Two-phase differential evolution for solving emergency response supplies optimisation problem

Qi Cao*

Business School,
Nankai University,
Tianjin 300071, China
and
Department of Logistics Command,
Army Logistics University,
Chongqing 401331, China
Email: roy1976@163.com
*Corresponding author

K.M. Leung

Tandon School of Engineering,
New York University,
Brooklyn, NY 11201, USA
Email: kml441@nyu.edu

Wenhua Hou

Business School,
Nankai University,
Tianjin 300071, China
Email: whhou@nankai.edu.cn

Abstract: A material supply model is constructed for serious disasters in which a large number of supply centres and disaster areas are involved. We introduce a new method referred to as two-phase differential evolution (TPDE) to solve this kind of complex nonlinear programming problem. In constraint handling phase, the goal is to explore the parameter space to identify a feasible solution quickly. In optimum seeking phase, the aim is to gradually improve the quality of current best solution. Different differential evolution schemes and special handling techniques are utilised in the two phases. Extensive numerical optimisation experiments are conducted where TPDE is compared with results obtained from using commercial software and three evolutionary optimisation methods. We determine that TPDE is always able to find a feasible solution with fewer generations and the optimal solution almost always ranks as the best. This work is beneficial to address large-scale nonlinear optimisation problems with constraints. [Received: 28 August 2019; Revised: 1 February 2020; Accepted: 8 March 2020]

Keywords: evolutionary computation; large-scale optimisation; emergency logistics; differential evolution; material supply model; two-phase optimisation.

Copyright © 2021 Inderscience Enterprises Ltd.

Reference to this paper should be made as follows: Cao, Q., Leung, K.M. and Hou, W. (2021) 'Two-phase differential evolution for solving emergency response supplies optimisation problem', *European J. Industrial Engineering*, Vol. 15, No. 1, pp.103–130.

Biographical notes: Qi Cao received his PhD in Computer Science from the Chongqing University, China. He was ever a Visiting Scholar in the Department of Computer Science and Engineering in the Tandon School of Engineering at the New York University, USA. Now he is a Postdoctoral Researcher in the Department of Management Science and Engineering in the Business School at the Nankai University, China. He is also a Professor in the Department of Logistics Command at the Army Logistics University, China. His research interests include computer modelling and simulation, nonlinear constrained optimisation, pattern recognition and machine learning.

K.M. Leung is a Full Professor in the Department of Computer Science and Engineering in the Tandon School of Engineering at the New York University, USA. He received his PhD from the University of Wisconsin at Madison, USA. His research focuses on the global optimisation of nonlinear problems with constraints and its applications to the real-world problems.

Wenhua Hou is a Full Professor in the Department of Management Science and Engineering in the Business School at the Nankai University, China. He received his PhD from the Tianjin University, China. His research interests include outsourcing and crowdsourcing, innovation in health industry, supply chain coordination, game theory and decision science.

1 Introduction

A massive emergency, such as a natural disaster, serious accident, environmental hazard or terrorist attack demands immediate crisis management and deployment of appropriate rescue efforts. As one of the earliest nations with a national emergency plan, the USA issued the Federal Response Plan in 1992 that was replaced by the National Response Plan in 2004 (Kapucu, 2006). China began developing a comprehensive emergency plan in 2001 and issued the National General Response Plan for Public Emergency in 2006 (Yu and Chen, 2007). An adequate supply of material and resources is one of the critical parts of any response plan. To improve rescue response time and reduce losses, material supply centres must be established at strategic locations. A typical emergency logistics problem is to ensure that needed material and resources are delivered from the proper supply centres to various disaster areas in a coordinated manner. Different from traditional commercial logistics, emergency logistics pays more attention to timeliness and demand satisfaction, while economy is set in a secondary place.

Optimisation modelling has become a powerful tool to solve emergency logistics problems since it was first adopted in maritime disaster situations in the 1970s (Caunhye et al., 2012). A bi-objective model for material supplies in devastating earthquakes was built up according to the principles of strong time efficiency and weak economic effectiveness (Zhao, 2012). A multi-objective, multi-mode, multi-commodity, and multi-period stochastic model was proposed to manage the logistics of both commodities and injured people in earthquake responses (Najafi et al., 2013). A post-disaster humanitarian logistic model was improved with an objective function that combines logistic and deprivation costs (Holguín-Veras et al., 2013). A logistical support scheduling model under stochastic travel times for emergency roadway repair work was constructed to minimise the total operating cost with an unanticipated penalty cost by employing network flow techniques (Yan et al., 2014). A multi-objective and multi-dynamic-constraint emergency material vehicle dispatching and routing model was proposed to maximise the transport efficiency and minimise the difference of material urgency degrees among multiple disaster points (Jiang et al., 2017). A dynamic programming model was constructed for a multi-period resource allocation dispatch problem which was extracted to represent the disaster response phase, with special attention paid to the human suffering resulting from the delivery delay (Yu et al., 2018). And a multi-objective programming models combined with data envelopment analysis was proposed to design humanitarian supply chain network which consisted of the optimal emergency response facility locations and allocation scheme of humanitarian supplies (Hong and Jeong, 2019). However there are other techniques that have not been used commonly and can provide a major contribution to many models such as the use of intelligent algorithms and the consideration of covering multiple demand points with multiple distribution centres to ensure the demand satisfaction (Hoyos et al., 2015).

Because of the complexity of rescue and relief environment, most of emergency response models are typical nonlinear programming (NLP) problems with many constraints. Thus, many popular optimisation software tools, such as CPLEX and GUROBI, cannot be used (Stanimirovic et al., 2014). There are several optimisation software tools, such as LINGO, that can handle some constrained NLP problems with limited success depending on the specific application. It may be difficult to identify a feasible solution when the problem is non-convex. With the development of stochastic optimisation methods and intelligent algorithms, more NLP problems, such as disaster relief problem, can be handled by evolutionary algorithms (EAs) (Zheng et al., 2015). Realistic emergency logistics problem is to formulate a reasonable response planning. It should rapidly select the proper material supply centres from many alternatives to deliver the needed emergency material to the various different disaster areas, in order to gain more rescue time and reduce the disaster losses. So it is a typical challenge and requires a powerful robust optimisation method. We ever tried to apply the conventional differential evolution (DE) with constraint handling methods to optimise the material supply model in an emergent disaster. Using the same model data as reported in Zhao (2012), we were able to obtain results that are in agreement with those obtained from a different method. However, the model involves four material supply centres and only two disaster areas. This is far too simplistic and unrealistic. For instance, in the 2008 earthquake in Sichuan province of China, there were ten extreme disaster areas and dozens of severe disaster areas. Demands for material supply came from more than ten disaster areas at the same time. Consequently, the number of optimisation variables and the number of constraints increased quickly. It is sometimes difficult to obtain even a feasible solution for such a large-scale problem with current methods. As a result, we need to find another method that is capable of solving more complex high dimensional emergency problems.

In this paper, a material supply model is constructed for serious disasters in which a large number of supply centres and disaster areas are involved at the same time. We develop a two-phase differential evolution (TPDE) to solve this kind of challenging nonlinear constrained optimisation problem. The major feature of our method is that it consists of two phases. If a feasible solution has not been found, it enters the constraint handling phase where several techniques are used to quickly find a possible feasible solution. When a feasible solution has been found, it enters the optimum seeking phase where the goal is to systematically locate a global optimal solution. Different evolution schemes and special constraint handling techniques are utilised in the two phases. The TPDE is then applied to the material supply model. Results of our findings are compared with those obtained from using these four methods: the commercial software LINGO and three EAs, namely, genetic algorithm (GA), particle swarm optimisation (PSO), and conventional DE. We find that our TPDE performed much better than any of these other methods for a number of datasets with various degrees of complexity. Therefore, we believe that our work is beneficial to address large-scale nonlinear optimisation problems with constraints.

The rest of the paper is organised as follows: Section 2 reviews the relative literature. Section 3 introduces the material supply model. Section 4 presents the TPDE algorithm for solving large-scale constrained nonlinear optimisation problems. In Section 5, extensive numerical optimisation experiments are conducted. Finally, conclusion of the paper and future research directions are presented in Sections 6.

2 State of art

For the literature review, the time period considered in the research ranges from 2010 to 2019. The Web of Science database was consulted and the keywords 'emergency model' and 'EA optimisation' were used in the search process. According to the abstract, only the papers that applied in emergency response modelling and optimised with EAs were taken into account. Then the references in each paper were reviewed and relevant papers from the references were filtered with the same process. In particular, the papers with weak relevance and review papers were removed. Finally, a total of 55 papers were included in the review and the full-text of each paper was retrieved from SpringerLink, ScienceDirect, IEEE Explore, ASCE, and other databases. The time distribution of these papers is shown in Figure 1. In recent years, the number of high-quality papers takes on ascend trend, which shows that the international community continues to pay attention to the applications of EAs for emergency optimisation. And it is becoming the research focus with the increase of global disasters.

A variety of EAs have been found a growing number of applications in relief operations and demonstrated their effectiveness on many challenging emergency problems. The detailed classification of the 55 papers by EAs is shown in Figure 2.

As a traditional EA, GA regards fitness function as criteria for evaluating population individual performance according to the survival of the fittest in nature. The individuals with good performance will be selected as parent individuals with a certain probability to participate in the genetic steps such as mutation, crossover and selection, which leads to generate a new population (Kellenbrink and Helber, 2016). The probability of being selected is controlled by the fitness value. Almost 1/3 papers used GA or improved GA as the optimisation method. For example, GA was joined with a case-based reasoner to form a calibration approach to predict the forest fire spread in environmental emergency modelling (Wendt et al., 2013). Non-dominated sorting GA-II (NSGA-II) was used to solve the problem for determining the locations of temporary storage facilities and planning for the collection and transportation of disaster waste in Istanbul (Onan et al., 2015). And a genetic-based EA (EV) was developed to optimise a stochastic model which determined the stockpile location and capacities of medical supplies for improved disaster preparedness in the event of a hurricane (Paul and MacDonald, 2016). According to the number and time distribution of literature, GA is the mainstream of emergency optimisation field. But it was mainly utilised in the previous years. With the popularisation of other emerging algorithms, the application of GA is gradual decreasing.

Figure 1 Time distribution of the related papers (see online version for colours)

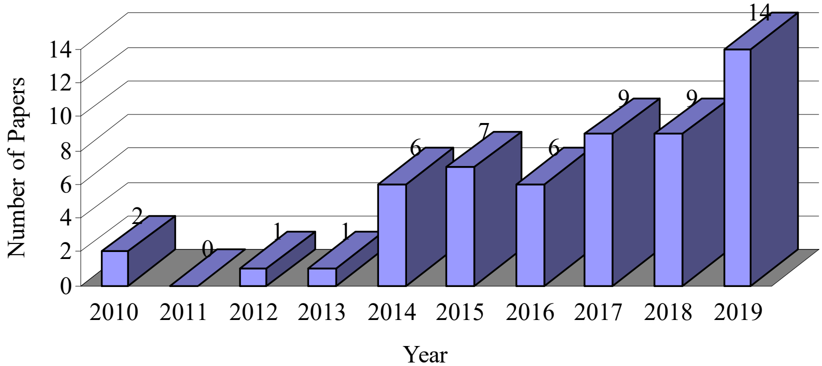
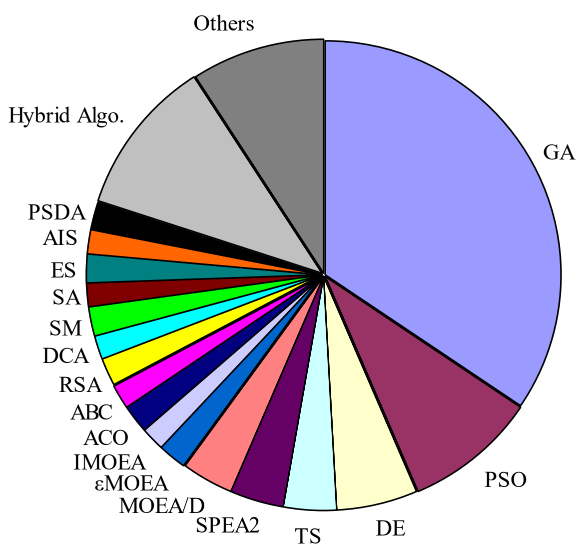


Figure 2 Classification of papers by EAs (2010–2019) (see online version for colours)



PSO grows out of the movement of organisms in a bird flock. It solves a problem by moving particles around in the search-space for the best particle according to simple formulae over the particle's position and velocity (Meziani et al., 2018). So there is no mutation and crossover steps for population individuals. PSO was also widely applied in the emergency optimisation in the past decade. For example, binary PSO and continuous PSO were designed to deal with the multi-objective stochastic programming model for developing an earthquake response plan (Mohammadi et al., 2016). A novel swam intelligence optimisation algorithm that combined PSO with the filled function method was presented to solve the evacuation routing optimisation problem (Zhu et al., 2017). And a novel PSO algorithm was proposed in order to improve the accuracy of traditional clustering approaches with applications in analysing real-time patient attendance data from an accident and emergency department in a local UK hospital (Liu et al., 2019).

As a real-valued parameters optimisation algorithm, DE is well-known for its simplicity, high convergence characteristics and robustness (Price et al., 2005). It operates through similar computational steps as employed by a standard EA. However, unlike the traditional EAs, the DE variants perturb the current-generation population individuals with the scaled differences of distinct individuals. Therefore, no separate probability distribution has to be used for generating the offspring. A new approach to belief rule-based (BRB) parameter learning via extended causal strength logic (BRBcast) was proposed and the detailed BRBcast procedure was presented with the DE algorithm, which provided a promising avenue for constructing accurate and robust disaster emergency and rapid response systems (Sun et al., 2018). A dynamic self-adaptive DE algorithm was proposed to solve the model of multi-period material supply in emergency logistics network (Wang et al., 2019). A new environment change detect operator and a new environment change response strategy are adopted, and an adaptive mutation strategy was used to improve the ability of global exploration and local exploitation.

Other EAs were also utilised in the field of emergency optimisation. A multi-objective tabu search (TS) was proposed for a real-world hospital emergency department layout problem, in which a penalty function was devised to handle infeasible solutions and local search was integrated to optimise the assignment of departments (Zuo et al., 2019). A multi-objective artificial bee colony (ABC) algorithm combined with random swap and random insertion methods for neighbourhood search, two-point crossover operator, and Pareto-based method was presented to solve the evacuation problem in city of Kigali, Rwanda (Niyomubyeyi et al., 2019). An improved simulated annealing (SA) algorithm and hierarchical optimisation method were used to deal with the emergency rush-repair task scheduling of distribution networks in power grid emergencies (Pang et al., 2016). In addition, other methods with emergency application include strength Pareto evolutionary algorithm 2 (SPEA2), multi-objective evolutionary algorithm based on decomposition (MOEA/D), ε-multi-objective evolutionary algorithm (εMOEA), improved multi-objective evolutionary algorithm (IMOEA), ant colony optimisation (ACO), ripple spreading algorithm (RSA), datastream clustering algorithm (DCA), surrogate model (SM), evolution strategy (ES), artificial immune system (AIS), and probabilistic solution discovery algorithm (PSDA). And hybrid algorithms are also the common methods in research, such as GA-PSO, GA-ACO, GA-FA (firefly algorithm) and EA-LS (local search).

Generally speaking, each algorithm has its own somewhat limited scope of application in emergency management. Especially for the complex high-dimensional problems in applications, it is hard to keep the balance between optimisation efficiency and solution quality with the sharp increase of the problem scale. Different methods have been presented in theses algorithms to attempt to deal with this issue, which can be classified into four aspects:

1 breaking away from the premature convergence or local optimal solution in time

2 boosting sufficient diversity of the population to maintain evolutionary ability

3 balancing the global searching of solution space and the local searching for efficient convergence

4 simple or adaptive operations without extensive fine-tuning to adjust the algorithm itself and the specific hyper-parameters.

However, there is still a lack of the comprehensive integrated methods and reasonably robust algorithms for the large-scale constrained NLP problem.

In view of the above aspects, we especially design the ES and propose the TPDE algorithm. And three most widely used algorithms in emergency response modelling, GA, PSO and DE, are chosen as the comparison methods in the later optimisation experiment.

3 Material supply model for serious disasters

3.1 Main features

There are three main features for material supply in emergency, especially in serious disasters. First, the roads between the supply centres and the disaster areas could be damaged. The transportation of emergency materials can only be carried out after the roads are repaired. Therefore, besides the transport time of materials, the repair time of roads should also be considered in the scheduling plan. Secondly, the supply of emergency materials would be large in serious disaster. The transport vehicles owned by each supply centre are limited, and the load of each vehicle is also limited. So the maximum quantity of materials from the supply centre in a single trip may be less than the demand of the disaster area. That is, the vehicles need to travel between the supply centre and the disaster area many times to meet the material demand of the disaster area. Third, as far as the supply of emergency materials is concerned, whether or not the supplies can be delivered to the disaster areas within the required time will directly affect the lives of the people. The time limits should be given priority. Consequently, the material supply model is constructed especially for serious disasters. For the supply centres, there may be a lot of demands come from a large number of disaster areas at the same time. The impacts of road damage and repair are especially considered. The multiple roundtrips derived from the shortage of transport capacity are designed particularly. And the time limits in emergency are regarded as the most important constraints.

3.2 Mathematical model

There is a material supply model with m supply centres ($\boldsymbol{A}_i$, $i = 1, 2, \ldots, m$) and l disaster areas ($\boldsymbol{B}_j$, $j = 1, 2, \ldots, l$). The notations of the model are presented as follows:

Input parameters:

$\boldsymbol{T}_j$ the maximum time limit of material demanded for disaster area $\boldsymbol{B}_j$ in an emergency

$\boldsymbol{r}_{ij}$ the actual repair time for the roads connecting supply centre $\boldsymbol{A}_i$ and disaster area $\boldsymbol{B}_j$, $\boldsymbol{r}_{ij} = \boldsymbol{\beta}_{ij} \times \boldsymbol{\alpha}_{ij}$

$\boldsymbol{\alpha}_{ij}$ the repair time required for the roads between supply centre $\boldsymbol{A}_i$ and disaster area $\boldsymbol{B}_j$ when all of the roads are damaged

$\boldsymbol{\beta}_{ij}$ the probability that the road between supply centre $\boldsymbol{A}_i$ and disaster area $\boldsymbol{B}_j$ is damaged, which can be obtained from the satellite and aerial remote sensing data or actual feedback information on the roads

$\boldsymbol{Z}_{ij}$ a Boolean variable representing whether or not supply centre $\boldsymbol{A}_i$ delivers material to disaster area $\boldsymbol{B}_j$ (1 if true, 0 if false)

$\boldsymbol{t}_{ij}$ the transport time required from supply centre $\boldsymbol{A}_i$ to disaster area $\boldsymbol{B}_j$ when the roads are kept in good repair, which is determined by the length of road, vehicle speed, and traffic flow

$\boldsymbol{c}_{ij}$ the unit transport cost from supply centre $\boldsymbol{A}_i$ to disaster area $\boldsymbol{B}_j$

$\boldsymbol{d}_i$ the startup cost of supply centre $\boldsymbol{A}_i$

$\boldsymbol{Y}_i$ a Boolean variable representing whether or not supply centre $\boldsymbol{A}_i$ is being used

$\boldsymbol{e}_{ij}$ the repair cost for damaged roads from supply centre $\boldsymbol{A}_i$ to disaster area $\boldsymbol{B}_j$

$\boldsymbol{G}_{ij}$ the transport capacity of supply centre $\boldsymbol{A}_i$ to disaster area $\boldsymbol{B}_j$ for a carriage assignment

$\boldsymbol{a}_i$ the total storage capacity of supply centre $\boldsymbol{A}_i$

$\boldsymbol{b}_j$ the material quantity demanded by disaster area $\boldsymbol{B}_j$.

Decision variables:

$\boldsymbol{X}_{ij}$ the material quantity to be delivered from supply centre $\boldsymbol{A}_i$ to disaster area $\boldsymbol{B}_j$ that is alternative material supply planning.

The model is presented as follows:

$$Minimise \begin{cases} f_1\left(\boldsymbol{X}_{ij}\right) = \underset{j}{Max}\left[\underset{i}{Max}\left(\boldsymbol{n}_{ij}\boldsymbol{t}_{ij} + \boldsymbol{r}_{ij}\boldsymbol{Z}_{ij}\right) - \boldsymbol{T}_j\right] \\ f_2\left(\boldsymbol{X}_{ij}\right) = \sum_{i=1}^{m}\sum_{j=1}^{l} c_{ij}\boldsymbol{X}_{ij} + \sum_{i=1}^{m}\boldsymbol{d}_i\boldsymbol{Y}_i + \sum_{i=1}^{m}\sum_{j=1}^{l}\boldsymbol{e}_{ij}\boldsymbol{\beta}_{ij}\boldsymbol{Z}_{ij} \end{cases} \tag{1}$$

where $f_1(\boldsymbol{X}_{ij})$ that represents the transport deviation time, is the objective function for the time efficiency; $f_2(\boldsymbol{X}_{ij})$ represents the total supply cost, the objective function that measures economic effectiveness; $\boldsymbol{n}_{ij}$ is an intrinsic function in $f_1(\boldsymbol{X}_{ij})$, which represents the number of a single trip between the supply centre $\boldsymbol{A}_i$ and disaster area $\boldsymbol{B}_j$ for the transport vehicle and is given by:

$$\boldsymbol{n}_{ij} = \begin{cases} \dfrac{2\boldsymbol{X}_{ij}}{Min\left\{\boldsymbol{X}_{ij}, \boldsymbol{G}_{ij}\right\}} - 1, & \dfrac{\boldsymbol{X}_{ij}}{Min\left\{\boldsymbol{X}_{ij}, \boldsymbol{G}_{ij}\right\}} \in Z \quad \text{and} \quad \boldsymbol{X}_{ij} \neq 0 \\ 2\left\lfloor \dfrac{\boldsymbol{X}_{ij}}{Min\left\{\boldsymbol{X}_{ij}, \boldsymbol{G}_{ij}\right\}} \right\rfloor + 1, & \dfrac{\boldsymbol{X}_{ij}}{Min\left\{\boldsymbol{X}_{ij}, \boldsymbol{G}_{ij}\right\}} \notin Z \quad \text{and} \quad \boldsymbol{X}_{ij} \neq 0 \\ 0, & \boldsymbol{X}_{ij} = 0 \end{cases} \tag{2}$$

The above model must be performed subject to the following inequality constraints:

$$\begin{cases} g_1\left(\boldsymbol{X}_{ij}\right) = \boldsymbol{X}_{ij} - M\boldsymbol{Z}_{ij} \le 0 \\ g_2\left(\boldsymbol{X}_{ij}\right) = \underset{i}{Max}\left(\boldsymbol{n}_{ij}\boldsymbol{t}_{ij} + \boldsymbol{r}_{ij}\boldsymbol{Z}_{ij}\right) - \boldsymbol{T}_j \le 0 \\ g_3\left(\boldsymbol{X}_{ij}\right) = \sum_{j}^{i} \boldsymbol{Z}_{ij} - M\boldsymbol{Y}_i \le 0 \\ g_4\left(\boldsymbol{X}_{ij}\right) = \sum_{j}^{i} \boldsymbol{X}_{ij} - \boldsymbol{a}_i \le 0 \end{cases} \tag{3}$$

where M is a positive infinity integer, and the following equality constraints:

$$h\left(\boldsymbol{X}_{ij}\right) = \sum_{i} \boldsymbol{X}_{ij} - \boldsymbol{b}_j = 0 \Rightarrow g_5\left(\boldsymbol{X}_{ij}\right) = \left|\sum_{i} \boldsymbol{X}_{ij} - \boldsymbol{b}_j\right| - \varepsilon \le 0 \tag{4}$$

where ε is the allowed tolerance (a very small positive value), which is customarily used to change equality constraints into inequality constraints.

In addition, the boundary constraints are:

$$\begin{cases} \boldsymbol{X}_{ij} \in \left[0, \boldsymbol{a}_i\right], m > l \\ \boldsymbol{X}_{ij} \in \left[0, \boldsymbol{b}_j\right], m \le l \end{cases} \tag{5}$$

A weighted sum method was used to change this bi-objective model into a single objective problem by setting weights 2/3 and 1/3, respectively, to combine the two objective functions (Zhao, 2012). However, there is no real physical significance in the weights. It is hard to identify their rationality in the practical application. For example, the measured transport deviation time is typically a couple of hours whereas the total supply cost is in millions of RMB. The widely different scales in function values are undesirable. It cannot guarantee the non-dominatedness property of the solution obtained. Instead, a constraint method can be used where the time efficiency function $f_1(\boldsymbol{X}_{ij})$ is converted into one of the constraints $g_2(\boldsymbol{X}_{ij})$, and the economic effectiveness $f_2(\boldsymbol{X}_{ij})$ is chosen as the only objective function to be minimised subject to all the constraints. In this paper, we assign a priority parameter k_i (i =1, 2, …, 5) for each of the different constraints to specify the degree of importance. Because time efficiency is the most important in emergency disaster relief, we choose k_2 = 100 and k_i = 1 for all others.

4 TPDE algorithm

4.1 Two-phase strategy

With the increase of the problem scale, the numbers of optimisation variables and constraints rise dramatically. For such large-scale problems, it is sometimes hard to find a feasible solution. So we propose a two-phase strategy for EA to deal with this kind of problems. The first phase is to find a feasible solution as quickly as possible. Take the emergency material supply problem as an example, the feasible solution mainly means satisfactory deviation time which is the most important factor. When a feasible solution has been found, the second phase is to systematically locate a global optimal solution in which the total cost is the essential factor. Therefore, the potential of the two-phase strategy is in the rapid manipulation of the evolution of the deviation time, and the best location of the evolution of the total cost. We ever successfully introduced constraints handle methods into conventional DE to deal with small-scale emergency problem. In view of this, we apply the two-phase strategy into the DE. The evolutionary process of DE is technically divided into the constraint handling phase and the optimum seeking phase, and an improved DE algorithm, namely TPDE, is presented. There are different evolution schemes and special constraint handling techniques in the two phases, which are elaborated in the solution algorithm.

4.2 Solution algorithm

The basic steps of the TPDE are illustrated in Figure 3.

Step 1 (Initialisation) The generation number G is set to 0. The initial population of NP individuals $P_G = \{X_{1,G}, X_{2,G}, \ldots, X_{NP,G}\}$ with the i^{th} individual $\boldsymbol{X}_{i,G} = \{x_{i,G}^1, x_{i,G}^2, ..., x_{i,G}^D\}$. The elements of $X_{i,G}$ are randomly and uniformly distributed in the range $[X_{\min}, X_{\max}]$, where $\boldsymbol{X}_{\min} = \{x_{\min}^1, x_{\min}^2, ..., x_{\min}^D\}$ and $\boldsymbol{X}_{\max} = \{x_{\max}^1, x_{\max}^2, ..., x_{\max}^D\}$. D is the number of variables in the objective function. To ensure the randomness and ergodicity, the one-dimensional mapping logistic chaotic model is introduced to compute the initial values of $X_{i,G}$. The chaotic sequences are computed as follows:

$$r_i^{j+1} = \mu \cdot r_i^j \left(1 - r_i^j\right), \; i = 1, 2, ..., NP, \; j = 1, 2, ..., D-1 \text{ and } 0 \le \mu \le 4 \tag{6}$$

where $r_i^j \in [0, 1)$ is the chaotic variable (with r_i^1 given by a uniform random number generator) and $\mu \in (3.57, 4]$ is a control parameter to ensure that the sequences are chaotic (Xiao et al., 2014). For convenience, we fix it to 3.8 in this work. Thus, the initial values of $X_{i,G}$ are given by:

$$x_{i,G}^j = r_i^j \cdot \left(x_{\max}^j - x_{\min}^j\right) + x_{\min}^j, \; j = 1, 2, ..., D \tag{7}$$

Step 2 (Main loop) If terminal conditions are not satisfied, we go through one evolutionary loop described below. Otherwise, the final results are produced, and the program is terminated.

The terminal conditions include three situations:

1 the iteration times reach the maximum N_a

2 the value of the objective function remains unchanged N_b times

3 the value of the objective function remains unchanged N_c times after all of the constrains are satisfied, where $N_a \ge N_b \ge N_c$ and the actual values are proportional to D.

In this step, we choose N_a = 5,000 D, N_b = 500 D, and N_c = 300 D.

Step 3 (Phase determination) Check whether all the constraints have been satisfied. If so, we use the optimum seeking phase, otherwise we use the constraint handling phase. These phases are described separately as step 4 and 5 below.

Step 4 (Constraint handling phase) The main goal of this phase is to quickly find a feasible solution.

Step 4.1 (Check catastrophic critical condition) Fast search sometimes leads to premature convergence which is called catastrophe (Guo et al., 2007). To prevent search from being in stagnation behaviour, the catastrophic check and handling are firstly introduced in the constraint handling phase. The catastrophic critical condition is satisfied if:

$$f_a(G) - f_a(G-S) < -c\left(\frac{f_m(G)}{f_m(G-1)} - 1\right)^{3/2} \tag{8}$$

where $f_a(G)$ and $f_m(G)$ are the average fitness and best fitness in generation G, respectively. The average fitness in generation $G - S$ is $f_a(G - S)$ and thus $[f_a(G) - f_a(G - S)] / S$ represents the average evolution rate in the last S generations. In addition, c is an empirical parameter given by $c = 2 \times 3^{1/2} S / 9$ with $S = 60\ D$ in this paper. If the catastrophic critical condition is satisfied, we proceed to Step 4.2 for catastrophic handling, otherwise we go to Step 4.3.

Step 4.2 (Catastrophic handling) In this step, the population P_G is reset in order to overcome the catastrophe. To enhance the diversity of population, the good point set method, random method and chaos method are employed in turn to generate the population.

The good point set method comes from number theory, which can make individuals in the initial population distribute uniformly throughout the search space as many as possible (Long, 2012). While ensuring the diversity of population, it improves the global search capability of this algorithm. The good point variable $p_i^j \in [0, 1)$ is given by:

$$p_i^j = 2\cos\left(\frac{2\pi j}{t}\right)\cdot i - \left\lfloor 2\cos\left(\frac{2\pi j}{t}\right)\cdot i \right\rfloor, \quad i = 1, 2, ..., NP \text{ and } j = 1, 2, ..., D \tag{9}$$

where t is chosen as the smallest prime number greater than or equal to $2D + 3$. We fix its value at 131 based on empirical value. The initial values of $X_{i,G}$ are then given by:

$$x_{i,G}^j = p_i^j \cdot \left(x_{\max}^j - x_{\min}^j\right) + x_{\min}^j, \quad j = 1, 2, ..., D \tag{10}$$

The random method is the most common way to randomly generate a population. The values of $X_{i,G}$ are given by:

$$x_{i,G}^j = rand_i[0, 1) \cdot \left(x_{\max}^j - x_{\min}^j\right) + x_{\min}^j, \quad j = 1, 2, ..., D \tag{11}$$

The chaos method was described in Step 1. The three methods are applied sequentially.

Step 4.3 (Fixed parameter setting) Optimisation methods all have fundamental parameters (referred to as the hyper-parameters) that must be tuned for each specific problem in order for the method to work well. Tuning is a tedious and labour-intensive process. In the TPDE, the hyper-parameters are kept fixed in the constraint handling phase. According to the results of verification experiment, the hyper-parameters are held fixed with $NP = 100$, $F = 0.95$ and $CR = 0.9$ (the meanings of F and CR will be explained in Step 4.4 and Step 4.5, respectively). Only when the maximal constraint violation is less than 10 and D is more than 50, the hyper-parameters will be lowered to $F = 0.3$ and $CR = 0.5$ to enhance local searching ability. However, this process is dynamically determined and the performance of the TPDE is rather insensitive to the choice of those hyper-parameter values, thus requiring practically no fine tuning.

Step 4.4 (Mutation with mixed evolution scheme) A mutation process generates a mutated vector $\boldsymbol{V}_{i,G} = \{v_{i,G}^1, v_{i,G}^2, ..., v_{i,G}^D\}$ for each target vector $X_{i,G}$. There are three main types of mutation schemes for DE (He et al., 2010). The first type includes DE/rand/1/bin and DE/rand/2/bin that are good at performing global search of the solution space but often result in slow convergence. The second type includes DE/best/1/bin and DE/best/2/bin that favours local searching and efficient convergence but is sometimes too greedy and thus converges to only local optima. The last type is DE/rand-to-best/1/bin, which has good adaptability but poor robustness. We use all three types of mutation schemes in succession in order to balance the goals to adequately explore the solution space and to enhance convergence. These three mutation schemes are:

DE/rand/1/bin:

$$\boldsymbol{V}_{i,G} = X_{r_3^i,G} + F \cdot \left(X_{r_1^i,G} - X_{r_2^i,G} \right) \tag{12}$$

DE/best/1/bin:

$$\boldsymbol{V}_{i,G} = X_{best,G} + F \cdot \left(X_{r_1^i,G} - X_{r_2^i,G} \right) \tag{13}$$

DE/rand-to-best/1/bin:

$$\boldsymbol{V}_{i,G} = X_{i,G} + F \cdot \left(X_{best,G} - X_{i,G} \right) + F \cdot \left(X_{r_1^i,G} - X_{r_2^i,G} \right) \tag{14}$$

where the indices r_1^i, r_2^i, r_3^i are mutually exclusive integers randomly generated within the range [1, NP]. These indices are randomly generated once for each mutant vector. $X_{best,G}$ is the best individual in the current population. The scaling factor F is a positive control parameter for scaling the difference vector. It is one of the hyper-parameters of the TPDE.

Step 4.5 (Crossover) Generate a trial vector $\boldsymbol{U}_{i,G} = \{u_{i,G}^1, u_{i,G}^2, ..., u_{i,G}^D\}$ for each target vector $X_{i,G}$ in the current population so that:

$$u_{i,G}^j = \begin{cases} v_{i,G}^j, & \text{if} \quad (rand[0,1) \leq CR) \text{ or } (j = j_{rand}) \\ x_{i,G}^j, & \text{other} \end{cases} \tag{15}$$

with $j_{rand} = \lceil rand[0, 1) \cdot D \rceil$. CR is the crossover probability controlling the fraction of component parameter values that are copied from the mutant. It is another hyper-parameter of the optimisation method.

Step 4.6 (Boundary handling) Mutation may result in potential solution vectors that violate the required boundary constraints. This is handled via the bounce-back method so that if $u_{i,G}^j > x_{\max}^j$ then it is replaced by:

$$u_{i,G}^j = x_{\max}^j - rand[0,1) \cdot \left(u_{i,G}^j - x_{\max}^j\right) \tag{16}$$

and if $u_{i,G}^j < x_{\min}^j$ then it is replaced by:

$$u_{i,G}^j = x_{\min}^j + rand[0,1) \cdot \left(x_{\min}^j - u_{i,G}^j\right) \tag{17}$$

Step 4.7 (Selection with constraint handling) Each of the vectors is then compared with its own potential offspring for selection to be included in the population pool in the next generation. The selection criteria depend on the maximum amount, the total number, and the total amount of constraint violations of each vector. Given two potential solution vectors, the maximal values of constraint violations are compared first, then the total numbers of constraint violations, and finally the total amount of constraint. When three of these quantities are exactly the same, then the better solution to be selected is the one with the lowest objective function value.

Step 5 (Optimum seeking phase) The main task of this phase is to approach the optimal solution step by step.

Step 5.1 (Adaptive parameter setting) To gradually stabilise searching for the local optimum, the hyper-parameter F in TPDE should be automatically adjusted (Chen et al., 2011). Specifically, F is dynamically reduced when the optimum remains unchanged in $2D$ generations computed as follows:

$$F = \frac{F_{\min}}{1 + \left(\frac{F_{\min}}{F_{\max}} - 1\right)e^{-at}} \tag{18}$$

where $F_{\min}$ and $F_{\max}$ are the minimal and maximal values of F; t is the iteration times of change in F; a is the reducing rate of F. In this study, we have $F_{\min} = 0.3$, $F_{\max} = 0.95$ and $a = 0.002/D$ after verification experiment.

Step 5.2 (Mutation with evolution scheme selection) When the dimension of the problem is not too high, the random evolution scheme is expected to be able to find the optimal solution quickly. However, with higher dimensions, the mixed evolution scheme in Step 4.4 should provide a more balanced, robust, and efficient mutation result. In this step, we use DE/rand/1/bin when $D <= 10$, otherwise we use the mixed evolution scheme.

Step 5.3 (Crossover) The trial vector for each target vector is generated by the operations described in Step 4.5.

Step 5.4 (Boundary handling) The bounce-back method as described in Step 4.6 is also used here to handle boundary constraint violations.

Step 5.5 (Selection with constraint satisfaction) Because all the constraints have been satisfied by the current solution vector, the newly generated vector must also satisfy all the constraints. Consequently, the better solution is the one with the lowest objective function values.

Step 5.6 (Adjustment based on crowding level) To lessen the chance that the best solution is stuck at one of the local optima, the evolving ability of current population needs to be checked (Guo et al., 2007). The crowding level of the current population is defined by:

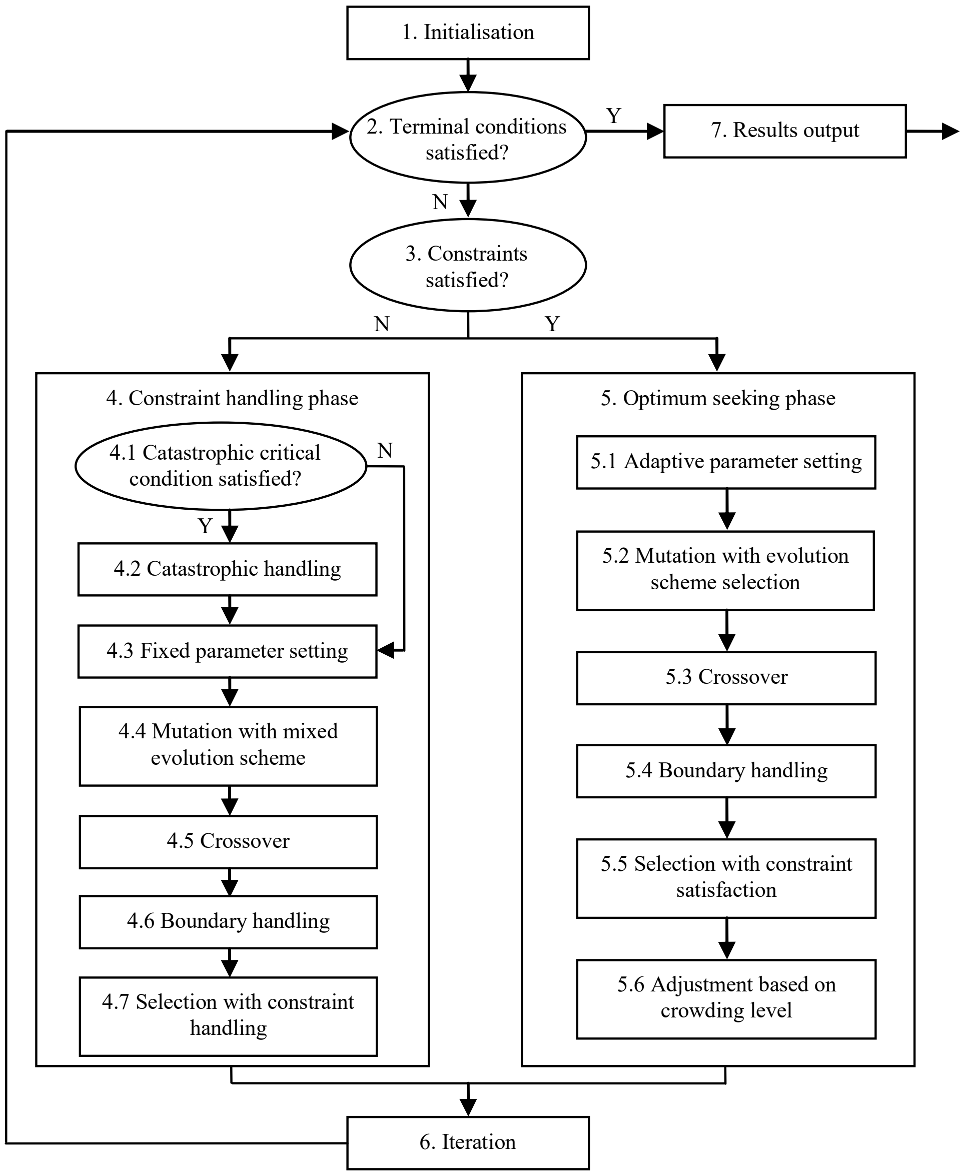
$$cl = \min\left(f_a(G), f_m(G)\right) / \max\left(f_a(G), f_m(G)\right) \quad (19)$$

where $f_a(G)$ and $f_m(G)$ are the average fitness and the best fitness of generation G, respectively. By definition, it has a value lying between zero and one. A larger cl means a higher crowding level and less distinction among the individuals in the population. When $cl = 1$ all the individual vectors are the same, and we will generate random vectors to replace the original vectors directly. When cl is larger than a certain critical value cl_c or the optimum remains unchanged in N_d times, the mutated vectors will replace the original vectors with a probability p_c. In this study, we have $cl_c = 0.999$, $N_d = 125\ D$ and $p_c = 0.5$ based on empirical values. These adjustments can further ensure the evolving ability of the current population.

Step 6 (Iteration) Increase the generation number $G = G + 1$ and jump to Step 2.

Step 7 (Results output) Output the optimisation results.

Figure 3 Optimisation process of the TPDE algorithm



5 Optimisation experiment and analysis

5.1 Experimental environment and data

The TPDE is later applied to the above material supply model. Our results are compared with those obtained using a commercial optimisation software and with several other evolutionary computational methods. Computation was performed on a machine with a

2.4 GHz CPU, a 1.92G RAM and a 32-bit OS. The software included LINGO 10.0 which was a commercial optimisation tool for NLP problems, and four EAs programs: GA, PSO, DE and TPDE. All the programs were compiled and run in MATLAB.

Table 1 Basic information of supply centre candidates

Supply centre no.	*Storage capacity (ton)*	*Startup cost (RMB)*
1	26,000	1,440,000
2	19,700	2,990,000
3	27,400	500,000
4	30,100	500,000

Table 2 Basic information of disaster areas

Disaster area no.	*Material quantity demanded (ton)*	*Time limit in emergency (h)*
5	440	24
6	450	24
7	320	24
8	320	48

Table 3 Transport information between supply centres and disaster areas

Starting point	*Terminal point*	*Transport capacity for an assignment (Ton)*	*Transport time in good condition (h)*	*Unit transport cost (RMB/ton)*	*Total repair cost (RMB)*	*Repair time in all damaged (h)*	*Damage probability of roads (%)*
1	5	90	6	100	30,000	33	12
2		50	4	130	0	17	17
3		120	7	190	190,000	35	13
4		50	6	140	300,000	38	2
1	6	100	7	100	50,000	20	9
2		90	5	130	160,000	22	0
3		160	8	190	260,000	21	10
4		50	7	140	170,000	34	13
1	7	50	10	100	100,000	40	1
2		170	5	130	30,000	19	17
3		180	9	190	290,000	33	4
4		180	4	140	150,000	5	12
1	8	180	6	100	90,000	35	5
2		80	12	130	40,000	37	2
3		190	4	190	170,000	45	11
4		120	7	140	100,000	19	1

According to China's emergency plan, material supply centres are established well in advance of any disasters. The number of these centres is small and their locations are fixed and are pre-determined based on historical facts and scientific insights. The locations and the number of disaster areas are unpredictable, and vary from one emergency to the next. The number of disaster areas needing material can vary with the situation of disaster. Based on actual emergency events that occurred in China, we selected nine sets of experimental data to work with. In addition to a dataset involving four material supply centres and two disaster areas (referred to as 4-2 for short) that was reported in Zhao (2012), the other eight sets of experimental data are referred to as 4-4, 4-8, 4-12, 4-16, 4-20, 4-24, 4-28 and 4-32. The 4-4 experimental data is selected as an example shown in Table 1 to Table 3.

Each optimisation method was run ten separate times for each of the nine datasets. The settings of fundamental control parameters for each optimisation method are shown in Table 4.

Table 4 Parameter settings for different optimisation methods

Optimisation methods	*Fundamental control parameters*
LINGO	Model: INLP, solver: global
	Strategy: branching – relative violation, box selection – worst bound, reformulation – high
GA	*NP* = 40, *Pm* = 0.8, crossover: single point with two-parent and two-offspring
PSO	*NP* = 100, *c*1 = 2, *c*2 = 2, *w* = 0.7298
DE	*NP* = 100, *F* = 0.7, *CR* = 0.9, strategy: DE/rand/1/bin
TPDE	*NP* = 100, *CR* = 0.9
	Constraint handling phase: *F* = 0.95, strategy is mixed evolution scheme
	Optimum seeking phase: *F* is adjusted by logistic model, strategy is selected according to dimension

5.2 Analysis of experimental results from different optimisation methods

The statistical experimental results for evolution iteration, constraint satisfied iteration, deviation time, function value and infeasible times for each of the datasets are tabulated in Table 5 to Table 13. The evolution iteration is the total number of generations required in order for at least one of the terminal conditions to be met. The constraint satisfied iteration shows the number of generations that pass before the first feasible solution is found. Unfortunately, this particular piece of information is not available for LINGO. The deviation time is defined as the maximum time difference between the actual supply time and the time limits imposed by each of the disaster areas. A negative deviation time means that the needed material can be delivered to all the disaster areas ahead of the required times. In some runs involving a large number of unknowns, a method may fail to find even one feasible solution before reaching the maximum allowed number of generations. In this paper, the function value is the supply cost after the time efficiency function is translated into a constraint. Infeasible times show for each of the methods the total number of failures to find a feasible solution in ten runs. The results of runs that fail to produce any acceptable solution are not included in the tables. For each given dataset, numbers enclosed in parentheses give the ranking among the five methods for each attribute.

Table 5 Statistical experimental results with 4-2 experimental data

Optimisation methods	*Mean evolution iteration*	*Mean constraint satisfied iteration*	*Mean deviation time (h)*	*Mean supply cost (RMB)*	*Infeasible times*
LINGO	133 (1)	-	–4	6,270,600 (3)	0
GA	5,182 (5)	331 (3)	–4.9	6,873,376 (5)	0
PSO	3,084 (2)	468 (4)	–4.4	6,667,678 (4)	0
DE	3,589 (3)	109 (1)	–5	6,165,700 (2)	0
TPDE	4,164 (4)	112 (2)	–4.4	6,063,252 (1)	0

Table 6 Statistical experimental results with 4-4 experimental data

Optimisation methods	*Mean evolution iteration*	*Mean constraint satisfied iteration*	*Mean deviation time (h)*	*Mean supply cost (RMB)*	*Infeasible times*
LINGO	239 (1)	-	–1.11	5,726,270 (3)	0
GA	32,212 (5)	1,096 (2)	–1.11	5,756,302 (5)	0
PSO	8,035 (2)	2,854 (3)	–1.13	5,739,468 (4)	0
DE	18,558 (4)	3,324 (4)	–1.12	5,712,192 (2)	0
TPDE	10,194 (3)	635 (1)	–1.11	5,698,500 (1)	0

Table 7 Statistical experimental results with 4-8 experimental data

Optimisation methods	*Mean evolution iteration*	*Mean constraint satisfied iteration*	*Mean deviation time (h)*	*Mean supply cost (RMB)*	*Infeasible times*
LINGO	462 (1)	-	–1.2	5,323,360 (3)	0
GA	150,597 (5)	3,487 (3)	–1.24	5,347,940 (4)	0
PSO	12,669 (2)	2,592 (1)	–1.36	5,248,570 (2)	0
DE	64,717 (4)	48,518 (4)	–1.6	5,373,322 (5)	0
TPDE	34,886 (3)	3,472 (2)	–1.2	5,034,876 (1)	0

Table 8 Statistical experimental results with 4-12 experimental data

Optimisation methods	*Mean evolution iteration*	*Mean constraint satisfied iteration*	*Mean deviation time (h)*	*Mean supply cost (RMB)*	*Infeasible times*
LINGO	2,549 (1)	-	–2.25	8,154,780 (3)	0
GA	164,849 (5)	10,099 (3)	–4.07	8,316,771 (5)	0
PSO	24,138 (2)	9,303 (2)	–1.71	7,912,477 (2)	0
DE	125,402 (4)	99,403 (4)	–4.49	8,294,667 (4)	4
TPDE	51,403 (3)	4,825 (1)	–2.96	6,243,220 (1)	0

Following is a summary of the overall performance of each method compared with those of all others for the nine datasets using the hyper-parameters as shown in Table 4.

The mean supply cost found by LINGO was mediocre compared with the others. However, the mean evolution iterations were less than other methods when the dimension

D of the problem was less than 64 (4-16) as shown in Table 5 to Table 9. In addition, LINGO was able to find feasible solutions for all of the cases. Thus, it appears that for the type of problems considered here, LINGO is rather reliable although the optimal results found are not very good.

Table 9 Statistical experimental results with 4-16 experimental data

Optimisation methods	*Mean evolution iteration*	*Mean constraint satisfied iteration*	*Mean deviation time (h)*	*Mean supply cost (RMB)*	*Infeasible times*
LINGO	43,272 (1)	-	–0.2	10,703,620 (3)	0
GA	320,000 (5)	25,823 (2)	–0.2	11,002,156 (4)	0
PSO	77,321 (2)	56,412 (3)	–0.2	10,571,930 (2)	0
DE	196,805 (4)	167,308 (4)	–0.2	11,046,293 (5)	4
TPDE	97,767 (3)	3,586 (1)	–0.2	10,391,835 (1)	0

Table 10 Statistical experimental results with 4-20 experimental data

Optimisation methods	*Mean evolution iteration*	*Mean constraint satisfied iteration*	*Mean deviation time (h)*	*Mean supply cost (RMB)*	*Infeasible times*
LINGO	619,973 (5)	-	–1.2	10,651,400 (3)	0
GA	284,148 (4)	212,213 (3)	–0.87	10,883,850 (5)	2
PSO	101,920 (1)	77,203 (2)	–0.96	10,330,388 (1)	6
DE	261,988 (3)	231,692 (4)	–1.2	10,840,267 (4)	1
TPDE	221,761 (2)	31,182 (1)	–0.72	10,343,280 (2)	0

Table 11 Statistical experimental results with 4-24 experimental data

Optimisation methods	*Mean evolution iteration*	*Mean constraint satisfied iteration*	*Mean deviation time (h)*	*Mean supply cost (RMB)*	*Infeasible times*
LINGO	206,142 (3)	-	–1.46	7,754,900 (3)	0
GA	480,000 (5)	36,497 (2)	–1.04	8,046,610 (4)	0
PSO	81,222 (1)	52,161 (3)	–2.22	7,169,606 (2)	0
DE	338,568 (4)	294,553 (4)	–1.53	8,252,790 (5)	4
TPDE	181,433 (2)	27,969 (1)	–1.48	6,469,738 (1)	0

Table 12 Statistical experimental results with 4-28 experimental data

Optimisation methods	*Mean evolution iteration*	*Mean constraint satisfied iteration*	*Mean deviation time (h)*	*Mean supply cost (RMB)*	*Infeasible times*
LINGO	431,850 (4)	-	–0.64	9,450,990 (3)	0
GA	560,000 (5)	34,341 (2)	–0.64	10,076,243 (4)	0
PSO	99,746 (1)	65,370 (3)	–0.87	9,411,035 (2)	0
DE	382,808 (2)	334,050 (4)	–1.34	10,264,009 (5)	3
TPDE	392,542 (3)	18,549 (1)	–0.64	8,820,348 (1)	0

Table 13 Statistical experimental results with 4-32 experimental data

Optimisation methods	*Mean evolution iteration*	*Mean constraint satisfied iteration*	*Mean deviation time (h)*	*Mean supply cost (RMB)*	*Infeasible times*
LINGO	280,533 (2)	-	–1.96	9,728,670 (3)	0
GA	633,035 (5)	34,241 (1)	–2.21	10,121,402 (4)	0
PSO	263,604 (1)	224,858 (3)	–1.36	9,108,950 (2)	6
DE	458,517 (3)	400,597 (4)	–3.06	10,331,187 (5)	3
TPDE	521,673 (4)	140,744 (2)	–1.82	8,186,763 (1)	0

The mean supply costs found by GA were either the highest or very near the highest compared with those found by other methods. The mean evolution iterations were almost always the highest, especially for the high *D* cases. The mean constraint satisfied iterations were nearly medium. GA was able to find feasible solutions in most cases except for the dataset 4-20 where it failed twice out of ten runs, as shown in Table 10. Thus, GA, at least in the form investigated here, is not quite suitable for handling the present emergency material supply model.

Table 14 Best solution for 4-4 experimental data with TPDE

Starting point	*Terminal point*	*Supply quantities (ton)*	*Times of single trip*	*Catastrophic times*	*Deviation time (h)*	*Total supply cost (RMB)*
1	5	180	3	0	–1.11	5,687,600
2		150	5			
3		10	1			
4		100	3			
1	6	200	3			
2		180	3			
3		70	1			
4		0	0			
1	7	50	1			
2		270	3			
3		0	0			
4		0	0			
1	8	320	3			
2		0	0			
3		0	0			
4		0	0			

In most cases, the supply costs found by PSO were only higher than those found by TPDE. The mean evolution iterations were also very respectable. However, PSO lacks robustness, with several failed runs for datasets 4-20 and 4-32, as shown in Table 10 and Table 13. And the mean constraint satisfied iterations frequently changed from the highest to the lowest. That finding weakens its reliability for the present study.

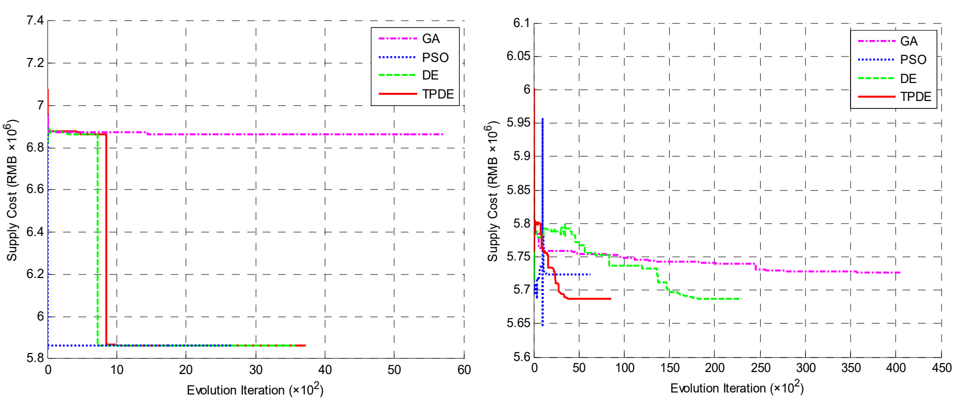
The conventional DE gives good optimisation results with the datasets at low dimensions as we found in an earlier study, but when dimensions of the problem increase, its performance drastically diminished. The mean supply costs were the highest when *D* was more than 32 (4-8), as shown in Table 7 to Table 13. The mean constraint satisfied iterations were the highest when *D* was more than 16 (4-4), as shown in Table 6 to Table 13. And there were always failed runs when *D* was more than 48 (4-12), as shown in Table 8 to Table 13. Thus, it is difficult to work with the conventional DE for very high dimensional data in these models.

TPDE obtained the lowest supply costs in each of the datasets except for 4-20 as shown in Table 10. For this dataset, TPDE came in second after PSO. However, PSO failed to find even a feasible solution in 6 out of 10 runs. There were no failed runs with TPDE. It was able to find a feasible solution in all cases, and almost always in the fewest number of generations. This potential is mostly owe to the two-phase strategy. Compared with the other methods, TPDE shows the rapid evolution of the deviation time in the constraint handling phase and the substantial evolution of the supply cost in the optimum seeking phase. The best solution for the 4-4 experimental data with TPDE is shown as an example in Table 14.

5.3 *Comparison of optimisation performances between TPDE and other EAs*

To compare the optimisation performance of different EAs, we particular concern how the supply cost evolves from one generation to the next for each of the datasets. In each case, the results for the best run are displayed in Figure 4.

Figure 4 Comparison in evolutionary processes of different EAs, (a) evolutionary processes in 4-2 experiment data (b) evolutionary processes in 4-4 experiment data (c) evolutionary processes in 4-8 experiment data (d) evolutionary processes in 4-12 experiment data (e) evolutionary processes in 4-16 experiment data (f) evolutionary processes in 4-20 experiment data (g) evolutionary processes in 4-24 experiment data (h) evolutionary processes in 4-28 experiment data (i) evolutionary processes in 4-32 experiment data (see online version for colours)



(a) (b)

Figure 4 Comparison in evolutionary processes of different EAs, (a) evolutionary processes in 4-2 experiment data (b) evolutionary processes in 4-4 experiment data (c) evolutionary processes in 4-8 experiment data (d) evolutionary processes in 4-12 experiment data (e) evolutionary processes in 4-16 experiment data (f) evolutionary processes in 4-20 experiment data (g) evolutionary processes in 4-24 experiment data (h) evolutionary processes in 4-28 experiment data (i) evolutionary processes in 4-32 experiment data (continued) (see online version for colours)

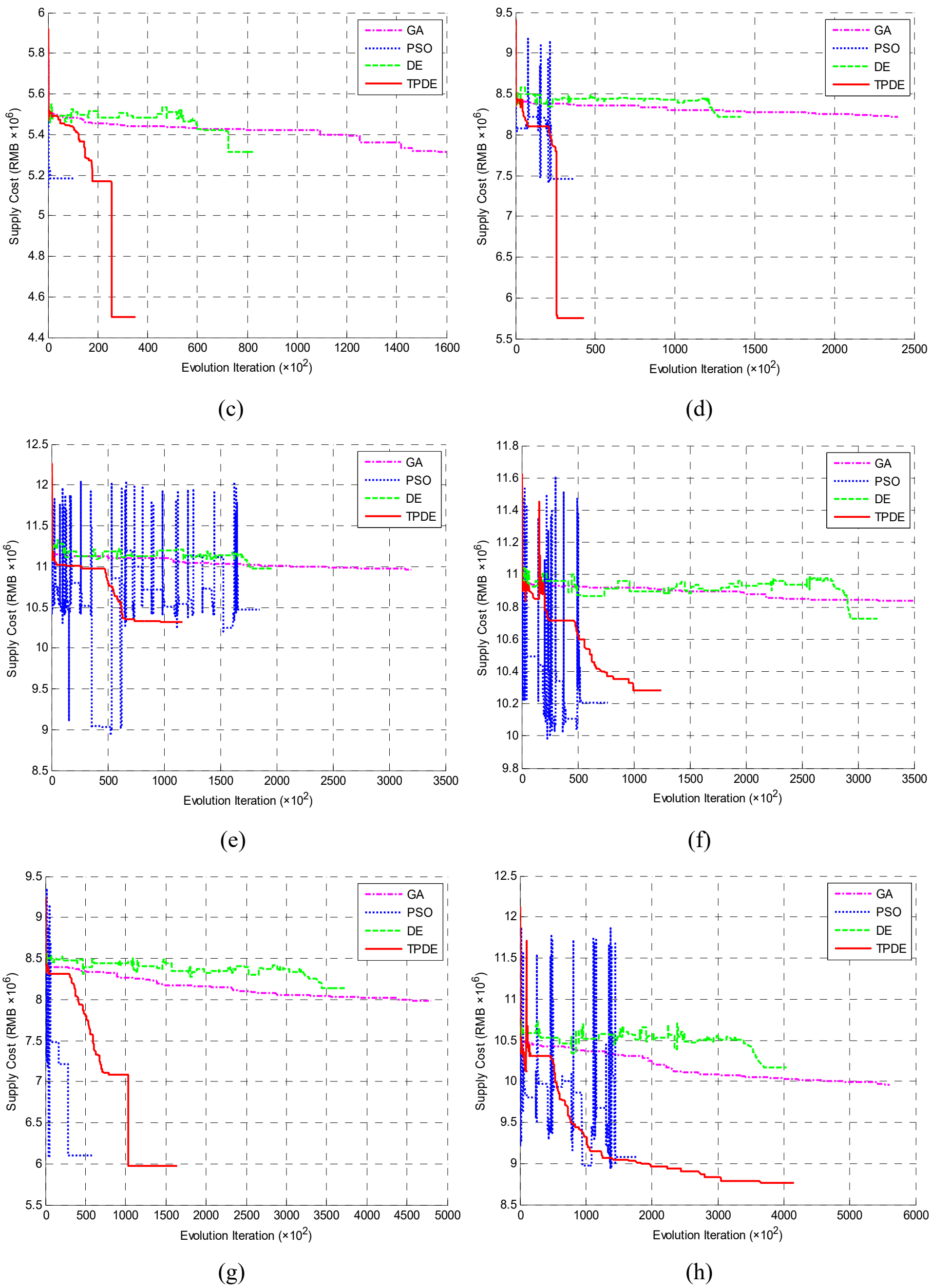
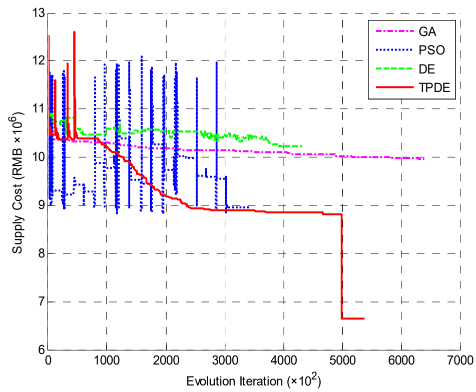


Figure 4 Comparison in evolutionary processes of different EAs, (a) evolutionary processes in 4-2 experiment data (b) evolutionary processes in 4-4 experiment data (c) evolutionary processes in 4-8 experiment data (d) evolutionary processes in 4-12 experiment data (e) evolutionary processes in 4-16 experiment data (f) evolutionary processes in 4-20 experiment data (g) evolutionary processes in 4-24 experiment data (h) evolutionary processes in 4-28 experiment data (i) evolutionary processes in 4-32 experiment data (continued) (see online version for colours)



(i)

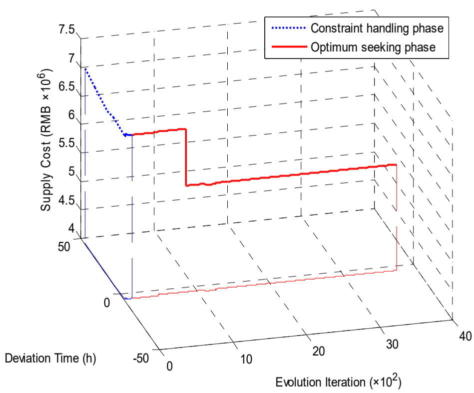
The evolution plot for GA often exhibits an extended flat portion at high supply cost, as shown in Figures 4(a) to 4(i). This finding means that GA encountered difficulties in finding a better solution. The final solutions found by GA were typically inferior to those of the other methods. The evolution plot of PSO typically had a fair number of spikes as shown in Figures 4(b) to 4(i). This is due to the weakness of PSO in finding feasible solutions. However, once a feasible solution was found, convergence occurred rather quickly. Except for low dimension data such as 4-2 and 4-4, the evolution plots of DE were also slightly flat initially with high supply cost, but for fewer generations than GA, as shown in Figures 4(c) to 4(i). The converged results of DE were very poor for high dimension data. The corresponding plots for TPDE always began with small spikes then were followed with gradual lowering of cost as shown in Figures 4(a) to 4(i). The small spikes are signatures of the constraint handling phase where feasible solutions were found quickly. Subsequent steps in the plots showed the cost of solutions found in the optimum seeking phase where better feasible solutions were found. Almost all of the best results in this study were obtained by using the TPDE. Even in the most complex data group of 4-32 as shown in Figure 4(i), the supply cost computed from TPDE was 67% of that from GA, 74% of that from PSO, and 65% of that from DE.

5.4 Comparison of evolutionary characteristics of two phases in TPDE

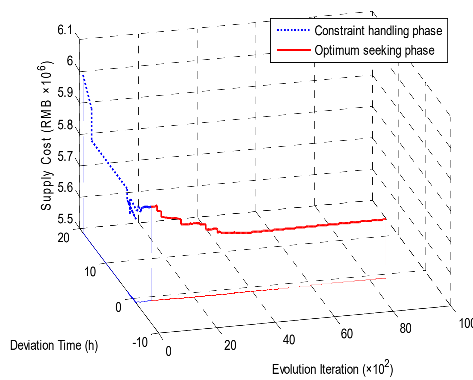
To elucidate the way that TPDE works in more detail, we focus on the best cases for the nine groups of experimental data and plot the variation of the supply cost and the deviation time of the best solution in each generation. These results, together with their projections onto a constant supply cost plane, are shown in Figure 5.

The typical influence of the constraint handling phase on the best solution occurs during the initial part of the evolutionary process as shown in Figures 5(f), 5(h) and 5(i). It plays a vital role in quickly finding feasible solutions by focusing on the constraints and not the supply cost. We see from these figures that the deviation time was almost always positive and the decrease in the supply cost was quite slow in this phase. Most of the evolutionary processes were spent in the subsequent optimum seeking phase of the best solution where the best solution continued to satisfy the constraints (with the deviation time remaining negative) as the supply cost was lowered. It may take many generations before finding another better feasible solution, but the resulting deduction in the supply cost can be substantial, as seen in Figures 5(a), 5(c), 5(d), 5(g) and 5(i). The variations in the properties of the best solution with the number of generations are quite different in the two phases. However, it is the dynamic interplay of both of these phases that enables the TPDE method to not only find the feasible solution quickly but also approach the optimal solution effectively.

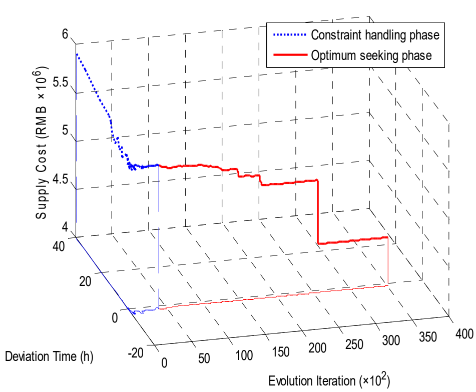
Figure 5 Two-phase evolutionary processes of TPDE in 3D, (a) two-phase evolutionary process in 4-2 experiment data (b) two-phase evolutionary process in 4-4 experiment data (c) two-phase evolutionary process in 4-8 experiment data (d) two-phase evolutionary process in 4-12 experiment data (e) two-phase evolutionary process in 4-16 experiment data (f) two-phase evolutionary process in 4-20 experiment data (g) two-phase evolutionary process in 4-24 experiment data (h) two-phase evolutionary process in 4-28 experiment data (i) two-phase evolutionary process in 4-32 experiment data (see online version for colours)



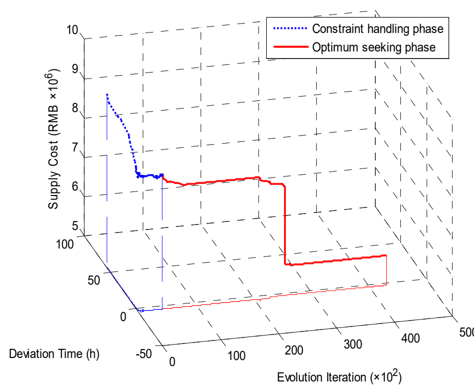
(a)



(b)

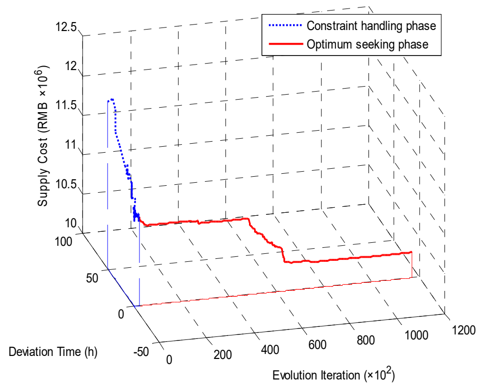


(c)

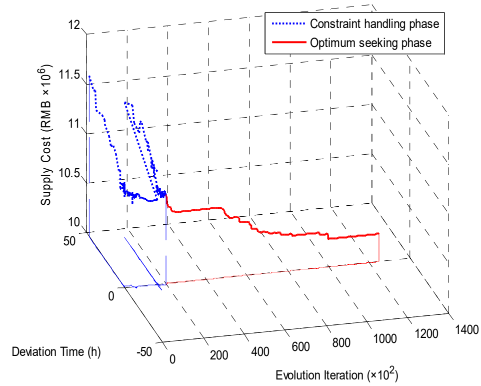


(d)

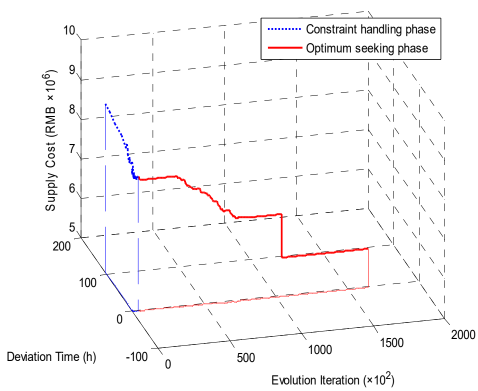
Figure 5 Two-phase evolutionary processes of TPDE in 3D, (a) two-phase evolutionary process in 4-2 experiment data (b) two-phase evolutionary process in 4-4 experiment data (c) two-phase evolutionary process in 4-8 experiment data (d) two-phase evolutionary process in 4-12 experiment data (e) two-phase evolutionary process in 4-16 experiment data (f) two-phase evolutionary process in 4-20 experiment data (g) two-phase evolutionary process in 4-24 experiment data (h) two-phase evolutionary process in 4-28 experiment data (i) two-phase evolutionary process in 4-32 experiment data (continued) (see online version for colours)



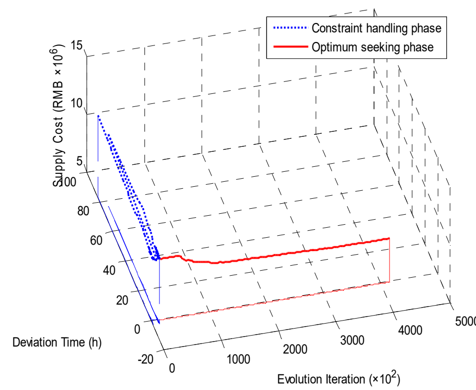
(e)



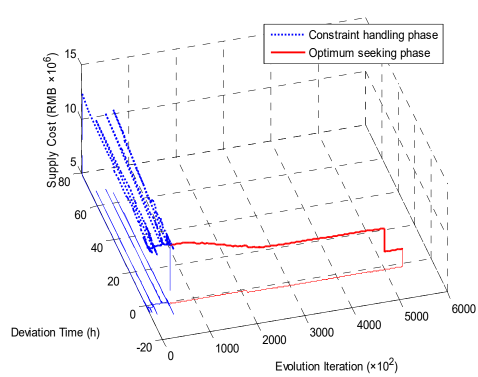
(f)



(g)



(h)



(i)

5.5 Extra experiment for robustness of TPDE

To further verify the robustness of TPDE, four sets of 4-28 and four sets of 4-32 experimental data are extra selected to work with. The statistical experimental results are tabulated in Table 15. There were still no failed runs in all extra datasets. The feasible solution could always be found in a short number of generations. And the final supply costs were always similar in ten runs of each extra dataset. Thus, TPDE is a robust and highly reliable method that produces the best solutions for the model considered here.

Table 15 Statistical experimental results with extra experimental data

Dataset	*Mean evolution iteration*	*Mean constraint satisfied iteration*	*Mean deviation time (h)*	*Mean supply cost (RMB)*	*Infeasible times*
4-28-E1	336,743	13,771	–1.34	7,823,452	0
4-28-E2	287,442	12,287	–2.16	6,843,226	0
4-28-E3	399,426	28,862	–1.24	8,542,421	0
4-28-E4	420,923	20,448	–0.96	8,043,027	0
4-32-E1	553,796	120,342	–1.35	9,025,678	0
4-32-E2	442,878	89,033	–2.32	8,295,547	0
4-32-E3	610,045	175,689	–1.08	9,420,029	0
4-32-E4	582,542	97,894	–0.73	8,754,329	0

6 Conclusions

An emergency material supply model was constructed especially for serious disasters. By dynamically moving from the constraint handling phase to the optimum seeking phase, the TPDE method can quickly find solutions to satisfy the required constraints and subsequently identify better solutions with lower cost. Different evolution schemes and special handling techniques are used to improve the ability of breaking away from the premature convergence or local optimal solution, enhance the diversity and evolutionary ability of the population, and balance the global searching of solution space and the local searching for efficient convergence. And there is not much fine-tuning required to adjust the TPDE and its specific hyper-parameters. By comparing the performance of TPDE with LINGO, GA, PSO and conventional DE on a variety of datasets in the challenging model, we found that TPDE was always able to find feasible solutions in the shortest amount of time, and produced the best optimal solutions in nearly all the cases. We encountered no failed cases in all 170 runs of TPDE, which demonstrates its superior robustness for the present material supply model. Altogether, this work has indicated the effectiveness and efficiency of the proposed algorithm.

The future research work mainly includes:

1 applying the TPDE to more complex benchmark problems and other types of nonlinear constrained optimisation problems

2 improving the TPDE to fit with high performance parallel computing environment and enhancing the efficiency of algorithm execution

3 improving the emergency material supply model from the perspectives of demand satisfaction and service fairness

4 formulating more standard test suites, real-world cases, and evaluation tools for assessment and analysis of different optimisation algorithms.

Acknowledgements

This study is partially supported by the National Social Science Foundation of China under Grant 20BGL310.

References

Caunhye, A.M., Nie, X. and Pokharel, S. (2012) 'Optimization models in emergency logistics: a literature review', *Socio-Economic Planning Sciences*, Vol. 46, No. 1, pp.4–13.

Chen, H., Fan, Y-R. and Deng, S-G. (2011) 'Adaptive differential evolution algorithm based on logistic model', *Control and Decision*, Vol. 26, No. 7, pp.1105–1108.

Guo, Z., Cheng, B., Ye, M., Kang, L. and Cao, B. (2007) 'Parallel chaos differential evolution algorithm', *Journal of Xi'an Jiaotong University*, Vol. 41, No. 3, pp.299–302.

He, Y-C., Wang, X-Z., Liu, K-Q. and Wang, Y-Q. (2010) 'Convergent analysis and algorithmic improvement of differential evolution', *Journal of Software*, Vol. 21, No. 5, pp.875–885.

Holguín-Veras, J., Pérez, N., Jaller, M., Wassenhove, L.N.V. and Aros-Vera, F. (2013) 'On the appropriate objective function for post-disaster humanitarian logistics models', *Journal of Operations Management*, Vol. 31, No. 5, pp.262–280.

Hong, J-D. and Jeong, K-Y. (2019) 'Humanitarian supply chain network design using data envelopment analysis and multi-objective programming models', *European Journal of Industrial Engineering*, Vol. 13, No. 5, pp.651–680.

Hoyos, M.C., Morales, R.S. and Akhavan-Tabatabaei, R. (2015) 'OR models with stochastic components in disaster operations management: a literature survey', *Computers & Industrial Engineering*, Vol. 82, pp.183–197.

Jiang, J-C., Li, Q-Q., Wu, L-X. and Tu, W. (2017) 'Multi-objective emergency material vehicle dispatching and routing under dynamic constraints in an earthquake disaster environment', *ISPRS International Journal of Geo-Information*, Vol. 6, No. 5, p.142.

Kapucu, N. (2006) 'Interagency communication networks during emergencies: boundary spanners in multiagency coordination', *The American Review of Public Administration*, Vol. 36, No. 2, pp.207–225.

Kellenbrink, C. and Helber, S. (2016) 'Quality- and profit-oriented scheduling of resource-constrained projects with flexible project structure via a genetic algorithm', *European Journal of Industrial Engineering*, Vol. 10, No. 5, pp.574–595.

Liu, W-B., Wang, Z-D., Liu, X-H., Zeng, N-Y. and Bell, D. (2019) 'A novel particle swarm optimization approach for patient clustering from emergency departments', *IEEE Transactions on Evolutionary Computation*, Vol. 23, No. 4, pp.632–644.

Long, W. (2012) 'Dynamic multi-species parallel differential evolution algorithm', *Application Research of Computers*, Vol. 29, No. 7, pp.2429–2431.

Meziani, N., Boudhar, M. and Oulamara, A. (2018) 'PSO and simulated annealing for the two-machine flowshop scheduling problem with coupled-operations', *European Journal of Industrial Engineering*, Vol. 12, No. 1, pp.43–66.

Mohammadi, R., Ghomi, S.M.T.F. and Jolai, F. (2016) 'Prepositioning emergency earthquake response supplies: a new multi-objective particle swarm optimization algorithm', *Applied Mathematical Modelling*, Vol. 40, Nos. 9–10, pp.5183–5199.

Najafi, M., Eshghi, K. and Dullaert, W. (2013) 'A multi-objective robust optimization model for logistics planning in the earthquake response phase', *Transportation Research Part E: Logistics and Transportation Review*, Vol. 49, No. 1, pp.217–249.

Niyomubyeyi, O., Pilesjo, P. and Mansourian, A. (2019) 'Evacuation planning optimization based on a multi-objective artificial bee colony algorithm', *ISPRS International Journal of Geo-Information*, Vol. 8, No. 3, p.110.

Onan, K., Ulengin, F. and Sennaroglu, B. (2015) 'An evolutionary multi-objective optimization approach to disaster waste management: a case study of Istanbul, Turkey', *Expert Systems with Applications*, Vol. 42, No. 22, pp.8850–8857.

Pang, N-S., Liu, H., Huang, S-Y. and Meng, J-J. (2016) 'Emergency rush repair task scheduling of distribution networks in large-scale blackouts', *International Journal of Electrical Power & Energy Systems*, Vol. 82, pp.373–381.

Paul, J.A. and MacDonald, L. (2016) 'Optimal location, capacity and timing of stockpiles for improved hurricane preparedness', *International Journal of Production Economics*, Vol. 174, pp.11–28.

Price, K.V., Storn, R.M. and Lampinen, J.A. (2005) *Differential Evolution: a Practical Approach to Global Optimization*, Springer Press, Heidelberg.

Stanimirovic, Z., Grujicic, I. and Trifunovic, D. (2014) 'Modeling the emergency service network of police special forces units for high-risk law enforcement operations', *Infor.*, Vol. 52, No. 4, pp.206–226.

Sun, J-B., Huang, J-X., Chang, L-L., Jiang, J. and Tan, Y-J. (2018) 'BRBcast: a new approach to belief rule-based system parameter learning via extended causal strength logic', *Information Sciences*, Vol. 444, pp.51–71.

Wang, Y-D., Shi, Q. and Hu, Q-W. (2019) 'Dynamic multi-objective optimization for multi-period emergency logistics network', *Journal of Intelligent & Fuzzy Systems*, Vol. 37, No. 6, pp.8471–8481.

Wendt, K., Cortes, A. and Margalef, T. (2013) 'Parameter calibration framework for environmental emergency models', *Simulation Modelling Practice and Theory*, Vol. 31, pp.10–21.

Xiao, W-X., Xu, L-J. and Ma, X-Q. (2014) 'Research on chaos differential evolution algorithm and its application to complex optimization problems', *Journal of Anhui University (Natural Science Edition)*, Vol. 38, No. 3, pp.32–36.

Yan, S., Lin, C-K. and Chen, S-Y. (2014) 'Logistical support scheduling under stochastic travel times given an emergency repair work schedule', *Computers & Industrial Engineering*, Vol. 67, pp.20–35.

Yu, H. and Chen, J. (2007) 'When to activate contingency plans within disruptive incident', *Systems Engineering Theory and Practice*, Vol. 27, No. 8, pp.27–32.

Yu, L., Zhang, C., Yang, H. and Miao, L. (2018) 'Novel methods for resource allocation in humanitarian logistics considering human suffering', *Computers & Industrial Engineering*, Vol. 119, pp.1–20.

Zhao, S-D. (2012) *Models of Scheduling Emergency Materials and the Applications for the Devastating Earthquake Rescue*, Dissertation, Southwest Jiaotong University, China.

Zheng, Y-J., Chen, S-Y. and Ling, H-F. (2015) 'Evolutionary optimization for disaster relief operations: a survey', *Applied Soft Computing*, Vol. 27, pp.553–566.

Zhu, J-L., Li, W-H., Li, H-Y., Wu, Q. and Zhang, L. (2017) 'A novel swarm intelligence algorithm for the evacuation routing optimization problem', *International Arab Journal of Information Technology*, Vol. 14, No. 6, pp.880–889.

Zuo, X-Q., Li, B., Huang, X-W., Zhou, M-C., Cheng, C-Y., Zhao, X-C. and Liu, Z-S. (2019) 'Optimizing hospital emergency department layout via multiobjective tabu search', *IEEE Transactions on Automation Science and Engineering*, Vol. 16, No. 3, pp.1137–1147.