Computers & Industrial Engineering 104 (2017) 51–67

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

A bi-objective multi-period series-parallel inventory-redundancy allocation problem with time value of money and inflation considerations

Najmeh Alikar [a], Seyed Mohsen Mousavi [b], Raja Ariffin Raja Ghazilla [a,*], Madjid Tavana [c,d], Ezutah Udoncy Olugu [e]

[a] Department of Mechanical Engineering, Faculty of Engineering, University of Malaya, Kuala Lumpur, Malaysia
[b] Young Researchers and Elite Club, Qazvin Branch, Islamic Azad University, Qazvin, Islamic Republic of Iran
[c] Business Systems and Analytics Department, La Salle University, Philadelphia, USA
[d] Business Information Systems Department, Faculty of Business Administration and Economics, University of Paderborn, D-33098 Paderborn, Germany
[e] Department of Mechanical & Mechatronic Engineering, Faculty of Engineering, Technology and Built Environment, UCSI University, 56000 Cheras, Kuala Lumpur, Malaysia

ARTICLE INFO

Article history:
Received 28 February 2016
Received in revised form 25 October 2016
Accepted 7 December 2016
Available online 9 December 2016

Keywords:
Multi-objective inventory system
Redundancy allocation problem
Inflation
Time value of money
CE-NRGA
NSGA-II
MOPSO

ABSTRACT

A large number of existing research studies on reliability redundancy allocation problems do not take into consideration the time value of the money and the inflations costs associated with the component inventories. In this study, we formulate a multi-component multi-period series-parallel inventory redundancy allocation problem (SPIRAP) as a mixed-integer nonlinear mathematical model where: (a) the costs are calculated by considering the time value of money and inflation rates; and (b) the total warehouse capacity to store the components, the total budget to purchase the components and the truck capacity are subject to constraints. The primary goal in this study is to find the optimal order quantity of the components for each subsystem in each period such that the total inventory costs are minimized and the system reliability is maximized, concurrently. A controlled elitism non-dominated ranked genetic algorithm (CE-NRGA), a NSGA-II, and a multi-objective particle swarm optimization (MOPSO) are presented to solve the proposed SPIRAP. A series of numerical examples are used to demonstrate the applicability and exhibit the efficacy of the procedures and algorithms. The results reveal that the CE-NRGA outperforms both NSGA-II and MOPSO.

1. Introduction

The implementation of redundancy plays an important role in elevating the reliability of a system. The redundancy allocation problem (RAP) engages the selection process of components and a series, parallel or series-parallel design configuration to simultaneously optimize some objective functions, such as system reliability, cost and weight, given certain design constraints (Zhang & Chen, 2016). RAP is crucial to design any modern complex systems such as safety systems, electrical power systems, transportation systems, satellite systems, and telecommunication systems with very strict reliability requirements (Wang & Xu, 2009). In the recent decade RAP has been common issue for investigation among the researchers. Salmasnia, Ameri, and Niaki (2016) modeled a series-parallel RAP in which the system reliability was maximized and the total cost became minimized. An electromagnetism-like mechanism meta-heuristic algorithm was developed by Teimouri, Zaretalab, Niaki, and Sharifi (2016) to solve a RAP. Kong, Gao, Ouyang, and Li (2015) introduced a RAP with multiple strategy choices at which both active redundancy and cold standby redundancy could be considered. They used a particle swarm optimization to optimize their problem. Mousavi, Alikar, Niaki, and Bahreininejad (2015b) formulated a multi-objective multi-state RAP in a series-parallel system in a fuzzy environment where the component types in each subsystem were identical. Mousavi, Alikar, and Niaki (2015) Optimized a fuzzy series-parallel RAP in which all types of the components in each subsystem were homogenous. A fruit fly optimization population-based meta-heuristic algorithm was derived to solve the problem.

Supplying the products and handling the products are the most important factors in today's businesses to reduce the total cost which is an essential goal for most companies. The companies try

* Corresponding author.
E-mail addresses: Najme.alikar@gmail.com (N. Alikar), Mousavi.mohsen8@gmail.com, mousavi@qiau.ac.ir (S.M. Mousavi), r_ariffin@um.edu.my (R.A.R. Ghazilla), tavana@lasalle.edu (M. Tavana), olugu@ucsiuniversity.edu.my (E.U. Olugu).

http://dx.doi.org/10.1016/j.cie.2016.12.006
0360-8352/© 2016 Published by Elsevier Ltd.

to utilize the best methods and approaches to reach this purpose given the constraints of real world applications. Therefore, more attention should be given by researchers to studying the inventory management aspects of business systems. In RAP the authors have focused more on the installation and the selection process of the components on the system and there is only a few works in the literature studying the production or preparing process of the components. Sadeghi, Sadeghi, and Niaki (2014) addressed a two-objective vendor managed inventory and redundancy allocation problem in a vendor-retailer supply chain problem. The aim of the problem was to obtain the optimal number of machines working in series to manufacture a single product. Xie, Liao, and Jin (2014) solved the system reliability of a repairable k-out-of-n RAP where the spares inventory in addition to redundancy allocation of components were taken into account.

In this work the costs imposed on the system are calculated under the time value of money and inflation rates which are two important factors associated with costs. Dey, Mondal, and Maiti (2008) developed an inventory problem for deteriorating items under inflation and time value of money at which two storages i.e. the one owned by the company and the rented one were available to store the items. A production repairing inventory control problem under inflation and the time value of money in a fuzzy environment was proposed by Mondal, Maity, Maiti, and Maiti (2013) where some parts of the spoiled items were repaired to be sold out. Mousavi, Hajipour, Niaki, and Alikar (2013) considered a multi-item multi-period inventory control problem in which the shortages were not allowed and both the time value of money and inflation rates were investigated to compute the total inventory costs. They used the genetic algorithm (GA), the simulated annealing (SA) algorithm and the branch and bound method to solve their problem. Wu, Skouri, Teng, and Hu (2016) modeled a seasonal/-fashionable inventory control problem under the time value of money where the items were deteriorating during the time. Yadav, Singh, and Kumari (2015) presented an inventory control problem in which the buyers could pay for the items at the end of period. The inflation rate, the deterioration rate and the delay in payment were considered in their model. Pareek and Dhaka (2014) formulated an inventory control problem for deteriorating products where shortages were not allowed, the deterioration rates were fuzzy values and the inflation and time value of money were calculated in modeling the total costs. An inventory control system was proposed by Duari and Chakraborti (2015) for deteriorating items with an exponentially increasing demand rate where shortages were allowed with inflation discount rates and delay in payment.

There are a variety of the related works on the inventory control problem recently in the literature (Cárdenas-Barrón, Sarkar, & Treviño-Garza, 2013; Kumar & Kumar, 2016; Sarkar, 2012, 2013, 2016; Sarkar & Majumder, 2013; Sarkar, Sana, & Chaudhuri, 2010; Sarkar & Saren, 2015; Sarkar, Saren, & Cárdenas-Barrón, 2015; Sarkar, Saren, & Wee, 2013; Sarkar & Sarkar, 2013; Sarkar, Sarkar, & Yun, 2016; Sett, Sarkar, & Goswami, 2012). Furthermore, Mousavi, Sadeghi, Niaki, and Tavana (2016) formulated a bi-objective multiple items multiple period inventory control problem in which shortages were allowed. The aim was to find out the optimal order quantity of items so that the total inventory costs and the total storage space were minimized simultaneously. The model was optimized using the three algorithms of MOPSO, NSGA-II and NRGA where the results were in the favor of MOPSO. Moreover, there are also some supply chain works related to this research performed recently in the literature (Pasandideh & Asadi, 2016; Pasandideh, Niaki, & Maleki, 2015). In addition, a number of works have recently considered the economic production order quantity problem (Jawla & Singh, 2016; Saxena, Singh, & Sangal, 2016; Shukla, Tripathi, & Sang, 2016).

Meta-heuristic algorithms have been common for solving the complex problems which have been hard to be optimized by exact solution methods. Multi-objective problems are not exempted from this complexity while many of researchers have implemented the multi-objective version of meta-heuristic algorithms to solve their problems. While the RAP is shown to be strongly NP-hard, a multi-objective meta-heuristic algorithm called the controlled elitism non-dominated ranked genetic algorithm (CE-NRGA) is developed to find the Pareto solutions of the problem under investigation. This algorithm is an elite version of NRGA has been used in many works in the literature. Pasandideh, Niaki, and Asadi (2015) addressed a multi-product multi-period inventory problem for a three-echelon supply chain problem in which a non-dominated sorting genetic algorithm (NSGA-II) and a NRGA were applied to solve their problem. Jalali, Seifbarghy, Sadeghi, and Ahmadi (2015) utilized a NRGA, a multi-objective biogeography-based optimization algorithm and a multi-objective simulation annealing algorithm to optimize a stochastic multi-facility location allocation-supply chain problem. A multi-objective vendor managed inventory was modeled by Sadeghi and Niaki (2015) for a single vendor multiple retailers supply chain problem in which NRGA and NSGA-II were used to find the Pareto solutions of the problem. Shahsavar, Najafi, and Niaki (2015) applied a NRGA to solve a multi-objective project scheduling problem with a triple-objective including (i) the minimization of the makespan, (ii) the minimization of the total cost associated with the resources, and (iii) the minimization of the variability in resources usage. Govindan, Jafarian, Khodaverdi, and Devika (2014) presented a multi-objective facility location problem for a two-echelon multiple vehicles routing problem by integrating sustainability in decision-making, on distribution in a perishable food supply chain network. NRGA, NSGA-II, and multi-objective particle swarm optimization (MOPSO) algorithms were implemented to solve the problem. Table 1 shows the recent literature most related to the current study.

The main novelties of the current work are: (i) considering a multi-components multi-period inventory control system for a series-parallel RAP, (ii) calculating the total inventory costs under the time value of money and inflation and (iii) deriving an elite version of NRGA called CE-NRGA, a NSGA-II and MOPSO to solve the problem.

The rest of the paper is organized as follows. In Section 2, we define the parameters, notations and decisions variables applied to formulate the problem. We also outline the proposed problem in detail along with a flowchart in this section. In Section 3, we describe and formulate the bi-objective SPIRAP problem. In Section 4, we present the multi-objective meta-heuristic algorithms proposed in this study. Some numerical examples and the analytical comparisons of the algorithms are generated in Section 5. Finally, in Section 6, we present our conclusion and future research directions.

2. The problem parameters and definition

In this section, we first define the parameters, notation and decision variables then outline the problem.

2.1. Notations

The indices, notations, parameters and the decision variables used to formulate the SPIRAP are described below:

Table 1
Recent literature most related to the current study.

Reference	Inventory costs	System reliability	Inflation & Time value of money	Discount policy	Design of experiment	Solving methodology
Dey et al. (2008)	Deteriorating items	–	Both	–	–	FEMGA
Mondal et al. (2013)	Production repairing inventory	–	–	–	–	Gradient method
Mousavi et al. (2013)	Multi-item multi period	–	Both	AUD, IQD	Taguchi	GA and SA
Sarkar (2013)	Deteriorating production-inventory	–	–	–	–	Probabilistic deterioration function
Pareek and Dhaka (2014)	Deteriorating inventory	–	Inflation	Fixed discount	–	DCF approach
Sadeghi et al. (2014)	Production-inventory machines	Series machines	–	AUD, IQD	Taguchi	NSGA-II, NRGA
Xie et al. (2014)	Spares logistics	Repairable Series system	–	–	–	Branch and bound
Duari and Chakraborti (2015)	Pricing and deteriorating inventory	–	Both	–	–	Weibull distribution
Kong et al. (2015)	–	Series	–	–	–	PSO
Mousavi et al. (2015)	–	Series-parallel	–	AUD, IQD	Taguchi	Fruit fly optimization
Mousavi et al. (2015b)	–	Series-parallel	–	AUD, IQD	Taguchi	CE-NRGA, NSGA-II
Sadeghi and Niaki (2015)	Fuzzy bi-objective VMI	–	–	–	Taguchi	NSGA-II, NRGA
Yadav et al. (2015)	Trade credit inventory	–	Inflation	–	–	Fuzzy method
Mousavi et al. (2016)	Multi-objective seasonal inventory	–	Inflation	AUD	Taguchi	NSGA-II, MOPSO, NRGA
Pasandideh and Asadi (2016)	Bi-objective multi-periodic supply chain network	–	–	–	–	SA, GA, ICA
Salmasnia et al. (2016)	–	Series-parallel	–	–	–	Loss function
Teimouri et al. (2016)	–	Series	–	–	–	Electromagnetism-like mechanism
Wu et al. (2016)	Seasonal deteriorating inventory	–	Time value of money	Fixed discount	–	Fuzzy method
Current paper	Bi-Objective Multi-Period Inventory	Series-Parallel RAP	Both	AUD, IQD	Taguchi	NSGA-II, CE-NRGA, MOPSO

Note: all-unit discount (AUD); controlled elitism (CE); fast and elitist multi-objective genetic algorithm (FEMGA); genetic algorithm (GA); incremental quantity discount (IQD); multi-objective particle swarm optimization (MOPSO); non-dominated ranked genetic algorithm (NRGA); non-dominated sorting genetic algorithm (NSGA-II); particle swarm optimization (PSO); and simulated annealing (SA).

Indices	
$i = 1, 2, \ldots, I$	Index for the components (products)
$j = 1, 2, \ldots, n_s$	Index for the subsystems ($s = 1, 2, \ldots, S$)
$t = 0, 1, \ldots, T$	Index for the time periods
$k = 1, 2, \ldots, K$	Index for the price break-points
Notation	
ho_{ijt}	Inventory holding cost per unit of ith component for subsystem j in period t
a_{ijt}	Ordering cost (transportation cost) per unit of ith component for subsystem j in period t
p_{ijtk}	Purchasing cost per unit of ith component for subsystem j at kth price break-point in period t
f_{ijt}	Required warehouse space to store per unit of ith component for subsystem j in period t
F	Available warehouse capacity
S	Total number of subsystems
T_t	Total time elapsed up to and including the tth replenishment cycle of components
λ_{ijtk}	kth price break-point for purchasing ith component for subsystem j in period t ($\lambda_{ijt1} = 0$)
l_{ij}	Reliability of component i purchased for subsystem j
w_{ij}	Weight of component i purchased for subsystem j
R	System reliability
d_{ijt}	Demand quantity of component i purchased for subsystem j in period t
E_t	Capacity of the transportation vehicle in period t
W	Total system weight
M	Upper bound for each order quantity
r	Interest rate representing TVM ($/$/year)
b	Constant annual inflation rate ($/$/year)
B	Net discount rate of inflation ($B = r - b$)
$A_i(t)$	Inventory level of component i in period t
Decision variables	
q_{ijt}	Ordering quantity of ith component purchased for subsystem j in period t ($q_{ijT} = 0$)
y_{ijtk}	A binary variable that is set to 1 if component i is purchased for subsystem j at kth price break-point in period t, and set to 0 otherwise
μ_{ijt}	A binary variable that is set to 1 if component i is ordered for subsystem j in period t, and set to 0 otherwise
x_{ijt}	Initial (remained) positive inventory of ith component purchased for subsystem j in period t ($x_{ij1} = 0$)

2.2. Problem formulation

In the current study, a mixed-integer nonlinear mathematical programming is considered for a multi-component multi-period inventory control problem in a binary-state series-parallel RAP. The components are manufactured in different types and periods and stored in a warehouse with a specific capacity wherein the particular spaces for each component exist. The components are purchased under all-unit discount (AUD) and incremental quantity discount (IQD) with an available budget to be installed on a series-parallel system in which subsystems are designed in series and the components are installed on each subsystem in parallel. The series-parallel system can carry a specific total weight with regards to the weight of the components installed on each subsystem. Once an order occurs in each period the components are delivered by the vehicles which are different than each other in different periods because of the capacity due to renting these vehicles from different transportation companies with different rental costs. The objective is to find the optimal order quantity of each component for each subsystem in each period so that the total inventory costs are minimized and the system reliability is maximized. In order to solve the proposed mathematical model, an elite version of NRGA i.e. CE-NRGA is utilized to find the optimal Pareto fronts where a NSGA-II and a MOPSO are also used to validate the performance of CE-NRGA as well. A Taguchi analysis is derived on the algorithms to set the optimal levels of these algorithms' parameters. A real world example is to consider the case of the engines installed on each wing of an airplane as a subsystem. The components (engines) of each wing work in parallel, meaning that the engine subsystem operates if at least one of the engines of a wing operates.

Fig. 1 shows some scenarios imposed on the inventory problem. In the first period of Fig. 1, the order quantity (q_{ij1}) arrives and is consumed by customers with the demand rate of d_{ij1} during the period. As a result, the remaining inventory in the first period (x_{ij2}) plus q_{ij2} are stored for the next period until the demand rate d_{ij2} is met. The description for the remaining periods is the same as the first period. Also, the series-parallel system configuration studied in this work is represented by Fig. 2.

In Fig. 2, the components with different types are installed on each subsystem in parallel while the subsystems are connected to each other in series. A flowchart of the definition of the problem is shown by Fig. 3. In Fig. 3, first the different types of the components are purchased from the related suppliers and then they are sent to the storage according to the scenarios mentioned in Fig. 1. At the end, components are installed on the subsystems based on the requested demand.

3. The bi-objective SPIRAP

The problem model is formulated under the following assumptions:

- The planning horizon is finite and known which includes T periods. It means the number of periods is limited and equal to T periods.
- The components are purchased under AUD and IQD and then installed on a series-parallel system. In other words, the suppliers sell some of the components under the AUD policy where the rest of the components are sold under the IQD policy to encourage the customers to purchase more.
- The components are delivered with the vehicles with different capacities in different periods. In fact, in each period different suppliers transfer the components with different trucks in terms of their capacity.
- The lead time is considered to be zero. It means that the duration time from ordering the components until they reach the storage is negligible.
- Order quantity is limited for each component in each period due to some production limitation. The suppliers are not able to produce an infinite amount of components due to their production capacity.

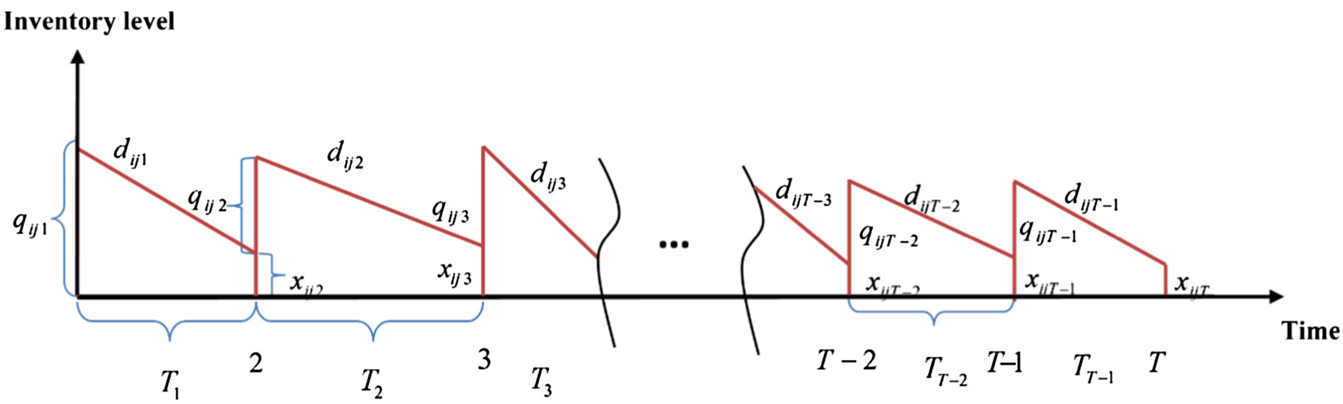


Fig. 1. A presentation of some scenarios of the proposed inventory problem.

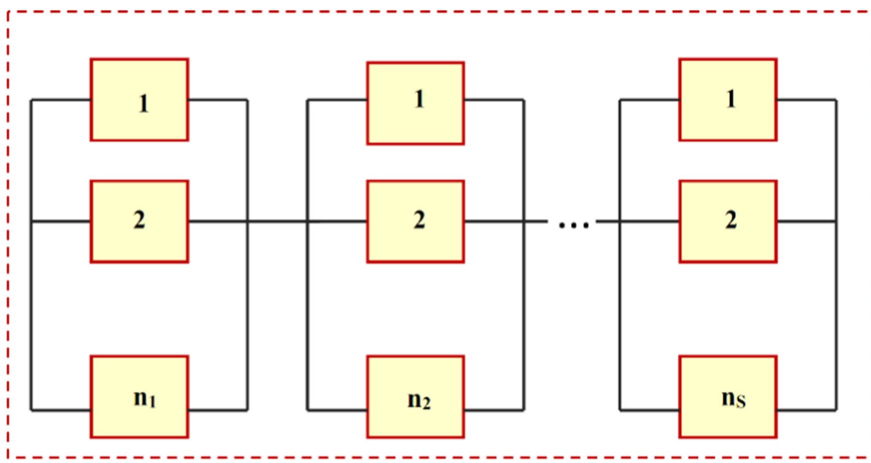


Fig. 2. A configuration of the designed system.

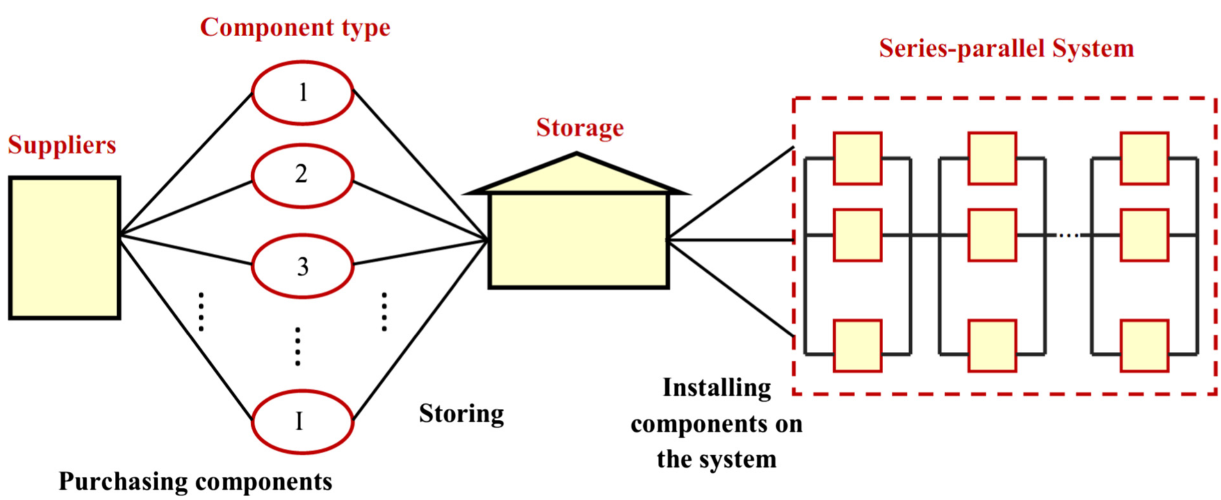


Fig. 3. The flowchart of the definition of the problem.

- There is no supply constraint for the components. This means that there are plenty of suppliers to supply the components.
- Failed components are not repairable. In other words, the broken down components are useless and unrepairable.
- Reliability, weight, and cost of all the components are deterministic and known. These attributes are usually labeled on the components.
- The number of subsystems is fixed. Each system has the pre-specified number of subsystems.
- The beginning inventory level at the first period is zero ($x_{ij1} = 0$) and also the order quantity in the last period is assumed to be zero ($q_{ijT} = 0$). This means there is no stock in the first storage while no order is made in the end period.

- Shortages are not allowed. In other words, all customers' demands are satisfied during each period.

The objectives of the developed SPIRAP are formulated as follows.

3.1. The objectives

The first objective is minimizing the total inventory costs including ordering cost (Oc), holding cost (Hc) and purchasing cost (Pc). The ordering cost comprises transportation and labor costs is modeled as follows:

$$\sum_{i=1}^{I}\sum_{j=1}^{n_s}\sum_{t=1}^{T-1} q_{ijt}a_{ijt}\mu_{ijt} \tag{1}$$

The total ordering cost under the time value of money and inflation is calculated as:

$$Oc = \sum_{i=1}^{I}\sum_{j=1}^{n_s}\sum_{t=1}^{T-1} q_{ijt}a_{ijt}\mu_{ijt}e^{-BTt} \tag{2}$$

The ordering cost includes all the costs carried once a component is ordered from the supplier while entering the storage. All costs as well as the labors for loading and uploading the products, negotiations for making an order, the cost related to transportation, etc. are included in this cost.

If $A_i(T_t) = x_{ijt} + q_{ijt}$ and $x_{ijt+1} = x_{ijt} + q_{ijt} - d_{ijt}T_t$ the holding cost is calculated using the aggregation of the areas of the trapezoid in Fig. 1 as follows:

$$ho_{ijt}\int_{t}^{t+1} A_i(t)e^{-Bt}dt \tag{3}$$

After simplification of Eq. (3), the total holding cost under the time value of money and inflation is computed as:

$$Hc = \sum_{i=1}^{I}\sum_{j=1}^{n_s}\sum_{t=1}^{T-1} ho_{ijt}e^{-BT_t}\left(\frac{e^{-BT_{t+1}} - e^{-BT_t}}{B}d_{ijt}\left(\frac{1}{B} - T_j - \left(\frac{x_{ijt} + q_{ijt}}{d_{ijt}}\right)\right.\right.$$
$$\left.\left. + \left(\frac{T_{j+1}e^{-BT_{t+1}} - T_j e^{-BT_t}}{e^{-BT_{t+1}} - e^{-BT_t}}\right)\right)\right) \tag{4}$$

This cost consists of all the costs associated with the storage such as the storage cost per unit of product in each period, the transportation cost, the labor cost, the rental cost of the warehouse, and the maintenance cost.

In order to motivate the customers to purchase more products, the suppliers encourage the owner of the system to buy all the components from them by using the AUD and IQD discount policies. Therefore, some of the components come up with the AUD policy and the rest of them with the IQD policy.

The purchasing cost under the AUD policy, the time value of money and inflation is formulated as:

$$Pca = \sum_{i=1}^{I}\sum_{j=1}^{n_s}\sum_{t=1}^{T-1}\sum_{k=1}^{K} q_{ijt}p_{ijtk}y_{ijtk}e^{-BT_t} \tag{5}$$

Also, the purchasing cost under the IQD policy, the time value of money and inflation is as:

$$Pci = \sum_{i=1}^{I}\sum_{j=1}^{n_s}\sum_{t=1}^{T-1}\sum_{k=1}^{K} y_{ijtk}p_{ijik}(q_{ijt} - \lambda_{ijtk})e^{-BT_t}$$
$$+ \sum_{i=1}^{I}\sum_{j=1}^{n_s}\sum_{t=1}^{T-1} y_{ijtK}p_{ijiK}(q_{ijt} - \lambda_{ijtK})e^{-BT_t} \tag{6}$$

Therefore, the total purchasing cost is formulated as Eq. (7).

$$Pc = Pca + Pci \tag{7}$$

Eq. (8) expresses the total inventory cost of the proposed problem as the first objective function.

$$TC = Oc + Hc + Pca + Pci \tag{8}$$

According to Fig. 2, the second objective i.e. maximizing the system reliability is calculated as Eq. (9).

$$R = \prod_{i=1}^{I}\left(1 - \prod_{j=1}^{n_s}(1 - l_{ij})^{\sum_{t=1}^{T-1} q_{ijt}}\right) \tag{9}$$

3.2. The constraints

The problem is modeled under some constraints to make the model more real as follows. The total storage capacity is restricted while there are particular places for each component in each period shown as:

$$\sum_{i=1}^{I}\sum_{j=1}^{n_s}\sum_{t=1}^{T}(q_{ijt} + x_{ijt})f_{ijt} \leqslant F \tag{10}$$

Eq. (10) depicts that the total order quantity of all the components in each period in addition to the components remained from the previous period should not exceed the total storage capacity.

The total budget allocated to purchase the components is limited. The budget available for purchasing the components under the time value of money, inflation and both the AUD and IQD policies are formulated as Eq. (11).

$$\left(\sum_{i=1}^{I}\sum_{j=1}^{n_s}\sum_{t=1}^{T-1}\sum_{k=1}^{K} p_{ijtk}y_{ijtk}(q_{ijt} - \lambda_{ijtk}) + \sum_{i=1}^{I}\sum_{j=1}^{n_s}\sum_{t=1}^{T-1} y_{ijtK}p_{ijtK}(q_{ijt} - \lambda_{ijtK})\right)e^{-BT_t}$$
$$+ \sum_{i=1}^{I}\sum_{j=1}^{n_s}\sum_{t=1}^{T-1} q_{ijt}a_{ijt}\mu_{ijt}e^{-BTt} \leqslant Tb \tag{11}$$

The total weight of the series-parallel system cannot exceed W which is formulated as:

$$\sum_{i=1}^{I}\sum_{j=1}^{n_s}\sum_{t=1}^{T-1} q_{ijt}\mu_{ijt}w_{ij} \leqslant W \tag{12}$$

where the weight of the total components ordered to be installed on the series-parallel system should not exceed the total system weight. To deliver the components ordered to be installed on the system, the owner of the system has to rent the vehicles can be different than each other in each period in terms of their capacity to choose the cheaper ones. This limitation is shown by Eq. (13).

$$\sum_{i=1}^{I}\sum_{j=1}^{n_s} q_{ijt}\mu_{ijt} \leqslant E_t \tag{13}$$

Eq. (13) shows that all the components ordered from the supplier are transferred by trucks with a limited capacity in each period. Also, due to some production restrictions, the order quantity of each component is limited which is formulated as:

$$q_{ijt} \leqslant M \tag{14}$$

Finally, since at most one order can be placed for a component for a subsystem in a period and it can be purchased at one price break point, we have

$$\sum_{k=1}^{K} y_{ijtk} = 1 \tag{15}$$

In other words, each component ordered for each subsystem in each period should be at one price break point. Therefore, the proposed mixed-integer nonlinear mathematical model of SPIRAP is:

automatically initialized in accordance with the values determined for q_{ijt}. Fig. 4 depicts a chromosome structure of the proposed CE-NRGA including all the decision variables mentioned in Eq. (16).

$$
\begin{aligned}
&\begin{cases} MinTC = \sum_{i=1}^{I}\sum_{j=1}^{n_s}\sum_{t=1}^{T-1} q_{ijt}a_{ijt}\mu_{ijt}e^{-BT_t} + \sum_{i=1}^{I}\sum_{j=1}^{n_s}\sum_{t=1}^{T-1} ho_{ijt}e^{-BT_t}\left(\frac{e^{-BT_{t+1}}-e^{-BT_t}}{B}d_{ijt}\left(\frac{1}{B}-T_j-\left(\frac{x_{ijt}+q_{ijt}}{d_{ijt}}\right)+\left(\frac{T_{j+1}e^{-BT_{t+1}}-T_je^{-BT_t}}{e^{-BT_{t+1}}-e^{-BT_t}}\right)\right)\right) \\ +\left(\sum_{i=1}^{I}\sum_{j=1}^{n_s}\sum_{t=1}^{T-1}\sum_{k=1}^{K} q_{ijt}p_{ijtk}y_{ijtk}e^{-BT_t} + \sum_{i=1}^{I}\sum_{j=1}^{n_s}\sum_{t=1}^{T-1}\sum_{k=1}^{K} p_{ijtk}(q_{ijt}-\lambda_{ijtk})e^{-BT_t} + \sum_{i=1}^{I}\sum_{j=1}^{n_s}\sum_{t=1}^{T-1} y_{ijtK}p_{ijtK}(q_{ijt}-\lambda_{ijtK})e^{-BT_t}\right) \end{cases} \\
&\text{Max } R = \prod_{i=1}^{I}\left(1-\prod_{j=1}^{n_s}(1-l_{ij})^{\sum_{t=1}^{T-1} q_{ijt}}\right) \\
&\text{Subject to :} \\
&x_{ijt+1} = x_{ijt} + q_{ijt} - d_{ijt}T_t \\
&\sum_{i=1}^{I}\sum_{j=1}^{n_s}\sum_{t=1}^{T}(q_{ijt}+x_{ijt})s_{ijt} \leqslant F \\
&\left(\sum_{i=1}^{I}\sum_{j=1}^{n_s}\sum_{t=1}^{T-1}\sum_{k=1}^{K} p_{ijtk}(q_{ijt}-\lambda_{ijtk}) + \sum_{i=1}^{I}\sum_{j=1}^{n_s}\sum_{t=1}^{T-1} y_{ijtK}p_{ijtK}(q_{ijt}-\lambda_{ijtK})\right)e^{-BT_t} \leqslant Tb \\
&\sum_{i=1}^{I}\sum_{j=1}^{n_s}\sum_{t=1}^{T-1} q_{ijt}\mu_{ijt}w_{ij} \leqslant W \\
&\sum_{i=1}^{I}\sum_{j=1}^{n_s} q_{ijt}\mu_{ijt} \leqslant E_t \\
&q_{ijt} \leqslant M \\
&\sum_{k=1}^{K} y_{ijtk} = 1 \\
&x_{ijt} \geqslant 0; q_{ijt} \geqslant 0; q_{ijt}, x_{ijt} \in Z\}
\end{aligned}
\tag{16}
$$

4. Solving methodologies

Due to complexity of SPIRAP under investigation, some well-known meta-heuristic algorithms are applied to solve the problem. To do that, a CE-NRGA algorithm is applied to obtain the optimal order quantities of the components purchased for each subsystem in each period. A NSGA-II and a MOPSO algorithm are also utilized to validate the results of the CE-NRGA. According to the related works done in the literature reviewed above, these three meta-heuristic algorithms are the most appropriate algorithms for solving the problem formulated in Eq. (16).

4.1. The proposed CE-NRGA

The following steps are included in the proposed CE-NRGA:

Initialize the parameters and chromosomes: the number of population (np), the crossover probability (P_c), the mutation probability (P_m) and the number of generation (ng) are the CE-NRGA parameters assumed in this step. Furthermore, the chromosomes of the population np are initialized randomly in the range $[0, M]$. The rest of decision variables as well as y_{ijtk}, μ_{ijt} and x_{ijt} are

Fig. 5 shows a presentation of a population generated by CE-NRGA where the number of the population is np.

Evaluate the chromosomes: All the chromosomes initialized in the prior step are evaluated using both objectives shown in Eqs. (8) and (9). The infeasible solutions are penalized by the following functions:

$$
\begin{cases} U_1(x) + (u(x) - Q)^{\delta} \\ U_2(x)/(u(x) - Q)^{\eta} \end{cases} \tag{17}
$$

where $U_1(x)$ and $U_2(x)$ are the first (Min-type) and second objectives (Max-type) in Eq. (16) respectively, $u(x)$ and Q is the left and right hand sides of constraint $u(x) \leqslant Q$ and δ and η are the penalty coefficients respectively. According to Eq. (17), if a constraint exceeds the upper bound, a coefficient of the extra amount will be added to the Min-type objective while the Max-type objective is penalized as much as the objective value divided by this extra amount.

Fast non-dominated sorting process: In this step, the np populations that were generated in the previous steps are compared and sorted. To do that, all solutions in the first non-dominated front are first found where the solutions are chosen using the concept of

q_{111}	q_{112}	…	q_{ijt}	…	q_{IJT-1}	y_{1111}	…	y_{IJTK}	μ_{111}	…	μ_{IJT}	x_{111}	…	x_{IJT}

Fig. 4. The chromosome representation.

1	q_{111}	q_{112}	…	q_{ijt}	…	q_{IJT-1}	y_{1111}	…	y_{IJTK}	μ_{111}	…	μ_{IJT}	x_{111}	…	x_{IJT}
2	q_{111}	q_{112}	…	q_{ijt}	…	q_{IJT-1}	y_{1111}	…	y_{IJTK}	μ_{111}	…	μ_{IJT}	x_{111}	…	x_{IJT}
⋮															
np	q_{111}	q_{112}	…	q_{ijt}	…	q_{IJT-1}	y_{1111}	…	y_{IJTK}	μ_{111}	…	μ_{IJT}	x_{111}	…	x_{IJT}

Fig. 5. A presentation of the population generated by CE-NRGA.

domination, in which a solution Y_m is said to dominate solution Y_n, if $\forall e \in \{1,2\}$ we have $f_e(Y_m) \leqslant f_e(Y_n)$ for $e \in \{1,2\}$ such that $f_e(Y_m) < f_e(Y_n)$. In this case, we say Y_m is the non-dominated solution within the solution set $\{Y_m, Y_n\}$. Otherwise, it is not. In addition, to find the solutions in the next non-dominated front, the solutions of the previous fronts are disregarded temporarily. This procedure is repeated until all solutions are set into fronts.

Crowding distance: After sorting the populations based on their ranks, a measure called the crowding distance (CD) is defined to evaluate solution fronts of populations in terms of the relative density of individual solutions. To do this, let V_b; $(b = 1, 2, \ldots, L)$ be the *b-th* front, where V_b is the number of non-dominated solutions in the particular front b, and f_1 and f_2 are the objective functions. Besides, let d_{z_t}; $(z_b = 1, 2, \ldots, V_b)$ be the value of crowding distance for the solution m_t. Then, the crowding distance is obtained using the steps proposed in Fig. 6. At the end, the populations are sorted based on their ranks and crowding distances.

Selection process: In this study, in order to select the chromosomes, the roulette wheel method is utilized. The roulette wheel process is performed using the following formulas.

$$P_f = \frac{2 \times Rank_f}{NF \times (NF + 1)}; \quad f = 1, \ldots, NF \tag{18}$$

$$P_{fs} = \frac{2 \times Rank_{fs}}{NS_f \times (NS_f + 1)}; \quad f = 1, \ldots, NF, \quad s = 1, \ldots, NS \tag{19}$$

where NF and NS_f are the number of fronts and the number of solutions in front f, respectively. Also, P_f is the selection probability of fronts, and P_{fs} is the selection probability of solutions. Eq. (18) ensures that a front with highest rank has the highest probability to be selected. Similarly, based on Eq. (19), solutions with more crowding distance are assigned a higher selection probability. The roulette wheel selection is iterated until a desired number of solutions are selected. At the end, the algorithm stops when a predetermined number of iterations is reached.

Do crossover operator: Let P_C be the crossover probability and r_1 be a uniform random number between zero and one. For each of the *np* populations, if r_1 is less than P_C, select two parent chromosomes q_1 and q_2 randomly. Then, the crossover operator of the proposed CE-NRGA algorithm is performed based on the following equations.

$$q'_1 = Round(\upsilon q_1 + (1 - \upsilon) q_2) \tag{20}$$

$$q'_2 = Round((1 - \upsilon) q_1 + \upsilon q_2) \tag{21}$$

where q'_1 and q'_2 are offspring and υ is a uniform random number between zero and one.

Do mutation operator: The rest of chromosomes are selected with the probability P_m to enter the mutation operator. To perform the mutation process on a chromosome, a gene of a value of q_{ijkt} is selected randomly and changed in the range $[0, M]$.

Combine and update the population: First, evaluate the chromosomes obtained from the operators (P) and combine it with the chromosomes obtained in the first step (Q) to generate a population $(Q_t \cup P_t)$, which represents a combination of populations P and Q in generation t.

To propose diversity in the new population of size *np*, the maximum number of individuals allowed in the fth front, p_f, is generally given by

$$p_f = np \frac{1 - r}{1 - r^{nf}} r^{f-1} \tag{22}$$

where r is the reduction rate, a user-defined value, that is less than 1 and *nf* is the number of non-dominated fronts (Bharti, Maheshwari, & Sharma, 2012).

The policy used here allows the solutions from all non-dominated fronts to distribute in the population. If a particular front has more solutions than required (to fill up the *np*, then Eq. (22) is employed to limit the number of solutions which come from the relevant front. However, if the number of solutions (in a particular front) is less than the value of *np*, the difference is added to the maximum allowed solutions in the next front and so on. After filling up the *np* solutions in the new population, the same process is repeated over a number of generations and the Pareto-optimal solutions are obtained. Since the solutions from all non-dominated fronts co-exist in the population, the diversity is maintained and the solutions obtained are true optimal solutions (Bharti et al., 2012; Mousavi et al., 2015b). Fig. 7 depicts a graphical

1: Set $d_{z_b} = 0$ for $b = 1, 2, \ldots, L$; $z_b = 1, 2, \ldots, V_b$
2: Sort both objective functions in ascending order
3: The crowding distance for end solutions in each front($z_b = 1, V_b$) are $d_1 = d_{V_b} = \infty$
4: The crowding distance for the objective functions $f_{1,2}$ for $z_b = 2, \ldots, V_b - 1$ is calculated by $d_{z_b} = d_{z_b} + (f_{1,2} - f_{1,2})$

Fig. 6. A pseudo code to determine the crowding distance.

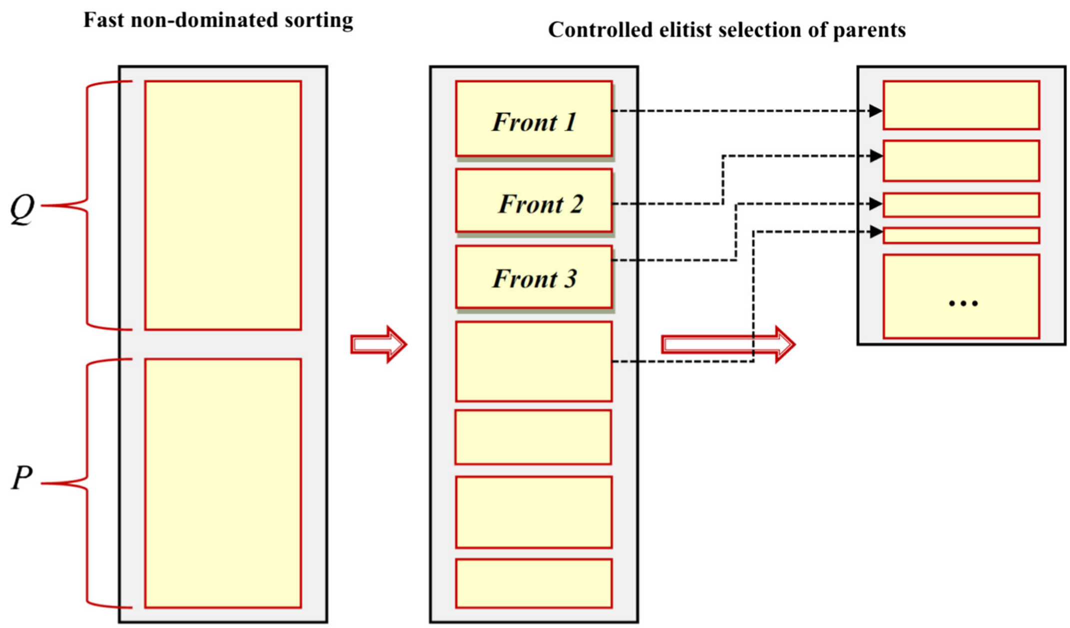


Fig. 7. The CE-NRGA procedure.

representation of the proposed CE-NRGA. In Fig. 7, in the fast non-dominated sorting stage the old and new populations P and Q are combined and then the chromosomes are allocated to the fronts while the crowding distance of the chromosomes is obtained in the second stage. Finally, a population of chromosomes is selected using the formula expressed by Eq. (22) for the next generation.

4.2. The proposed NSGA-II

The steps of the NSGA-II employed in this research are explained briefly as follows:

1. Initializing the parameters and chromosomes as well as the process done for CE-NRGA.
2. Evaluating the solutions which are the same as CE-NRGA.
3. Fast non-dominated sorting and crowding distance process are done exactly similar to CE-NRGA.
4. Selection process in which a two-chromosome tournament selection operator is used to find the best solutions for the next generation in which two chromosomes are selected randomly and then compared in terms of the front rank and crowding distance. The one with least front rank is chosen. In case of the front ranks being equal, the one with maximum crowding distance will be chosen.
5. Crossover and mutation operators are performed similar to the process explained in CE-NRGA.
6. Combining the populations at which both np solutions and new solutions (after updating the population according to their ranks and crowding distance) are combined to generate a population of size $(2 \times np)$ to ensure elitism. Then, $(2 \times np)$ popula-

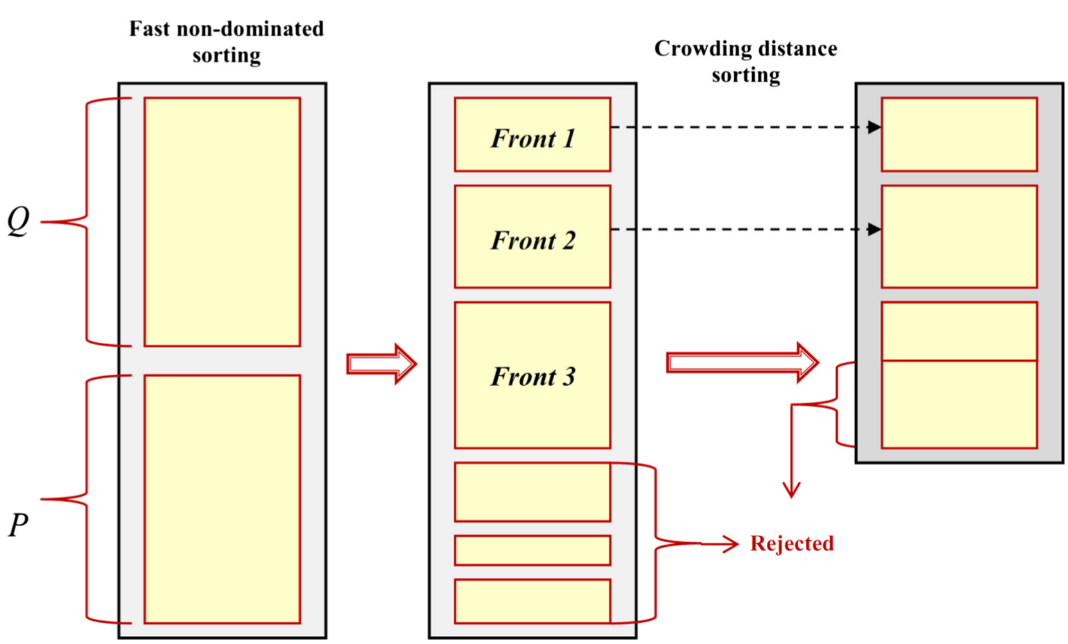


Fig. 8. The CE-NRGA procedure.

Table 2
The input data of the generated numerical examples.

Problem No.	(I, n_s, T)	d	ho	f	w	l	E_t	a	W	F	Tb	T	p	q
1	(3,3,3)	[1,4]	[1,8]	[1,5]	[3,8]	[0.03,0.15]	[500,700]	[1,5]	[500,800]	[500,700]	[500,700]	[1,2]	[1,5]	[1,10]
2	(1,3,3)	[1,3]	[1,8]	[1,5]	[3,8]	[0.03,0.15]	[200,300]	[1,5]	[100,200]	[200, 300]	[200,,300]	[1,2]	[1,5]	[1,5]
3	(2,2,2)	[1,4]	[1,8]	[1, 5]	[3,8]	[0. 2,0.5]	[150,200]	[1,5]	[100,200]	[150,200]	[150,200]	[1,2]	[1,5]	[1,5]
4	(3,2,2)	[1,4]	[1,8]	[1,5]	[3,8]	[0.4,0.8]	[250,300]	[1,5]	[250,300]	[250,300]	[250,300]	[1,2]	[1,5]	[1,5]
5	(3,3,2)	[1,4]	[1,8]	[1,5]	[3,8]	[0.4,0.8]	[250,300]	[1,5]	[250,300]	[250,300]	[250,300]	[1,2]	[1,5]	[1,5]
6	(4,1,2)	[1,4]	[1,8]	[1,5]	[3,8]	[0.15,0.45]	[300,400]	[1,5]	[300,400]	[300,400]	[300,400]	[1,2]	[1,5]	[1,5]
7	(4,2,2)	[1,4]	[1,8]	[1,5]	[3,8]	[0.25,0.55]	[300,500]	[1,5]	[300,500]	[300,500]	[300,500]	[1,2]	[1,5]	[1,5]
8	(4,2,3)	[1,4]	[1,8]	[1,5]	[3,8]	[0.25,0.55]	[300,500]	[1,5]	[300,500]	[300,500]	[300,500]	[1,2]	[1,5]	[1,5]
9	(4,3,3)	[1,4]	[1,8]	[1,5]	[3,8]	[0.1,0.25]	[800,1000]	[1,5]	[800,1000]	[800,1000]	[800,1000]	[1,2]	[1,5]	[1,5]
10	(4,4,4)	[1,4]	[1,8]	[1,5]	[3,8]	[0.1,0.25]	[1100,1300]	[1,5]	[1000,1200]	[1000,1200]	[1000,1200]	[1,2]	[1,5]	[1,5]
11	(5,3,3)	[1,4]	[1,8]	[1,5]	[3,8]	[0.1,0.25]	[1100,1300]	[1,5]	[1000,1200]	[1000,1200]	[1000,1200]	[1,2]	[1,5]	[1,5]
12	(5,5,5)	[1,4]	[1,8]	[1,5]	[3,8]	[0.1,0.25]	[1500,1800]	[1,5]	[1400,1600]	[1400,1600]	[1400,1600]	[1,2]	[1,5]	[1,5]
13	(6,4,4)	[1,4]	[1,8]	[1,5]	[3,8]	[0.1,0.25]	[1500,1800]	[1,5]	[1400,1600]	[1400,1600]	[1400,1600]	[1,2]	[1,5]	[1,5]
14	(6,6,6)	[1,4]	[1,8]	[1,5]	[3,8]	[0.1,0.25]	[1900,2200]	[1,5]	[1900,2100]	[1900,2100]	[1900,2100]	[1,2]	[1,5]	[1,5]
15	(7,7,7)	[1,4]	[1,8]	[1,5]	[3,8]	[0.1,0.25]	[2500,2800]	[1,5]	[2500,2800]	[2500,2800]	[2500,2800]	[1,2]	[1,5]	[1,5]

tions are evaluated and sorted based on their ranks and crowding distance. Finally, according to the sorted solutions, *np* solutions are selected as an initial population for the next generation. Fig. 8 demonstrates the approach taken in NSGA-II to combine the populations and to select *np* solutions for the next generation where the solutions in first fronts have the priority to be selected for the next generation.

The algorithm stops when a predetermined number of iterations is reached.

4.3. The MOPSO

The proposed MOPSO is briefly explained as follows:

- Initializing the parameters i.e. *np*, number of generation *ng*, and two parameters C_1 and C_2 and also articles randomly in the range.
- Evaluating the objectives modeled in Eq. (16) for each article of the population.
- Fast non-dominated sorting process wherein *np* articles are compared and ranked as well as the process explained in step 3 of the CE-NRGA.
- Crowding distance is also done the same as in step 4 of the CE-NRGA.
- The articles of the population are updated as the position and velocity, which are two variables in the MOPSO algorithm, and are initialized using Eqs. (23) and (24), respectively.

$$v^l_{n+1} = \omega \cdot v^l_n + C_1 \cdot u_1 \cdot (pBest^l_n - z^l_n) + C_2 \cdot u_2 \cdot (gBest_n - z^l_n) \quad (23)$$

$$z^l_{n+1} = z^l_n + \eta \cdot v^l_{n+1} \quad (24)$$

In Eq. (24), u_1 and u_2 are two numbers generated randomly in the interval of (0, 1), the coefficients C_1 and C_2 are the given acceleration constants towards *pBest* and *gBest*, respectively, and ω is the inertia weight where is expressed as Eq. (25) (Naka, Genji, Yura, & Fukuyama, 2001). Furthermore, $pBest^l_n$ and $gBest_n$ are the best fitness value for particle l until time n, ($n = 1, 2, \ldots, np$) and the best particle among all until time n, respectively.

$$\omega = \omega_{\max} - \frac{(\omega_{\max} - \omega_{\min})}{ng} \cdot n \quad (25)$$

In Eq. (25), *nog* is the maximum number of iterations and n is the current number of iteration. Shi and Eberhart (1999) and Naka et al. (2001) have claimed the best result will be obtained since $[\omega_{\min}, \omega_{\max}] = [0.4, 0.9]$.

- Combining the populations is done as in Step 8 mentioned in NSGA-II.

Table 3
The parameters of the algorithms and their levels.

Algorithm	Parameter	Low (1)	Medium (2)	High (3)
CE-NRGA	*Np* (*A*)	20	30	40
	Pc (*B*)	0.6	0.7	0.8
	Pm (*C*)	0.05	0.1	0.2
	Ng (*D*)	100	200	300
NSGA-II	*Np* (*A*)	20	30	40
	Pc (*B*)	0.6	0.65	0.75
	Pm (*C*)	0.05	0.1	0.2
	Ng (*D*)	100	300	400
MOPSO	*Np* (*A*)	40	60	80
	C1 (*B*)	1.5	2	2.5
	C2 (*C*)	1.5	2	2.5
	Ng (*D*)	400	500	600

Table 4
The obtained response values of CE-NRGA and NSGA-II.

Run order	Parameter levels				Response value	
	np	P_c	P_m	*ng*	CE-NRGA	NSGA-II
1	1	1	1	1	9.82657146	2.4425818
2	1	2	2	2	13.1061369	3.2166137
3	1	3	3	3	8.91801678	9.7222129
4	2	1	2	3	14.5691781	2.2194309
5	2	2	3	1	4.94753202	5.5990300
6	2	3	1	2	8.48109848	2.8011139
7	3	1	3	2	13.4993125	3.4846371
8	3	2	1	3	13.9708680	2.8315674
9	3	3	2	1	14.8876788	7.2266019

Table 5
The obtained response values of MOPSO.

Run order	Parameter levels				Response value
	C_1	C_2	*np*	*ng*	MOPSO
1	1	1	1	1	9.95241
2	1	2	2	2	3.37730
3	1	3	3	3	4.41263
4	2	1	2	3	7.89612
5	2	2	3	1	3.93310
6	2	3	1	2	5.43368
7	3	1	3	2	9.44769
8	3	2	1	3	4.15401
9	3	3	2	1	2.69597

Table 6
The optimal levels of the algorithms' parameters.

Algorithm	Parameter	Optimal level
CE-NRGA	*np*	30
	Pc	0.7
	Pm	0.2
	ng	100
NSGA-II	*np*	30
	Pc	0.6
	Pm	0.05
	ng	300
MOPSO	*np*	80
	C_1	2.5
	C_2	2.5
	ng	400

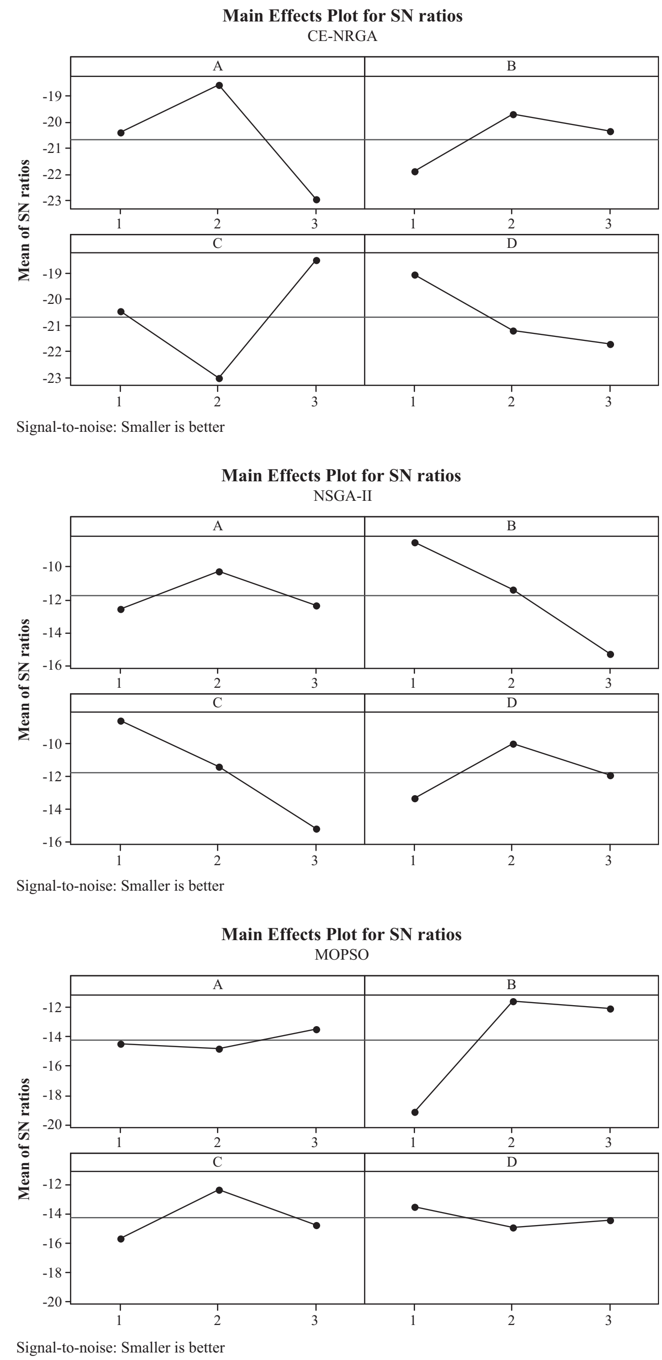


Fig. 9. Mean S/N ratios of the parameters of the proposed algorithms.

5. Results and discussion

This section represents the application of the proposed CE-NRGA along with the NSGA-II and MOPSO algorithms on some test problems. To accomplish so, first, some well-known multi-objective metrics are introduced. Then, the parameters of the algorithms are adjusted via the Taguchi method. Finally, the defined metrics are compared statistically and graphically.

5.1. The proposed multi-objective metrics

This work implements six metrics to make comparisons between the algorithms. These metrics are briefly explained as follows:

MID: This metric is the mean ideal distance introduced by Zitzler and Thiele (1998) that gauges the convergence rate of the Pareto fronts towards a specific point (0, 0).
NOS: Displays the number of non-dominated solutions.
MS: The maximum spread or diversity proposed by Zitzler and Thiele (1998) employed to measure the diagonal length of a hyper box that is formed by extreme function values observed in the Pareto curve.
SNS: It is the diversity measure of the Pareto archive solutions (Karimi, Zandieh, & Karamooz, 2010).
CT (Sec): This is a metric to measure the CPU time taken by running the algorithms.
SP: This metric calculates the closest distance of pairwise solutions in the set of solutions (Schott, 1995). In other words, the spacing metric measures the standard deviation of the distances among the solutions belonging to the Pareto front (Hajipour, Fattahi, Tavana, & Di Caprio, 2016; Zitzler & Thiele, 1998).

5.2. Numerical illustrations

In order to evaluate the proposed CE-NRGA, NSGA-II and MOPSO algorithms on the SPIRAP, some numerical examples are generated randomly while there is no benchmark in the literature used to assess the algorithms on the problem. In this study, 15 numerical examples are generated randomly with different number of components, subsystems and periods in the range $[1,7]$. The rest of the parameters involved in formulating the problem are generated randomly according to Table 2. In order to solve the numerical examples generated randomly, the interest rate (r) and inflation rate (b) values are considered to be 0.2 and 0.15, respectively.

5.3. The parameter tuning procedure

The parameters of meta-heuristic algorithms play an important role to achieve solutions with better qualities. Therefore, tuning the parameters using a design of experiments is an effective way to reach better results. The Taguchi method is one approach to calibrate the parameters. Since this method has received more attention in the literature recently, it is utilized in this study as well to set the parameters of the three algorithms (Mousavi et al., 2015; Mousavi, Alikar, Niaki, & Bahreininejad, 2015a, 2015b; Mousavi, Bahreininejad, Musa, & Yusof, 2014). A multi-objective version of the Taguchi method is defined as:

Table 7
Experimental results of different test problems obtained by CE-NRGA and NSGA-II.

Problem No.	CE-NRGA						NSGA-II					
	MID	NOS	MS	SP	SNS	CT (sec)	MID	NOS	MS	SP	SNS	CT (sec)
1	34700.90	20	34	86,350,045	13571.18	14.81204	51319.68	18	52	24,536,988	16856.39	24.56135
2	20427.89	14	44	41,569,850	9242.359	15.55318	59928.16	13	36	34,567,896	23444.89	21.43876
3	36680.65	13	43	76,980,050	5040.945	15.68972	57377.64	11	50	425,786,325	15659.91	24.10777
4	32952.51	10	47	156,980,045	8191.715	16.06494	56857.17	6	58	135,469,857	19031.48	24.32600
5	37286.41	14	22	76,582,200	5026.488	16.94644	50792.33	13	83	73,587,589	18466.15	22.04509
6	22857.26	16	23	98,553,020	8614.854	15.95958	54371.63	12	57	45,236,852	18103.08	23.07782
7	27919.75	15	27	95,500,500	14072.76	16.21237	45618.31	13	63	56,984,236	19000.32	21.26250
8	21729.60	17	35	86,450,012	11352.84	16.63901	56466.96	12	37	136,850,025	21174.97	21.84416
9	36666.32	15	45	358,972,005	12141.91	16.33359	51864.87	13	69	200,132,560	23722.28	24.48584
10	33913.50	18	32	135,682,202	7595.678	15.96771	56306.96	16	77	145,873,258	20761.26	23.08330
11	30413.35	14	33	88,699,218	13440.58	16.36229	53112.19	10	92	95,638,954	23068.57	24.40818
12	25195.18	13	45	32,556,984	14855.26	15.40642	48534.43	12	60	982,436,581	18734.24	21.75041
13	28558.01	16	23	233,565,002	12715.34	16.16357	48727.48	13	63	72,563,687	15690.84	21.03034
14	39460.87	17	25	58,997,530	13615.94	16.58055	59661.75	13	45	48,693,254	19572.75	21.66396
15	22829.10	15	21	754,233,690	13992.86	19.43500	55467.94	14	49	654,869,321	23846.84	24.71812
Ave.	30106.1	15.1333	33.26	158,778,156	10,898	16.0494	53760.5	12.6	59.4	208,881,825	19808.9	22.9202

Table 8
Experimental results of different test problems obtained from MOPSO.

Problem No.	MOPSO					
	MID	NOS	MS	SP	SNS	CT (sec)
1	64254.89	18	79	33,164,466	31580.44	34.33206
2	69177.62	6	76	21,354,602	31012.85	36.72862
3	59473.30	4	74	98,013,620	27856.97	32.11887
4	61718.01	4	67	125,980,310	29723.81	32.71135
5	79038.62	4	69	85,324,569	31299.49	32.34423
6	52880.24	12	64	20,302,655	27309.52	35.43016
7	55230.45	14	78	865,321,233	34384.33	34.15378
8	71464.50	11	73	64,589,856	29623.15	34.82747
9	76500.45	10	74	112,362,100	33941.97	33.54657
10	57781.57	12	75	113,256,465	29022.19	35.73251
11	75700.39	11	65	631,322,625	26067.91	36.41034
12	76165.98	10	70	74,565,420	31434.73	36.38742
13	67008.11	12	75	83,644,656	27960.46	32.67796
14	71570.36	13	71	36,549,800	26308.99	36.39700
15	74083.90	10	74	589,255,000	34489.36	34.58421
Ave.	67469.9	10.0667	72.2667	197,000,492	30134.4	34.5588

Table 9
The ANOVA for the metrics resulting from the three algorithms.

Metric	Response value	
	P-value	Result
MID	0.00009	H_0 is rejected
NOS	0.00089	H_0 is rejected
MS	0.00009	H_0 is rejected
SP	0.84442	H_0 is accepted
SNS	0.00013	H_0 is rejected
CT (*Sec*)	0.00008	H_0 is rejected

$$S/N = -10\log\left(\sum_{i=1}^{n}\frac{(MID/SNS)_i^2}{n}\right) \tag{26}$$

In Eq. (26), *MID*/*SNS* is the response in *i*th experiment and *n* represents the number of orthogonal arrays, based on which the experiments are performed.

5.4. Comparisons and discussions

In this subsection, the proposed algorithms are compared graphically and statistically as well. In order to establish the Tagu-

Table 10
The Pareto solution of the algorithms for Problem No. 1.

Sol. No.	CE-NRGA		NSGA-II		MOPSO	
	Cost	Reliability	Cost	Reliability	Cost	Reliability
1	1373	0.743	1473	0.777	1381	0.678
2	1447	0.771	1482	0.781	1354	0.662
3	1357	0.737	1442	0.767	1567	0.766
4	1379	0.746	1522	0.794	1594	0.773
5	1169	0.664	1364	0.73	1428	0.731
6	1210	0.671	1532	0.796	1474	0.734
7	1448	0.772	1425	0.761	1418	0.705
8	1366	0.739	1478	0.779	1389	0.691
9	1431	0.766	1506	0.786	1502	0.747
10	1469	0.781	1462	0.772	1514	0.75
11	1425	0.765	1378	0.739	1416	0.704
12	1488	0.782	1395	0.744	1527	0.754
13	1286	0.705	1384	0.744	1534	0.761
14	1355	0.732	1295	0.711	1487	0.742
15	1293	0.712	1403	0.748	1405	0.698
16	1222	0.678	1456	0.768	1546	0.765
17	1424	0.763	1565	0.799	1591	0.769
18	1252	0.693	1315	0.714	1364	0.671
19	1264	0.700	–	–	–	–
20	1392	0.754	–	–	–	–

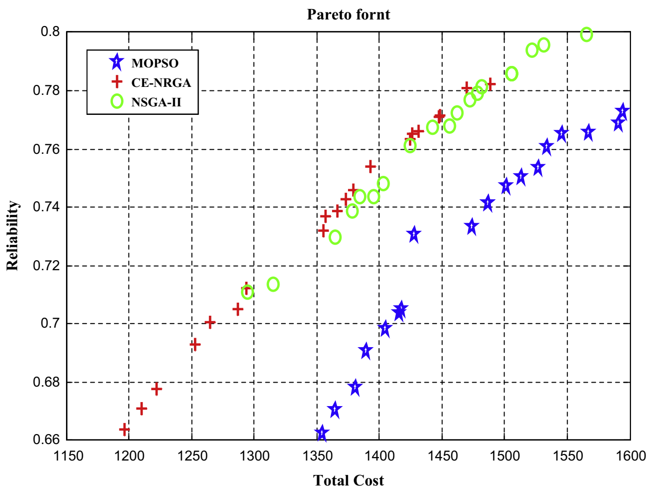


Fig. 10. The Pareto fronts obtained by the three algorithms for Problem No. 1.

Table 11
The Pareto solution of the algorithms for Problem No. 2.

Sol. No.	CE-NRGA		NSGA-II		MOPSO	
	Cost	Reliability	Cost	Reliability	Cost	Reliability
1	355	0.933	376	0.933	357	0.925
2	317	0.901	376	0.926	337	0.914
3	411	0.951	362	0.914	388	0.937
4	428	0.915	390	0.937	426	0.946
5	329	0.916	426	0.945	391	0.943
6	347	0.928	349	0.908	324	0.899
7	427	0.952	478	0.957	–	–
8	410	0.950	442	0.951	–	–
9	316	0.898	363	0.920	–	–
10	324	0.908	412	0.943	–	–
11	375	0.942	441	0.950	–	–
12	360	0.939	460	0.954	–	–
13	392	0.944	408	0.941	–	–
14	396	0.947	–	–	–	–

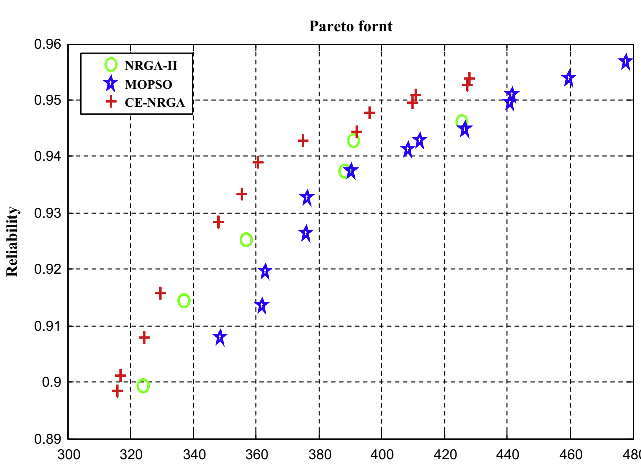


Fig. 11. The Pareto fronts obtained by the three algorithms for Problem No. 2.

Table 12
The Pareto solution of the algorithms for Problem No. 3.

Sol. No.	CE-NRGA		NSGA-II		MOPSO	
	Cost	Reliability	Cost	Reliability	Cost	Reliability
1	22	0.973	147	0.892	135	0.874
2	182	0.943	247	0.897	193	0.955
3	250	0.983	222	0.973	159	0.925
4	239	0.981	238	0.979	165	0.927
5	135	0.874	181	0.943	–	–
6	129	0.850	158	0.915	–	–
7	200	0.966	250	0.983	–	–
8	210	0.967	158	0.915	–	–
9	148	0.901	238	0.983	–	–
10	217	0.971	250	0.975	–	–
11	227	0.975	250	0.981	–	–
12	131	0.859	–	–	–	–
13	200	0.957	–	–	–	–

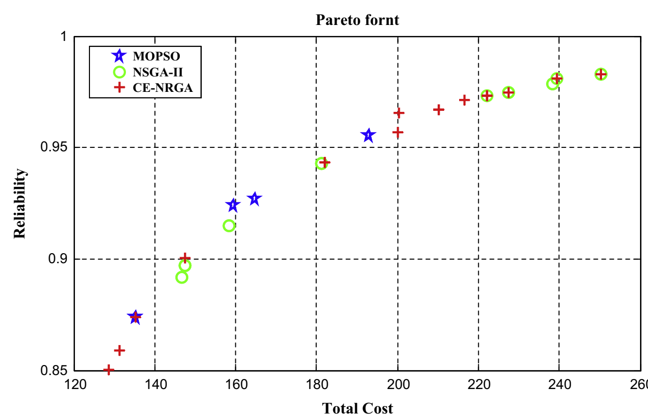


Fig. 12. The Pareto fronts obtained by the three algorithms for Problem No. 3.

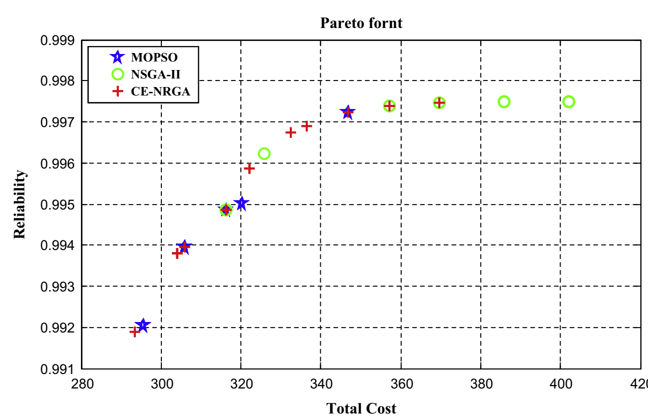


Fig. 13. The Pareto fronts obtained by the three algorithms for Problem No. 4.

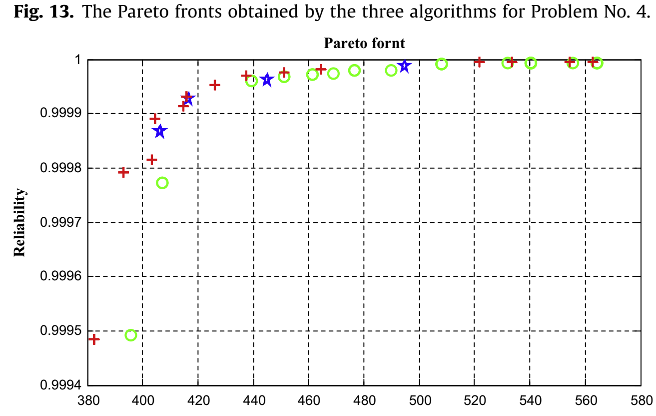


Fig. 14. The Pareto fronts obtained by the three algorithms for Problem No. 5.

Table 13
A partial system configuration of problem No. 2 generated by CE-NRGA.

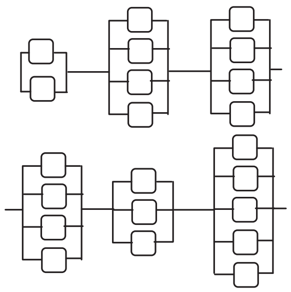
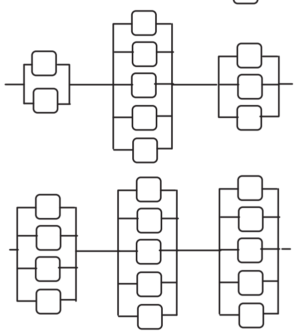
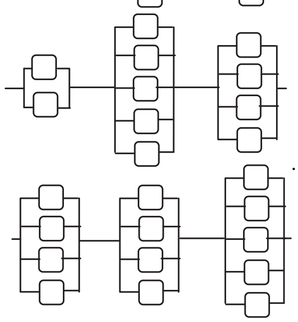
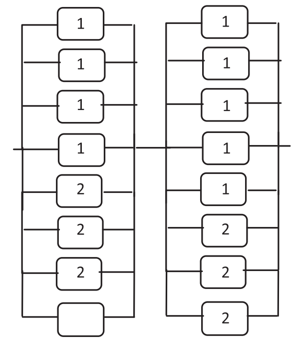
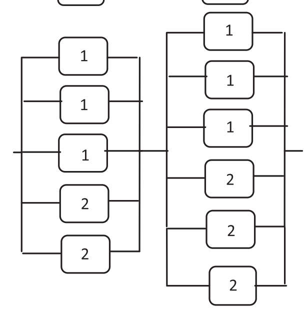
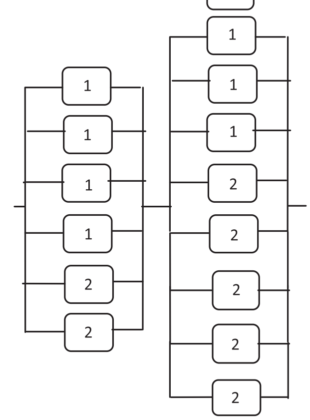
Sol. No.	Period	System configuration diagram	Cost	Reliability
2	1 2		317	0.901
1	1 2		355	0.933
6	1 2		348	0.928

Table 14
A partial system configuration of problem No. 3 generated by CE-NRGA.

Sol. No.	System configuration diagram	Cost	Reliability
6		129	0.850
5		135	0.874
2		182	0.943

chi experiments, the values of the parameters are classified into three levels Low, Medium, and High, coded respectively by 1, 2, and 3. Table 3 shows the parameters, their denotations in parentheses, and their levels.

These values and their levels are obtained after running the algorithms a multitude of times on the generated numerical examples. In this work, a Taguchi approach with 9 arrays (L_9) is used to reach the optimal values of the algorithms' parameters. Table 4 depicts the response values obtained by both the CE-NRGA and NSGA-II algorithms while these values for MOPSO are presented in Table 5. The optimal levels of the parameters of the three algorithms resulting from the Taguchi method are shown in Table 6.

The mean S/N ratios of the algorithms' parameters are shown in Fig. 9 where the smaller value corresponds to the optimal level in each. In Fig. 9, characters *A*, *B*, *C* and *D* are associated with the algorithms' parameters as mentioned in Table 2. The experimental results of the applied metrics on the numerical problems obtained from CE-NRGA and NSGA-II are displayed in Table 7 while the values for MOPSO are demonstrated in Table 8.

To compare the performance of the algorithms, the averages of the metrics on the different numerical problems obtained from the three algorithms are computed which show that CE-NRGA outperforms the others in almost all of the cases. Also, the ANOVA test of an one-way analysis of variance shown in Table 9 depicts that there is a significant difference among the algorithms' performance in terms of metrics MID, NOS, MS, SP, SNS and CT while there is no significant different among the algorithms in terms of SNS.

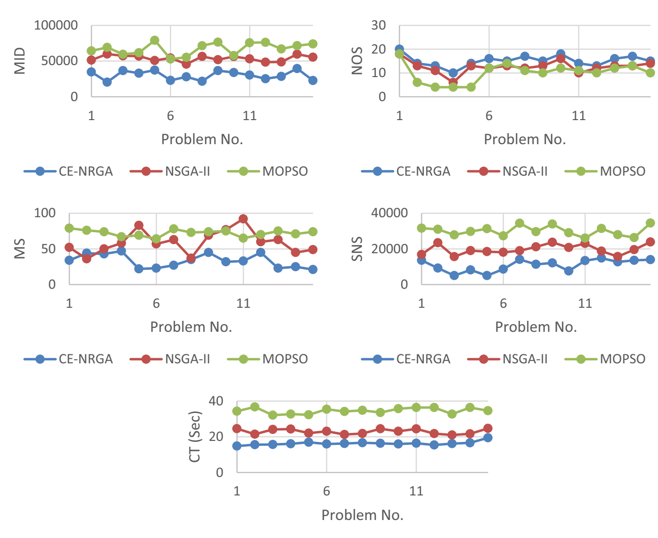


Fig. 15. Graphical summary of the performance of the algorithms.

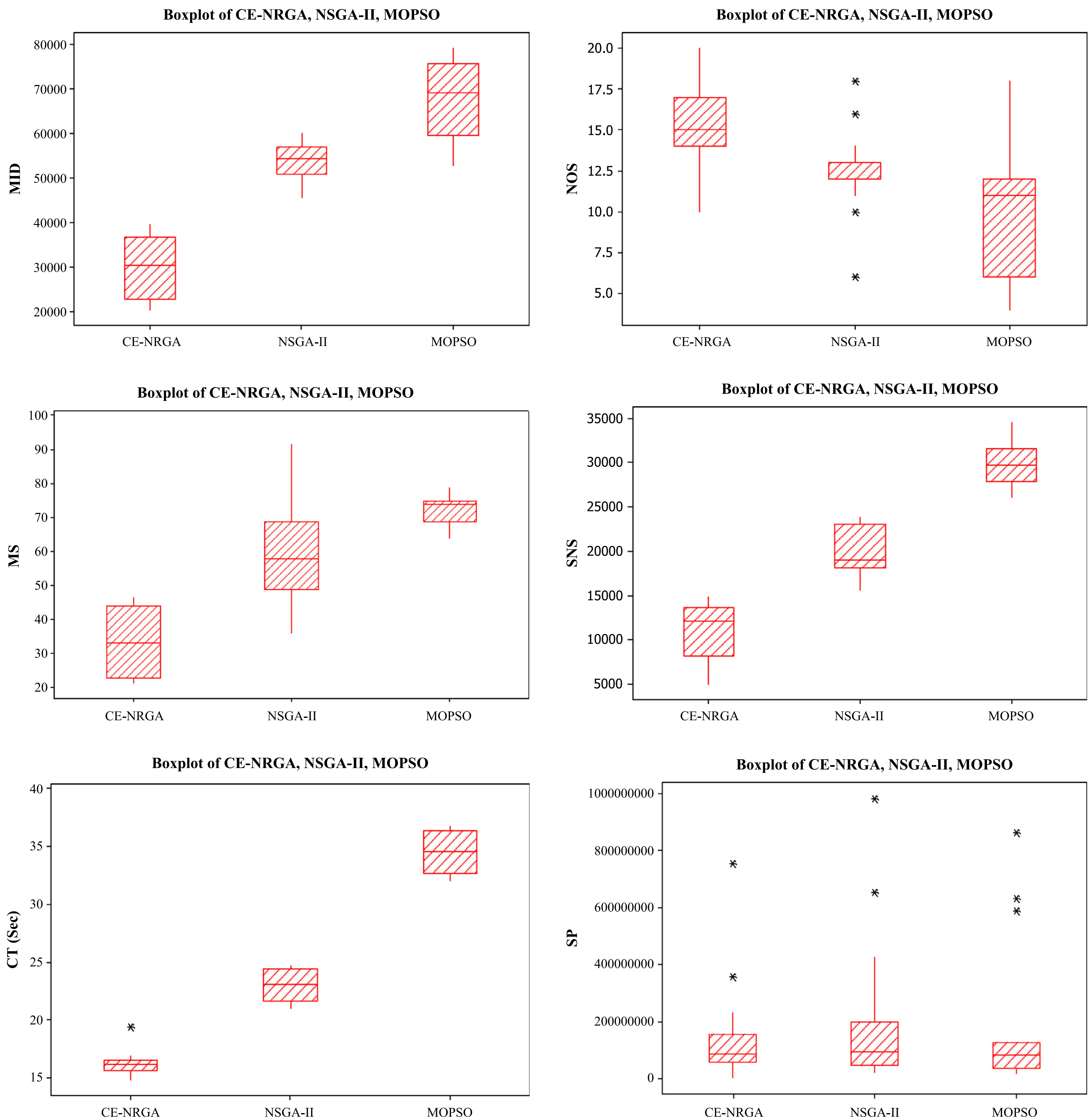


Fig. 16. Box-plots of the statistical tests on all metrics.

To make it clear how the optimal solutions are obtained, Table 10 shows the Pareto solution of the three algorithms for Problem No. 1 where Fig. 10 displays it pictorially. Table 11 also demonstrates the Pareto fronts of the proposed algorithms for Problem No. 2 where a pictorial representation of these solutions is shown in Fig. 11. The Pareto solutions of CE-NRGA, NSGA-II and MOPSO for Problem No. 3 are displayed in Table 12 and their graphical representation is shown in Fig. 12. Furthermore, the Pareto fronts of the algorithms for Problem No. 4 and Problem No. 5 are shown in Figs. 13 and 14, respectively. Figs. 10–14 show the convergence and diversity of the Pareto solutions obtained by the three algorithms for Problem No. 1 to Problem No. 4 respectively, as well. To specify how a parallel system is designed by the Pareto fronts obtained from CE-NRGA, Tables 13 and 14 depict the partial system configuration of Problem No. 2 and Problem No. 3 generated by CE-NRGA, respectively, along with their optimum values of the total cost and the system reliability. In Table 13, the type of all the components is identical while the components in Table 14 are in two types. The graphical summary of the performance of the algorithms for the metrics is depicted in Fig. 14 where the results are in the favor of CE-NRGA (see Fig. 15). Fig. 16 displays Box-plots of the statistical tests on all the metrics which shows CE-NRGA outperforms NSGA-II and MOPSO.

6. Conclusion and recommendations for future

In this study, a mixed-integer nonlinear mathematical model was used to formulate a multi-objective SPIRAP in which the components were purchased in multiple types and stored in the storage units with limited capacity. The components were delivered to the owner of a series-parallel system using the vehicles with different capacities. The aim was to find the optimal number of components to be installed on a series-parallel system so that the total inventory cost is minimized and the system reliability is maximized. Three well-known algorithms i.e. CE-NRGA, NSGA-II and MOPSO were used to solve the problem. Some numerical examples were generated where several metrics were applied to compare the results of the algorithms graphically and statistically.

As a recommendation for future works, the model can be improved for a supply chain network. Moreover, the model can be considered for the case where shortages are allowed. A stochastic or fuzzy model of the problem is also recommended for future research. In addition, different solution methodologies as well as other meta-heuristic algorithms can be studied in future research.

Acknowledgement

The authors would like to thank the anonymous reviewers and the editor for their insightful comments and suggestions.

References

Bharti, P. S., Maheshwari, S., & Sharma, C. (2012). Multi-objective optimization of electric-discharge machining process using controlled elitist NSGA-II. *Journal of Mechanical Science and Technology, 26*(6), 1875–1883.
Cárdenas-Barrón, L. E., Sarkar, B., & Treviño-Garza, G. (2013). Easy and improved algorithms to joint determination of the replenishment lot size and number of shipments for an EPQ model with rework. *Mathematical and Computational Applications, 18*(2), 132–138.
Dey, J. K., Mondal, S. K., & Maiti, M. (2008). Two storage inventory problem with dynamic demand and interval valued lead-time over finite time horizon under inflation and time-value of money. *European Journal of Operational Research, 185*(1), 170–194.
Duari, N. K., & Chakraborti, T. (2015). A cash flow-oriented EOQ model of deteriorating items with increasing demand rate and shortages under permissible delay in payment. *International Journal of Operational Research, 24*(2), 145–160.
Govindan, K., Jafarian, A., Khodaverdi, R., & Devika, K. (2014). Two-echelon multiple-vehicle location–routing problem with time windows for optimization of sustainable supply chain network of perishable food. *International Journal of Production Economics, 152*, 9–28.
Hajipour, V., Fattahi, P., Tavana, M., & Di Caprio, D. (2016). Multi-objective multi-layer congested facility location-allocation problem optimization with Pareto-based meta-heuristics. *Applied Mathematical Modelling, 40*(7), 4948–4969.
Jalali, S., Seifbarghy, M., Sadeghi, J., & Ahmadi, S. (2015). Optimizing a bi-objective reliable facility location problem with adapted stochastic measures using tuned-parameter multi-objective algorithms. *Knowledge-Based Systems*.
Jawla, P., & Singh, S. (2016). A reverse logistic inventory model for imperfect production process with preservation technology investment under learning and inflationary environment. *Uncertain Supply Chain Management, 4*(2), 107–122.
Karimi, N., Zandieh, M., & Karamooz, H. (2010). Bi-objective group scheduling in hybrid flexible flowshop: A multi-phase approach. *Expert Systems with Applications, 37*(6), 4024–4032.
Kong, X., Gao, L., Ouyang, H., & Li, S. (2015). Solving the redundancy allocation problem with multiple strategy choices using a new simplified particle swarm optimization. *Reliability Engineering & System Safety, 144*, 147–158.
Kumar, N., & Kumar, S. (2016). Effect of learning and salvage worth on an inventory model for deteriorating items with inventory-dependent demand rate and partial backlogging with capability constraints. *Uncertain Supply Chain Management, 4*(2), 123–136.
Mondal, M., Maity, A. K., Maiti, M. K., & Maiti, M. (2013). A production-repairing inventory model with fuzzy rough coefficients under inflation and time value of money. *Applied Mathematical Modelling, 37*(5), 3200–3215.
Mousavi, S. M., Alikar, N., & Niaki, S. T. A. (2015). An improved fruit fly optimization algorithm to solve the homogeneous fuzzy series–parallel redundancy allocation problem under discount strategies. *Soft Computing*, 1–27.
Mousavi, S. M., Alikar, N., Niaki, S. T. A., & Bahreininejad, A. (2015a). Optimizing a location allocation-inventory problem in a two-echelon supply chain network: A modified fruit fly optimization algorithm. *Computers & Industrial Engineering, 87*, 543–560.
Mousavi, S. M., Alikar, N., Niaki, S. T. A., & Bahreininejad, A. (2015b). Two tuned multi-objective meta-heuristic algorithms for solving a fuzzy multi-state redundancy allocation problem under discount strategies. *Applied Mathematical Modelling*.
Mousavi, S. M., Bahreininejad, A., Musa, S. N., & Yusof, F. (2014). A modified particle swarm optimization for solving the integrated location and inventory control problems in a two-echelon supply chain network. *Journal of Intelligent Manufacturing*, 1–16.
Mousavi, S. M., Hajipour, V., Niaki, S. T. A., & Alikar, N. (2013). Optimizing multi-item multi-period inventory control system with discounted cash flow and inflation: Two calibrated meta-heuristic algorithms. *Applied Mathematical Modelling, 37*(4), 2241–2256.
Mousavi, S. M., Sadeghi, J., Niaki, S. T. A., & Tavana, M. (2016). A bi-objective inventory optimization model under inflation and discount using tuned Pareto-based algorithms: NSGA-II, NRGA, and MOPSO. *Applied Soft Computing, 43*, 57–72.
Naka, S., Genji, T., Yura, T., & Fukuyama, Y. (2001). Practical distribution state estimation using hybrid particle swarm optimization. *Power engineering society winter meeting, 2001. IEEE* (Vol. 2, pp. 815–820). IEEE.
Pareek, S., & Dhaka, V. (2014). Fuzzy EOQ models for deterioration items under discounted cash flow approach when supplier credits are linked to order quantity. *International Journal of Logistics Systems and Management, 20*(1), 24–41.
Pasandideh, S. H. R., & Asadi, K. (2016). A priority-based modified encoding–decoding procedure for the design of a bi-objective SC network using meta-heuristic algorithms. *International Journal of Management Science and Engineering Management, 11*(1), 8–21.
Pasandideh, S. H. R., Niaki, S. T. A., & Asadi, K. (2015). Bi-objective optimization of a multi-product multi-period three-echelon supply chain problem under uncertain environments: NSGA-II and NRGA. *Information Sciences, 292*, 57–74.
Pasandideh, S. H. R., Niaki, S. T. A., & Maleki, L. (2015). A bi-objective remanufacturing problem within queuing framework: An imperialist competitive algorithm. *International Journal of Management Science and Engineering Management, 10*(3), 199–209.
Sadeghi, J., & Niaki, S. T. A. (2015). Two parameter tuned multi-objective evolutionary algorithms for a bi-objective vendor managed inventory model with trapezoidal fuzzy demand. *Applied Soft Computing, 30*, 567–576.
Sadeghi, J., Sadeghi, S., & Niaki, S. T. A. (2014). A hybrid vendor managed inventory and redundancy allocation optimization problem in supply chain management: An NSGA-II with tuned parameters. *Computers & Operations Research, 41*, 53–64.
Salmasnia, A., Ameri, E., & Niaki, S. T. A. (2016). A robust loss function approach for a multi-objective redundancy allocation problem. *Applied Mathematical Modelling, 40*(1), 635–645.
Sarkar, B. (2012). An EOQ model with delay in payments and time varying deterioration rate. *Mathematical and Computer Modelling, 55*(3), 367–377.
Sarkar, B. (2013). A production-inventory model with probabilistic deterioration in two-echelon supply chain management. *Applied Mathematical Modelling, 37*(5), 3138–3151.
Sarkar, B. (2016). Supply chain coordination with variable backorder, inspections, and discount policy for fixed lifetime products. *Mathematical Problems in Engineering*.
Sarkar, B., & Majumder, A. (2013). Integrated vendor–buyer supply chain model with vendor's setup cost reduction. *Applied Mathematics and Computation, 224*, 362–371.
Sarkar, B., Sana, S. S., & Chaudhuri, K. (2010). Optimal reliability, production lotsize and safety stock: An economic manufacturing quantity model. *International Journal of Management Science and Engineering Management, 5*(3), 192–202.
Sarkar, B., & Saren, S. (2015). Partial trade-credit policy of retailer with exponentially deteriorating items. *International Journal of Applied and Computational Mathematics, 1*(3), 343–368.
Sarkar, B., Saren, S., & Cárdenas-Barrón, L. E. (2015). An inventory model with trade-credit policy and variable deterioration for fixed lifetime products. *Annals of Operations Research, 229*(1), 677–702.
Sarkar, B., Saren, S., & Wee, H.-M. (2013). An inventory model with variable demand, component cost and selling price for deteriorating items. *Economic Modelling, 30*, 306–310.
Sarkar, B., & Sarkar, S. (2013). Variable deterioration and demand—An inventory model. *Economic Modelling, 31*, 548–556.
Sarkar, B., Sarkar, S., & Yun, W. Y. (2016). Retailer's optimal strategy for fixed lifetime products. *International Journal of Machine Learning and Cybernetics, 7*(1), 121–133.
Saxena, P., Singh, S., & Sangal, I. (2016). A two-warehouse production inventory model with trade credit under reliability consideration. *Uncertain Supply Chain Management, 4*(4), 319–330.
Schott, J. R. (1995). Fault tolerant design using single and multicriteria genetic algorithm optimization. In *DTIC document*. .
Sett, B. K., Sarkar, B., & Goswami, A. (2012). A two-warehouse inventory model with increasing demand and time varying deterioration. *Scientia Iranica, 19*(6), 1969–1977.
Shahsavar, A., Najafi, A. A., & Niaki, S. T. A. (2015). Three self-adaptive multi-objective evolutionary algorithms for a triple-objective project scheduling problem. *Computers & Industrial Engineering, 87*, 4–15.
Shi, Y., & Eberhart, R. C. (1999). Empirical study of particle swarm optimization. *Evolutionary computation, 1999. CEC 99. Proceedings of the 1999 congress on* (Vol. 3). IEEE.

Shukla, H., Tripathi, R., & Sang, N. (2016). Economic production quantity model for defective items under deterioration. *Uncertain Supply Chain Management, 4*(3), 221–232.

Teimouri, M., Zaretalab, A., Niaki, S., & Sharifi, M. (2016). An efficient memory-based electromagnetism-like mechanism for the redundancy allocation problem. *Applied Soft Computing, 38*, 423–436.

Wang, W.-L., & Xu, G.-Q. (2009). Stability analysis of a complex standby system with constant waiting and different repairman criteria incorporating environmental failure. *Applied Mathematical Modelling, 33*(2), 724–743.

Wu, J., Skouri, K., Teng, J.-T., & Hu, Y. (2016). Two inventory systems with trapezoidal-type demand rate and time-dependent deterioration and backlogging. *Expert Systems with Applications, 46*, 367–379.

Xie, W., Liao, H., & Jin, T. (2014). Maximizing system availability through joint decision on component redundancy and spares inventory. *European Journal of Operational Research, 237*(1), 164–176.

Yadav, D., Singh, S., & Kumari, R. (2015). Retailer's optimal policy under inflation in fuzzy environment with trade credit. *International Journal of Systems Science, 46*(4), 754–762.

Zhang, E., & Chen, Q. (2016). Multi-objective reliability redundancy allocation in an interval environment using particle swarm optimization. *Reliability Engineering & System Safety, 145*, 83–92.

Zitzler, E., & Thiele, L. (1998). Multiobjective optimization using evolutionary algorithms—A comparative case study. In *Parallel problem solving from nature—PPSN V* (pp. 292–301). Springer.