA algorithm. Future research could be the incorporation of uncertainty issue in constrained JRD, for example demand and budget.

Acknowledgments

This work was supported by grant No.12541142 from the Research Program of the Education Department of Heilongjiang Province, China.

References

[1] K. Moon, B. C. Cha and C. U. Lee, "The joint replenishment and freight consolidation of a warehouse in a supply chain", Int. J. Prod Econ. vol. 1, no. 133, **(2011)**.
[2] B. C. Cha, I. K. Moon and J. H. Park, "The joint replenishment and delivery scheduling of the one-warehouse, n-retailer system", Transport Res E-Log., vol. 5, no. 44, **(2008)**.
[3] L. Wang, D. CX, C. G. Lee, Q. L. F and Y. R. Z, Model and algorithm for fuzzy joint replenishment and delivery scheduling with explicit membership function. Int. Journal of Adv Manuf. vol. 66, **(2013)**, pp. 9-12.
[4] V. K. Mishra, L. S. Singh and R. Kumar, "An inventory model for deteriorating items with time-dependent demand and time-varying holding cost under partial backlogging", Journal of Industrial Engineering Int., vol. 4, no. 9, **(2013)**.
[5] C. Singh and S. R. Singh, "Optimal ordering policy for deteriorating items with power-form stock dependent demand under two-warehouse storage facility", OP search, vol. 2, no. 50, **(2013)**.
[6] H. Xu, C. Y. Li, and L. B. Zhou, "Research on order problem for perishable products based on ant colony algorithms", J. Harbin University Of Science and Technology, vol. 3, no. 15 **(2010)**.
[7] D. Debasis, B. K. Mohuby, R Arindam and K. Samarjit, "Two-warehouse production inventory model for a deteriorating item with time-varying demand and shortages: a genetic algorithm with varying population size approach", Optimal Engineering, vol. 15, **(2014)**.
[8] R. Storn and K. K. Price, "Differential evolution- a simple and efficient heuristic for global optimization over continuous spaces", J. Glob Optim., vol. 4, no. 11, **(1997)**.
[9] H. Kazemipoor, R. T. Moghaddam, P. S. Shahrezaei and A. Azaron, "A differential evolution algorithm to solve multi-skilled project portfolio scheduling problems", Int. Journal of Advance Manufacture Technology, vol. 5-8, no. 64, **(2013)**.
[10] L. Wang, J. He and Y. R. Zeng, "A differential evolution algorithm for joint replenishment problem using direct grouping and its application", Expert Syst., vol. 5, no. 29, **(2012)**.
[11] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art", IEEE T Evolution Computer, vol. 15, **(2011)**.

