Supply network Optimization Based on Supper Efficiency Non-dominated Sorting Genetic Algorithm

Yadong Wang, Quan Shi, Zhifeng You, Lei Li, Fang Zhang

1Department of Equipment Command and Management, Army Engineering University of PLA, Shijiazhuang 050003
2Science and Technology Innovation Research Center, No.32178 troops of PLA, Beijing 100012
Quan Shi, E-mail: xwj0003@163.com

Abstract: In order to optimize the supply network, a multi-objective constraint optimization model is established with the constraints of demand satisfaction, lead time and inventory capacity, aiming at minimizing total supply time and the supply cost. By using the proposed upper efficiency non-dominated sorting genetic algorithm, the set of non-dominant solutions are obtained. During optimization process, the upper efficiency data envelopment analysis (SE-DEA) is used to calculate the supper efficiency of the non-dominant solutions. In this way, on the one hand, the algorithm is guided to converge to the optimal efficiency individuals. On the other hand, the non-dominant solutions are sorted to select the optimal one. The example shows that 16 non-dominant solutions are obtained by using proposed algorithm, and the solution with supper efficiency of 0.9538 is determined as the optimal scheme.

1. Introduction
The supply network optimization should take into account the factors of time and cost comprehensively. Thus, the problem is a multi-objective optimization problem and is difficult to model and solve. At present, many scholars have carried out the relevant research on the supply network optimization. Fazli et.al. minimized the total supply cost and transit time between the facilities while maximizing the supply reliability to establish an emergency supply three-objective optimization model [1]. Zhang et.al. developed a multi-objective three-stage stochastic programming model to minimize the transportation time, the transportation cost and the non-satisfaction degree, and the multi-objective model is processed by using the substitute single objective model based on the membership fuzzy variable [2]. Mohammadi et.al. established a multi-objective random planning model to maximize the total expected demand coverage, the total expected cost, and the satisfaction rate among the nodes. A multi-objective particle swarm optimization algorithm was used to solve the model [3]. Su et.al constructed a double-objective multiplication-constrained integer linear programming model based on total response time and emergency resource cost, and used the differential evolution algorithm to search for optimal solution of the model [4].

It can be seen that most of the current research uses multi-objective optimization model. However, the multi-objective optimization solution is difficult to be solved, and cannot obtain the unique optimal solution. In order to solve the problem of supply network optimization, a multi-objective optimization model of supply network is constructed to minimize the supply time and cost. The super efficiency non-dominant sorting genetic algorithm is used to solve the algorithm.
2. Supply network optimization model

2.1 Problem description
A three-level supply network consists of supply centers, distribution centers and customers. The supply network optimization is based on the demands and the information of the nodes to determine the best product allocation scheme, thereby ensuring that the demands are met with the shortest time and the minimum cost.

The model is built on the following assumptions: (1) the transportation cost and transportation time between nodes are known and fixed, and the transportation capacity between nodes is sufficient; (2) the open cost, inventory cost and maximum capacity of distribution centers are known and fixed; (3) the demands for customers and the shortage costs are known.

2.2 Parameter declaration
- \( I \) : the index of supply centers, \( i = 1, 2, \cdots \); 
- \( J \) : the index of distribution center, \( j = 1, 2, \cdots \); 
- \( K \) : the index of customers, \( k = 1, 2, \cdots \); 
- \( U_j \) : the maximum capacity of distribution center \( j \); 
- \( d_k \) : demand of customer \( k \); 
- \( C_j^{\text{open}} \) : open cost of distribution center \( j \); 
- \( C_j^{\text{invent}} \) : inventory cost of distribution center \( j \); 
- \( C_j^{\text{short}} \) : shortage cost of customer \( k \); 
- \( T_k^{\text{lead}} \) : the maximum lead time of customer \( k \); 
- \( T_{ij}^{\text{trans}} \) : transport time from supply centers \( i \) to distribution centers \( j \); 
- \( T_{jk}^{\text{trans}} \) : transport time from distribution centers \( j \) to customers \( k \); 
- \( C_{ij}^{\text{trans}} \) : transport cost from supply centers \( i \) to distribution centers \( j \); 
- \( C_{jk}^{\text{trans}} \) : transport cost from distribution centers \( j \) to customers \( k \).

Decision variables:
- \( x_{ij} \) : quantity of production supplied from supply centers \( i \) to distribution centers \( j \); 
- \( x_{jk} \) : quantity of production supplied from distribution centers \( j \) to customers \( k \); 
- \( y_j \) : binary variables, when distribution centers \( j \) is operated, \( y_j = 1 \), otherwise, \( y_j = 0 \).

2.3 Modeling
Objective function 1: maximizing the total supply time:

\[
\min \sum_{i \in I} \sum_{j \in J} T_{ij}^{\text{trans}} \cdot x_{ij} + \sum_{j \in J} \sum_{k \in K} T_{jk}^{\text{trans}} \cdot x_{jk}
\]  

Objective Function 2: minimizing the total supply cost of product.

\[
\min C = C^{\text{open}} + C^{\text{trans}} + C^{\text{invent}} + C^{\text{short}}
\]

where, \( C^{\text{open}} \) indicates the open cost of the distribution center, \( C^{\text{trans}} \) indicates the transport cost, \( C^{\text{invent}} \) indicates inventory cost, and \( C^{\text{short}} \) indicates shortage cost, which are formulate as follows:

\[
C^{\text{open}} = \sum_{j \in J} C_j^{\text{open}} \cdot y_j
\]  
\[
C^{\text{trans}} = \sum_{i \in I} \sum_{j \in J} C_{ij}^{\text{trans}} \cdot x_{ij} + \sum_{j \in J} \sum_{k \in K} C_{jk}^{\text{trans}} \cdot x_{jk}
\]  
\[
C^{\text{invent}} = \sum_{j \in J} C_j^{\text{invent}} \cdot (\sum_{i \in I} x_{ij} - \sum_{k \in K} x_{jk})
\]  
\[
C^{\text{short}} = \sum_{k \in K} C_k^{\text{short}} \cdot \left| d_k - \sum_{j \in J} x_{jk} \right|
\]

The following constraints are met:

\[
s.t.
\]
\[ \sum_{i=1}^{\text{all}} x_{ij} \leq y_j \cdot U_j \quad j = 1,2,\ldots,J \tag{7} \]
\[ \sum_{j=1}^{\text{all}} x_{jk} \leq y_j \cdot U_j \quad j = 1,2,\ldots,J \tag{8} \]
\[ \sum_{j=1}^{\text{all}} y_j \cdot U_j \sum_{j=1}^{\text{all}} x_{jk} \leq 1 \quad k = 1,2,\ldots,K \tag{9} \]
\[ \sum_{j=1}^{\text{all}} x_{jk} \leq \sum_{i=1}^{\text{all}} x_{ij} \quad j = 1,2,\ldots,J \tag{10} \]
\[ \max \left\{ T_{ij}^{\text{trans}} \cdot \text{sgn}(x_{ij}) \right\} \mid i \in I, j \in J \] +
\[ \max \left\{ T_{jk}^{\text{trans}} \cdot \text{sgn}(x_{jk}) \right\} \mid j \in J \leq T_k^{\text{lead}} \quad k = 1,2,\ldots,K \tag{11} \]
\[ x_{ij} \in N_+ \quad x_{jk} \in N_+ \quad y_j = \{0,1\} \tag{12} \]

where, the constraint (7) and (8) indicate that the closed distribution centers does not participate in the product supply, and also specifies that the number of products supplied should not exceed the maximum capacity of the distribution centers. Constraint (9) stipulates that the product demand of customers must be met; The constraint (10) specifies that the product demand of customers must be met; The constraint (11) specifies that the lead time for each customers shall not exceed the maximum allowable deadline.

3. Super efficiency non-dominated sorting genetic algorithm

In order to solve the optimal solution of multi-objective optimization model of supply network, an super efficiency non-dominated sorting genetic algorithm (SENSGA) is proposed in this paper. The algorithm adopts the basic framework of genetic algorithm. In each iteration, the non-dominant solutions of the population are stored into archive, and the elite individual is selected from the archive to guide the evolution of the population. Different from traditional genetic algorithm, SENSGA algorithm uses sorting mechanism to select elite individuals. The non-dominant individuals are sorted based on their efficiency values and the individual with the maximum efficiency is selected as the elite individual. The efficiency can reflect the comprehensive performance of the individual in the optimization model, which has practical significance. The efficiency is calculated by using an improved Data Envelopment Analysis (DEA). Each non-dominant individual is a Decision Making Units (DMU).

3.1 Evolution strategy

The evolution strategy is the process of the mutation, crossover, selection. By evolution, the population is constantly improved to obtain a better solution.

Step 1 Mutation

\[ x_i = x_{i1}, x_{i2}, \ldots, x_{ID} \] is the \( i \)-th parent individual, and \( v_i = v_{i1}, v_{i2}, \ldots, v_{ID} \) is the corresponding mutation individual:

\[ v_i = x_i + F \times (x_{\text{elite}} - x_i) + F \times (x_{\text{rand}} - x_i) \tag{13} \]

where, \( x_{\text{elite}} \) is the elite individual selected from archive, \( x_{\text{rand}} \) is the random individual selected form the archive randomly, \( x_i \neq x_{\text{elite}} \neq x_{\text{rand}} \), \( F \) is mutation rate.

Step 2 Crossover

The crossover individual \( u_i \) is produced by parent individual \( x_i \) and the mutation individual \( v_i \) :
where, \( rand_{ij} \) is uniformly distributed random number between \([0,1]\), \( CR \in [0,1] \) is crossover rate.

Step 3 Selection.

Compared the fitness function of the parent and the crossover individual. The non-dominant one is selected as offspring individual.

Step 4 Update the archive

Mix the non-dominant solution of the offspring population to the archive, the non-dominant solution of the mixed population is selected as a new archive.

3.2 Fitness function

The multi-objective optimization model constructed in this paper is a constrained optimization problem. In this paper, the dynamic penalty function method is used to deal with the model constraints [5]:

\[
f(\bar{x}) = O(\bar{x}) + (M_0 - \frac{\text{iter}}{\text{max iter}}) \cdot M_K \cdot P(\bar{x})
\]

where, \( f(\bar{x}) \) is fitness function, \( O(\bar{x}) \) is objective function, \( M_0 \) is the initial penalty factor, \( \text{iter} \) is the current iterations, \( \text{max iter} \) is the maximum iterations, \( M_K \) is penalty coefficient, and \( P(\bar{x}) \) is the constraint violation.

3.3 Metrics

According to the nature of the DEA model, the evaluation metrics is divided into the input metrics and the output metrics, the smaller the input metrics, the higher the efficiency of the corresponding DMU is, and the higher the value of the output metrics, the higher the efficiency of the DMU is.

The input metrics in this paper include:

1. Supply cost: the calculation formula is the objective function (2);
2. Supply time: the calculation formula is the objective function (1).

The output metrics includes:

1. Timeliness metrics: The inverse of the lead time is taken as timeliness metrics, the calculation formula of the delay time is the constraint (11);
2. Fill rate: the calculation formula is the constraint (9);
3. Constraint violation: because the model also involves constraints such as capacity and flow balance, the reciprocal of the overall constraint violation of the model is used as the metrics.

3.4 Super efficiency sorting

The DEA model is used to calculate the individual efficiency [6]:

\[
\max E_{dd} = \frac{\sum_{r=1}^{s} \mu_{rd} \cdot y_{rd}}{\sum_{i=1}^{m} \nu_{id} \cdot x_{id}}
\]

s.t. \( E_j = \frac{\sum_{r=1}^{s} \mu_{rj} \cdot y_{rj}}{\sum_{i=1}^{m} \nu_{ij} \cdot x_{ij}} \leq 1 \quad j = 1, 2, \cdots, n \) (17)

\[
\mu_{ij} \geq \varepsilon \quad \nu_{ij} \geq \varepsilon \quad r = 1, 2, \cdots, s \quad i = 1, 2, \cdots, m
\]

(18)

where, \( y_{rj} \) indicate the \( r \) th output metrics of the \( j \) th DMU, \( \mu_{ij} \) is the weight of \( y_{rj} \); \( x_{ij} \) indicate the
ith input metrics of the jth DMU, \( \nu_{ij} \) is the weight of \( x_{ij} \). \( E_{dd} \) is the self efficiency of \( DMU_d \).

Constraint (17) specifies that the efficiency value of all DMU should be between 0 and 1. Constraint (18) stipulates that the weight of ownership shall be greater than 1, \( \varepsilon \) is a non-Archimedes number smaller than any positive number.

According to the self-evaluation efficiency, the traditional DEA model can only distinguish the effective units (the efficiency value is equal to 1) and the ineffective unit (the efficiency value is less than 1), and the effective units cannot be sorted [7]. SENSGA algorithm can evaluate and sort all DMUs by calculating the super efficiency of each DMU.

The calculation steps for super efficiency are as follows:

First of all, the dual form of the traditional DEA model is obtained. Then, on the basis of the dual model, the super efficiency DEA model will exclude the evaluated DMU, and its input and output will be replaced by the linear combination of the input and output of all other DMEs. In this way, the efficiency value is no longer limited to between 0 and 1 but is allowed to exceed 1. Therefore, the model of the super-efficiency DEA is as follows:

\[
\begin{align*}
\max & \quad \theta - \varepsilon (\bar{e}^T s^- + \bar{e}^T s^+) \\
\text{s.t.} & \quad \sum_{j=1}^{n} \lambda_j x_{ij} + s^- = \theta \tilde{x}_{ij} \\
& \quad \sum_{j=1}^{n} \lambda_j y_{ij} - s^+ = y_{ij} \\
& \quad \lambda_j \geq 0, j = 1, 2, \cdots, n \\
& \quad s^- \geq 0 \\
& \quad s^+ \geq 0
\end{align*}
\]

(19)

4. Numerical examples

4.1 Example description

A supply network consists of 2 supply centers, potential distribution centers, and 6 customers. The product is shipped to distribution centers from supply centers, and distributed to customers based on their demands. The relevant data for supply centers, distribution centers and customers are shown in tables 1 to 3.

| Distribution center | Units | Inventory cost | Opening cost |
|---------------------|-------|----------------|--------------|
| Supply centers1     | 56/325| 30             | 5500         |
| Supply centers2     | 28/330| 30             | 5500         |
| Customers1          | 28/330| 30             | 5500         |
| Customers2          | 28/330| 30             | 5500         |
| Customers3          | 28/330| 30             | 5500         |
| Customers4          | 28/330| 30             | 5500         |
| Customers5          | 28/330| 30             | 5500         |
| Customers6          | 28/330| 30             | 5500         |
4.2 Optimization results

The total of 16 non-dominant solutions are obtained by the algorithm, and each solution represents a supply scheme. Table 4 shows the values of the input metrics, output metrics, super efficiency and self efficiency of all the DMUs. It can be seen that, there are 9 effective DMUs. Of which self efficiency are all equal to 1. The optimal solution cannot be further selected from these effective DMUs only according to their self efficiency. By calculating and comparing the super efficiency of each DMU, the one with supper efficiency equal to 0.9538 is selected as the optimal solution.

| DMU | Input metrics | Output metrics | Constraint violation | Supper efficiency | Self efficiency |
|-----|---------------|----------------|----------------------|------------------|----------------|
|     | Cost | Time | Timeless | Fill rate1 | Fill rate2 | Fill rate3 | Fill rate4 | Fill rate5 | Fill rate6 |
| 1   | 10755 | 377 | 0.1667 | 1.33 | 0.70 | 0.89 | 1.00 | 0.65 | 0.87 | 0.0058 | 0.9538 | 1 |
| 2   | 18401 | 383 | 0.1667 | 1.42 | 0.70 | 0.83 | 1.00 | 0.63 | 1.00 | 0.0069 | 0.9421 | 2 |
| 3   | 18905 | 378 | 0.1667 | 1.33 | 0.70 | 0.83 | 1.00 | 0.63 | 1.00 | 0.0069 | 0.9530 | 1 |
| 4   | 23898 | 388 | 0.1667 | 1.50 | 0.70 | 0.83 | 1.00 | 0.63 | 1.00 | 0.0072 | 0.9325 | 1 |
| 5   | 25554 | 388 | 0.1667 | 1.33 | 0.75 | 0.89 | 1.00 | 0.63 | 1.00 | 0.0078 | 0.8865 | 1 |
| 6   | 31461 | 388 | 0.1667 | 1.42 | 0.70 | 0.89 | 1.00 | 0.63 | 1.00 | 0.0081 | 0.8889 | 1 |
| 7   | 24811 | 397 | 0.1667 | 1.58 | 0.75 | 0.89 | 1.00 | 0.63 | 0.87 | 0.0072 | 0.9258 | 1 |
| 8   | 10846 | 459 | 0.1667 | 1.75 | 0.70 | 0.94 | 1.00 | 0.63 | 1.00 | 0.0052 | 0.9286 | 1 |
| 9   | 11425 | 457 | 0.1667 | 1.75 | 0.70 | 0.94 | 0.80 | 0.63 | 1.00 | 0.0053 | 0.9215 | 1 |
| 10  | 19276 | 462 | 0.1667 | 1.75 | 0.75 | 0.94 | 1.00 | 0.63 | 1.00 | 0.0062 | 0.8819 | 0.9739 |
| 11  | 20386 | 452 | 0.1667 | 1.75 | 0.65 | 0.94 | 1.00 | 0.63 | 1.00 | 0.0063 | 0.8895 | 0.9924 |
| 12  | 16846 | 459 | 0.1667 | 1.75 | 0.70 | 0.94 | 1.00 | 0.63 | 1.00 | 0.0057 | 0.8904 | 0.9852 |
| 13  | 25230 | 452 | 0.1667 | 1.75 | 0.75 | 0.83 | 1.00 | 0.63 | 1.00 | 0.0069 | 0.8772 | 0.9827 |
| 14  | 25276 | 462 | 0.1667 | 1.75 | 0.75 | 0.94 | 1.00 | 0.63 | 1.00 | 0.0068 | 0.8628 | 0.9624 |
| 15  | 32776 | 462 | 0.1667 | 1.75 | 0.75 | 0.94 | 1.00 | 0.63 | 1.00 | 0.0070 | 0.8371 | 0.9558 |
| 16  | 39298 | 683 | 0.0179 | 1.92 | 0.85 | 0.83 | 1.20 | 0.63 | 1.20 | 0.0023 | 0.5640 | 0.7212 |

5. Conclusion

In this paper, the multi-objective optimization model for supply network optimization is established, and the optimization decision-making problem of the supply network is solved. The main contributions of this paper are as follows: firstly, the proposed multi-objective optimization algorithm and constraint processing method can solve the multi-objective constrained optimization problem of supply network; secondly, the proposed efficiency sorting strategy can further sort the non-dominant solutions, and guide the convergence of the algorithm to the most efficient individuals. Thirdly, by calculating the super efficiency, the problem that the effective DMUs cannot be sorted according to their self efficiency is overcome. In summary, the model and algorithm in this paper provide a framework for supply network optimization of and decision making.

References

[1] Fazli Khalaf M., Khalipourazari S., Mohammadi M. Mixed robust possibilistic flexible chance constraint optimization model for emergency blood supply chain network design [J]. Annals of Operations Research. 2017, 12, 1-31.

[2] Zhang J., Liu H., Yu G., et.al. A three-stage and multi-objective stochastic programming model to improve the sustainable rescue ability by considering secondary disasters in emergency logistics [J]. Computers & Industrial Engineering. 2019, 135, 1145-1154.

[3] Mohammadi R., Ghomi S. M. T. F., Jalal F. Prepositioning emergency earthquake response supplies: A new multi-objective particle swarm optimization algorithm [J]. Applied Mathematical Modelling.
2016, 40(9-10): 5183–5199.

[4] Su Z., Zhang G., Liu Y., et.al. Multiple emergency resource allocation for concurrent incidents in natural disasters [J]. International Journal of Disaster Risk Reduction. 2016, 17: 199–212.

[5] Panda A., Pani S. A Symbiotic Organisms Search algorithm with adaptive penalty function to solve multi-objective constrained optimization problems [J]. Applied Soft Computing. 2016, 46, 344–360.

[6] Coello C A C . Constraint-Handling Techniques used with Evolutionary Algorithms.[C]// Conference Companion on Genetic & Evolutionary Computation Conference: Late Breaking Papers. ACM, Philadelphia, 2012: 2445-2466

[7] Ramanathan R. An Introduction to Data Envelopment Analysis [J]. Journal of the Operational Research Society, 2013, 56(6):751-752.