Journal of Optimization in Industrial Engineering,
Vol.17, Issue 2, Summer & Autumn 2024, 151-182
DOI: 10.22094/QJIE.2024.950699

Research Article

Constrained portfolio selection model at considering risk-adjusted measure by using the Genetic Network Programming

Iman Bavarsad Salehpoor [a, b,*], Saber Molla-Alizadeh-Zavardehi [b]

[a] *Industrial Engineering Department, Amirkabir University of Technology, P.O. Box 15875-4413, Tehran, Iran*
[b] *Department of Industrial Engineering, Masjed-Soleiman BranchIslamic Azad UniversityMasjed-Soleiman, Iran*

Received 19 March 2018 ; Revised 15 August 2024 ; Accepted 30 August 2024

Abstract

This paper presents a new decision-making method of stock portfolio optimization issues in different risk measures by using new evolutionary computing method and cardinality constrains which is mentioned as hybrid meta-heuristic algorithms. Based on Mean-Variance (MV) Method by Markowitz we collected three risk levels; Mean-Absolute-Deviation (MAD), Semi-Variance (SV) and Variance-With-Skewness (VWS). The developed algorithms are Electromagnetism-like Algorithm (EM), Genetic Algorithm (GA), Genetic Network Programming (GNP), Particle Swarm Optimization (PSO) and Simulated Annealing (SA). Also, a diversification mechanism strategy is implemented and hybridized with the developed algorithms to increase the diversity and overcome local optimality. The sustainability of this proposed model is verified by 50 factories on the Iranian stock exchange. Finally, experimental results of proposed algorithms with cardinality constraint are compared with each other by four effective metrics in which the algorithms performance for achieving the optimal solution discussed. In addition, to verify the statistical validity and accurately analyze of the results, we have done the Analysis Of Variance (ANOVA) technique which the success of this method was proved.

***Keywords*:** Portfolio optimization; hybrid meta-heuristic algorithms; Iran Stock Exchange; mean–variance, mean absolute deviation, semi variance, variance with skewness.

1. Introduction

With the application of an Evolutionary Commutating named GNP, this article presents a decision-making model of stock portfolio optimization. On most portfolio selection issues, there should be a tradeoff between risk and return. Also, investors are looking for a set of the assets that have the optimal weight in the portfolio. By application of an Evolutionary Computation method, this article presents a new decision-making model of stock portfolio optimization named hybrid meta-heuristic algorithms. The most notable parameters of portfolio optimization problems are the risk and the expected return (Mabu, Hirasawa, & Hu, 2007). Also, the return of the assets has a normal Distribution. At first, (Markowitz, 1952) who presented one of the main contributions of this problem named mean–variance model. According to the founding of standard mean–variance model, the investors are avoiding from risk. (Jia & Dyer, 1996) demonstrated that these conditions are Sometimes to be located. Regarding the convenient risk measure, the mean–variance of objective function may not be the best selection of Shareholders. In addition, other risk measures may be more suitable. From a workable point of view, real world capitalists have to confront many restrictions in risk formulations, extent of portfolio, business constraint, etc. These restraints may be formulated in nonlinear programming models which are significantly difficult to answer than the main model.

* Corresponding author Email address: iman.bssp@gmail.com

The mean–variance model proposed by (Markowitz, 1952) which assumed that asset returns are a random variable that follows a normal distribution. On the other hand, the return on assets can be determined by means of variance properties which considers this variation here as a risk. Moreover, based on Markowitz model, other researchers have introduced other dimensions of the risks mentioned at the beginning of the paper. In the following, the risk measures for selecting and optimizing stock portfolios base on the math structure are shown.

1.1. Mean–variance structure

As stated, Markowitz is one of the leading researchers in the providing variance as a risk measure. The structure of the model formulation as follows.

$$\min imize \sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \sigma_{ij} \quad (1)$$

$$\text{St:} \sum_{i=1}^{N} w_i \mu_i = R^* \quad (2)$$

$$\sum_{i=1}^{N} w_i = 1 \quad (3)$$

$$0 \le w_i \le 1, i = 1, 2, ..., N \quad (3)$$

N is the total assets are currently available, and w_i is the part of the portfolio assigned to assets i, $0 \le w_i \le 1, i = 1,2,...,N$. Here, the symbol μ_i intend to return of the asset i. The parameter σ_{ij} is the covariance between assets i and j. Correspondingly, the subsets i and j vary from 1 to N. In the first equation, the risk of the whole stock portfolio is minimized. In the second equation, the yield rate of R^* is guaranteed on portfolio. The third equation states that the total weights are equal to one. In the fourth equation, the proportion of any stock in the portfolio is between zero and one. The equations (1)–(4) are quadratic programming (QP) that the optimal answer given by the current nonlinear programming methods and software. As long as the formulation is solved by different R^*, a sequence of effective points will be created. This series creates the effective points of the optimization problem by the desirability of the shareholders. Also, this series has the lowest risk and the highest returns on portfolio. In other words, all portfolio selection issues need to look an effective rate of risk and return.

Also, the weighting parameter λ $(0 \le \lambda \le 1)$ proposes as follow

$$\min \lambda[\sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \sigma_{ij}] - (1-\lambda)[\sum_{i=1}^{N} w_i \mu_i] \quad (5)$$

$$\text{Subject to } \sum_{i=1}^{N} w_i = 1 \quad (6)$$

$$\sum_{i=1}^{N} z_i = k \quad (7)$$

$$\varepsilon_i z_i \le w_i \le \delta_i z_i, i = 1,2,...,N \quad (8)$$

$$Z_i \in \{0,1\}; i = 1,2,...,n \quad (9)$$

The symbol k indicates the number of assets that the investor intends to keep in the portfolio. The parameters ε_i and δ_i are a least and maximum proportion of those assets, respectively. These parameters must be held in the portfolio on asset i $(i = 1,2,...,N)$. The parameter z_i is set to one as the asset i $(i = 1,2,...,N)$; otherwise, the parameter z_i is set to zero.

In the fifth equation, by putting the value between zero and one for λ , all risk combinations and returns are achieved for the modeling. Also, the values of λ is between zero to one. In general, the case $\lambda = 0$ and $\lambda = 1$ demonstrate maximum return and minimum risk, respectively. Further, the sixth equation states that the total weights are equal to one. The seventh equation showed a limitation on the number of intended assets. It certifies that the total assets held must be equal to k assets. The eighth constraint explains the minimum and maximum share of each asset in the stock which can be held in the portfolio. It certifies that the ratio w_i should be between ε_i and δ_i as asset is held in the portfolio ($z_i = 1$). As none of asset i is held ($z_i = 0$) its proportion w_i is zero on portfolio. The ninth equation indicates integer constraints regarding to z_i. At this time, by weighting the parameter λ $(0 \le \lambda \le 1)$ and by using the mentioned equations above, we can draw the efficient boundary of these constraints. (T-J Chang, Meade, Beasley, & Sharaiha, 2000) used the heuristic methods to solve this model with constraints for the portfolio optimization. Also, the portfolio optimization with cardinality constraints efficient frontier and based on the hybrid differential evolution presented by other researchers (H.-H. Huang & Wang, 2013; Ma, Gao, & Wang, 2012; Mishra, Panda, & Majhi, 2016). The researchers (H.-H. Huang & Wang, 2013; Ma et al., 2012; Mishra et al., 2016) used multi-objective evolutionary algorithms and predicted the expected returns.

1.2. SV structure

The mathematical structure of Standard mean–variance model is based on the principle. The return on assets is a variable that has a normal distribution. According to the definition of the risk, the investors are intrinsically risk averse (Frajtova-Michalikova, Spuchľakova, & Misankova, 2015). For this reason, there are new definitions of the math structure of the risk like semi-variance. By these assumptions, the objective function and others component of model can be changed. Due to the symmetry of the return distribution, the initial model was not able to accurately estimate the stock portfolio behavior. For this reason, the above model was proposed by Markowitz (Markowitz, 1952) to escape this type of behavior in a time series.

The semi-variance model for the cardinality constraints efficient frontier (CCEF) is as follow

$$\min \lambda[\sum_{t=1, r_t \le \bar{r}}^{T} (r_t - \bar{r})^2 / T] - (1-\lambda)\bar{r} \quad (10)$$

$$W_{i=} v_{iT} x_i / c_{cash}, i = 1,...,n \quad (11)$$

$$r_t = \log_e(\sum_{i=1}^{n} v_{it} x_i / \sum_{i=1}^{n} v_{it-1} x_i), t = 1,...,T \quad (12)$$

$$\sum_{i=1}^{n} z_i = k \quad (13)$$

$$\varepsilon_i z_i \le w_i \le \delta_i z_i, i = 1,...,n \quad (14)$$

$$\sum_{i=1}^{n} w_i = 1 \tag{15}$$

$$r_t = \log_e[(\sum_{i=1}^{n} w_i v_{it} / v_{iT}) / (\sum_{i=1}^{n} w_i v_{it-1} / v_{iT})], t = 1,...,T \tag{16}$$

$$w_i \geq 0, i = 1,...,n \tag{17}$$

$$z_i \in \{0,1\}, i = 1,...,n \tag{18}$$

$$\bar{r} = \sum_{t=1}^{T} r_t / T \tag{19}$$

The parameter $\bar{r}$ is the mean portfolio return through the time period $1,2,\dots,T$. Thus, semi-variance notices only downside risk and in other words considers returns below $\bar{r}$ that it is obtained from the following equation:

$$\sum_{t=1;r_t \leq \bar{r}}^{T} (r_t - \bar{r})^2 / T \tag{20}$$

This means that risk is no longer symmetric, and time periods are not interested in which $r_t \geq \bar{r}$.

In order to extend the relevant historical data, the symbol T should be determined. The relevant historical data is reliable over time $(t = 0,\dots,T)$. Also, the parameter v_{it} is the value of each unit of stock i within the relevant time period. The symbol c_{cash} is stated that all cash of the investors to consider in portfolio. The parameter x_i is the number of stocks i that selected for hold in the portfolio. The parameter z_i is set to one as the asset i $(i = 1,2,...,N)$; otherwise, the parameter z_i is set to zero. Also, the parameter w_i is the weight of c_{cash}. This weight is related to the time T in stock i. r_t is the rate of return within a time period given by the portfolio at time t $(t = 1,2,\dots,T)$. Based on the variables presented above, the values w_i and r_t are calculated. The seventh constraint explains that the w_i is the weight of the asset i in the portfolio at the time t. Moreover, the total value of the portfolio at time t is $\sum_{i=1}^{n} v_{it} x_i$. The twelfth constraint explains r_t to be the return on the portfolio.

In terms of discrete time, the following constraints are assigned to the problem:

$$\sum_{i=1}^{N} z_i = k \tag{21}$$

$$\varepsilon_i z_i \leq v_{it} x_i / c_{cash} \leq \delta_i z_i, i = 1,...,n \tag{22}$$

$$\sum_{i=1}^{n} v_{it} x_i = c_{cash} \tag{23}$$

$$x_i \geq 0, i = 1,...,n \tag{24}$$

$$z_i \in \{0,1\}, i = 1,...,n \tag{25}$$

After some simplification mathematical, the above equations can be converted to Eqs (13-19).

In addition, (Jaaman, Lam, & Isa, 2011) considered about portfolio optimization models and evaluated the implementation, as well as portfolio composition of the mean-variance model with other downside risk models, which are conditional value at risk models, semi-variance, and below target semi-variance. They showed that the conditional value at risk is effective to control downside risk and is a better option for risk opposing investors. A new definition of semi variance for portfolio selection suggested by (Yang, Lin, Chang, & Chang, 2011) about investing on military area. According to their suggestion, a semi-variance model is assigned as objective function and to create efficient frontier a measure of risk containing cardinality constraints is supplied for portfolio optimization. Moreover, genetic algorithm has been used to obtain optimal weights of assets. According to stability on the results of the experiment, the designed algorithms are appropriate to investing at military assets and a model with this formulation is impressive for resolving the portfolio selection problem.

1.3. MAD structure

(Konno & Yamazaki, 1991) based on standard deviation parameter of normal distribution introduced the first idea of MAD portfolio optimization model. Their model is an alternating to the Markowitz mean–variance portfolio selection model. Their idea is the linear programming to resolve the portfolio optimization problem. (Konno & Yamazaki, 1991) showed in their work that the MAD model results were equivalent to the results of the Markowitz mean–variance model. Unlike the mean-variance model, the MAD model can solve this problem by using linear programing. The following form is the mean absolute deviation model for the CCEF:

$$\min \lambda[\sum_{t=1}^{T} \left| r_t - \bar{r} \right| / T] - (1-\lambda)\bar{r} \tag{26}$$

St to Eqs. (13) to (19).

One of the researchers who solved the MAD model by using the classical approach was (Liu, 2011) that presented the computing profits of the MAD model and solving it by that techniques over the Markowitz mean–variance model. To calculate the lower and upper bounds of the investment return of the portfolio selection problem, they constructed a pair of two-level mathematical programming models. Results illustrated that the return of the portfolio problem was certainly in a range. A new model for portfolio optimization at cardinality constraint mean-absolute semi-variance (MASV) presented by (Vercher & Bermúdez, 2015). They presented acceptable results based on the assumptions of the relevant model in portfolio optimization.

1.4. Variance with skewness

First idea of variance with skewness portfolio optimization model as supplementary option to the Markowitz mean–variance portfolio selection model suggested by (Samuelson, 1958) . According to many experimental studies, portfolio returns are usually asymmetric. In this case, the investors would choose a portfolio return with larger degree of asymmetry as the mean value and variance are similar (Jiang, Ma, & An, 2016).

(Canela & Collazo, 2007) studied about different parts of industry analysis on the impacts of portfolio selection when the skewness on the desired variables is taken into account as a risk measure. To prioritize the objectives of the investor on emerging industries at stock market, they applied multiple goals programming to select the most optimal portfolio. The main idea of their research at the time of selecting optimal portfolio was that the best type of skewness of parameters considered. In this way, the optimal decision will take place where in portfolio selection will be done by specify effective range of skewness. At this time, investors can be done sensitivity analysis. (Konno & Yamamoto, 2005) applied the integer programming idea to solve a mean–variance skewness portfolio optimization model.

Follow is the Variance with skewness model for the CCEF:

$$\min \lambda[\sum_{t=1}^{T}(r_t-\bar{r})^2/T]-(1-\lambda)\bar{r}-\theta[(\sum_{t=1}^{T}(r_t-\bar{r})^3/T)/(\sum_{t=1}^{T}(r_t-\bar{r})^2/T)^{\frac{3}{2}}] \quad (27)$$

St to Eqs. (13) – (19).

Parameter θ is a weighting factor of statistical society. The amount of its variance skewness must be measured for continue the solution. This parameter states its weights in Eq. (27) which is famous as the coefficient of skewness. Eq. (27) balances variance, expected return and skewness simultaneously. For a specific value of θ, we can produce an efficient frontier which mirrors attitude of skewness. Hence, the downside risk will be decrease when considering positive skewness in portfolio returns. This is favorable to investors. As is clear from the Eq. (27), the coefficient of skewness is impressed by changing the returns. Thus, it is possible to have portfolios with a like skewness but quite different downside behavior.

2. Literature Review

By a variety of methods about portfolio optimization, various the authors have attempted to find solution to these problems. To solve of the problem in large scale, accurate models cannot be most successful. In order to solve portfolio problems, many authors tried to find solution by using mathematical programming model. (Aguilar-Rivera, Valenzuela-Rendón, & Rodríguez-Ortiz, 2015) investigated the performance of multi-objective evolutionary algorithms (MOEA), GA and genetic programming (GP) as a techniques to select the optimal portfolio (Macedo, Godinho, & Alves, 2017) studied in the field of MOEA.

A new MOEA with name non-dominated sorting and Local search (NSLS) proposed by (B. Chen, Lin, Zeng, Xu, & Zhang, 2017). This method has a new convergence approach. This is because that this algorithm is capable to change the solution space to create a better generation. Comparison results of NSLS with other algorithms show a better performance in term of the diversity and convergence to CCEF.

(Tun-Jen Chang, Yang, & Chang, 2009) considered GA to solve hard portfolio optimization problems with various risk measures. They also presented some cases of optimization models with cardinality limitations to select portfolio. For portfolio optimization with cardinality constraints, (Ma et al., 2012) applied hybrid Differential Evolution (DE). The amount of risk on their paper was taken as the objective function. In addition, they pointed that the gained numerical results were logical and the given algorithm was practical. (X. Huang, 2008) expressed a new description of risk for the portfolio selection. According to that description, Huang introduced a new optimization model. The suggested hybrid intelligent algorithm was offered as a good alternate to resolve the optimization problem in intricate cases. (K. Anagnostopoulos & G. Mamanis, 2011) introduced Computational collation of the multi-purpose evolutionary process for portfolio optimization models based on the problems that defined on mean–variance cardinality limitations.

By utilizing mathematical programming methods and mean absolute deviation risk as functional constraints, (Liu, 2011) distinguished a model. That model detected the upper and lower ranges of the return from the portfolio. Another scale of risk is Semi-variance that employed by (Yang et al., 2011). They used Genetic Algorithm as a heuristic method to provide an impressive weighting ratio to purchasing on assets.

Also, Tabu Search was utilized as a meta-heuristic method by them. By using the results of the above algorithms, their experiments indicated that consistency to invest on assets for solving the portfolio optimization models.

A comprehensive survey about abilities of swarm intelligence (SI) for portfolio optimization presented by (Ertenlice & Kalayci, 2018). This algorithm had a significant application in terms of the researchers' attention to solving portfolio optimization problems. In that research, they used different risk measure such as Variance with skewness (VWS), Value-at-Risk (VAR), Conditional Value-at-Risk (CVAR) and etc. In that research, the constraints were cardinality constraints (CC), transaction costs (TC) and transaction lots (TL). By this risk measure and constraints, exact method cannot able to determine the optimal efficient frontier. This type of problems can only be solved through heuristic or Meta heuristic algorithms that Evolutionary Algorithms (EA) and SI are two main approaches to solve them. with improved constraint (Deng, Lin, & Lo, 2012) presented a new particle swarm optimization method (PSO) to identify efficient frontier at portfolio optimization.

As a new computationally method, the Artificial Neural Network (ANN) is an important Achievement at the Artificial Intelligence field. The ANN is able to predict financial performances and caught a lot of desirability in the last years. The ANN was used by (Atiya, 2001) to determine and to predict risk. Also, (Dropsy, 2011) used ANN as a nonlinear predicting tools. Those results showed that the prediction was accepted in comparison with other random methods. (Lam, 2004) utilized neural networks (NN) to integrate fundamental and technical analysis for financial performance prediction. Based on NN methodology, (L. Yu, Wang, & Lai, 2008) introduced a new unnatural intelligent technique that the technique used fast and capable radial basis function (RBF) for portfolio selection. Likewise artificial neural network, other methods were used to solve monetary problems (Holland, 1975).

Based on prediction of future returns of stocks and optimization of portfolio, a new method introduced by (Mishra et al., 2016). They used the heuristic functional link-artificial neural network (HFL-ANN) to predict futures return and portfolio optimization. One of the reasons for predicting future returns is the unreliability of past returns as criteria for portfolio optimization (PO). The result demonstrated that Pareto optimal solutions preserve desired diversity and comparable with other models.

(Can B Kalayci, Okkes Ertenlice, Hasan Akyer, & Hakan Aygoren, 2017) presented artificial bee colony (ABC) algorithm to solve cardinality constrained portfolio optimization (CCPO) problem. In that paper, by using repair mechanism (effective limitation on the solution space) and feasibility enforcement along with infeasibility tolerance (temporary violation of constraint) presented a fast and efficient way to achieve optimal solution. Also, (Baykasoğlu, Yunusoglu, & Özsoydan, 2015) presented a greedy randomized adaptive search procedure (GRASP) to solve CCPO. The GRASP has two phases, selection level (to ensure cardinality constraints) and determining the proportions (to remove redundant constraint) of stocks. Another study to cardinality constraint portfolio optimization (CCPO) with nonlinear mixed quadratic programming problem presented by (Woodside-Oriakhi, Lucas, & Beasley, 2011). Their results were compared with (T-J Chang et al., 2000). In that paper, three Meta heuristic approaches such as genetic algorithm (GA), tabu search (TS) and simulated annealing (SA) were used and corresponding results were compared separately.

GA is a good tool for Meta heuristic algorithm. This technique is an acceptable tool for optimization. The multi-purpose genetic algorithm discussed by (D. Lin, Wang, & Yan, 2001) as an appropriate option for portfolio selection problem. (Oh, Kim, Min, & Lee, 2006) suggested another portfolio optimization algorithm based on portfolio beta and GA. Also, (C.-C. Lin & Liu, 2008) discussed Markowitz's model with least iteration lots and they introduced three other models using GA. Development of GA can be studied by GP, which has been recommended by (Koza, 1992).

In financial markets, the structural frameworks of phenotypes are the important interest of GP in stating business rules. Capabilities of GP as a creative approach are an efficient method to automatically develop technical trading principles. Therefore, GP as a good artificial intelligence tool has been used to a vast area of financial markets. (Esfahanipour & Mousavi, 2011) presented a new model by using adjusted risk which has been used GP. (Aguilar-Rivera et al., 2015) introduced evolutionary computing methods to resolve financial problems that include problems which were not discussed by others. They used many techniques in their paper such as GP, GA, learning classifier systems and etc.

(Jalota & Thakur, 2018) presented a new method based on one of new coded GA for constrained portfolio optimization at budget constraint (ensure that available money is utilized completely), cardinality constraints (introduce integer variables in addition to money proportion variables) and lower/upper bound constraints (restrict the allocation of proportion of the budget). This method was named as bounded exponential crossover-power mutation-repair mechanism (BEXPM-RM) by (Thakur, Meghwani, & Jalota, 2014). This research introduced an effective method for searching the solution space based on the repair mechanism which was presented by (T-J Chang et al., 2000).

GNP as the expanded method of GA, which proposed by (Golberg, 1989), and GP, which proposed by (Koza, 1992), developed by researchers (Y. Chen, Mabu, Hirasawa, & Hu, 2007; Eguchi, Hirasawa, Hu, & Ota, 2006; Hirasawa et al., 2008; Mabu et al., 2007). GNP is a capable method mainly for difficult problems such as portfolio optimization. GNP is used to solve the difficult problems because the graphical structure is used in the method. It keeps up all the past operations during the

network stream in the memory. Multi-method search Algorithm to portfolio selection introduced by (Yuen, Chow, Zhang, & Lou, 2016). As there is a dynamic condition in the optimal stock selection, a new model was introduced that was named time adapting genetic network programming (TA-GNP) (Y. Chen, Mabu, & Hirasawa, 2010).

In order to adapt to immediate and dynamic environmental changes, Online learning of GNP was proposed by (Mabu, Hirasawa, Hu, & Murata, 2002). In order to interact with the environment in an effective way, they used Q learning to improve situational processes. GNP was used to develop a more impressive evolutionary optimization technique (Katagiri, Hirasawa, Hu, Murata, & Kosaka, 2002). They illustrated the detailed description about GNP and remarked the differences among GNP, GP and other conventional graph. To compare the range of capabilities, the result of GNP will be compared with the result of GA on efficient frontier to select optimum algorithm (Yuen et al., 2016).

The most obvious difference between GNP and GP is in the presence of a network structure rather than a tree structure. Boot up of GNP is from the start node and because it has no end node; so, the active node does not need to return to the start node (unlike other methods) as GP executes from the root node at each phase. Therefore, due to such a structure, it may be caught up in undesired situations. Also, one of the major problems is creating a bloat of the tree in GP and this problem will be greater when multi branch tree used. Then, its performance to achieve searching the solution space is strongly affected. Due to the presence of large branches at GP root, evolutionary operators may not have the desired performance; unless, randomly desired initial generation happened from the beginning of the program. GNP with network structure, developed to overcome above inefficiency of GP. Because of the structure of the network at GNP, as well as the type of its nodes (judgment nodes and processing nodes) opposite of other methods like GP, there is no need to define many transaction functions (Hirasawa, Okubo, Katagiri, Hu, & Murata, 2001; Katagiri et al., 2002).

Mabu et al. (Mabu, Hirasawa, & Hu, 2004) presented a new evolutionary programming named GNP with reinforcement learning that due to the use of information obtained, the speed of searching solution space is very quickly. In addition to changing the connections, GNP with RL can be changed function nodes which in turn can lead to reduced memory and operating time.

In Table 1, research on solving cardinality constrained portfolio optimization (CCPO) problem is shown.

Table 1
Attempts on solving cardinality constrained portfolio optimization problem.

Authors	**Exact attempt**	**Heuristic attempt**
(Gunjan & Bhattacharyya, 2023)	*	*
(X. Wang, Wang, Li, Li, & Watada, 2023)	Fuzzy	-
(Ahmad, Hasan, Shahid, Imran, & Alam, 2023)	-	GEO
(Lim, 2024)	covariance matrix	-
(Feng, Zhang, & Peng, 2023)	NLP	-
(Kandakoglu, Walther, & Ben Amor, 2024)	MCDM	-
(Jawad, Naz, & Muqaddus, 2024)	Fuzzy-AHP	-
(T-J Chang et al., 2000) T. J. Chang et al. (2000)	-	GA, TS, SA
(Maringer & Kellerer, 2003) Maringer & Kellerer (2003)	-	SA, EA
(Fieldsend, Matatko, & Peng, 2004) Fieldsend et al. (2004)	-	MOEA
(Busetti, 2006) Busetti (2006)	-	GA, TS
(Tian et al., 2006) D. Li et al. (2006)	Lagrangian	-
(Moral-Escudero, Ruiz-Torrubiano, & Suárez, 2006) Moral-Escudero et al. (2006)	QP	GA
(Streichert & Tanaka-Yamawaki, 2006) Streichen &Tanaka-Yamawaki (2006)	QP	MOEA
(S. C. Chiam, Al Mamun, & Low, 2007) Chiam et al. (2007)	-	MOEA
(Fernández & Gómez, 2007) Fernández & Gómez (2007)	-	NN
(Skolpadungket, Dahal, & Harnpornchai, 2007) Skolpadungket et al. (2007)	-	VEGA, SPEA
(S. Chiam, Tan, & Al Mamum, 2008) Chiam et al. (2008)	-	MOEA
(Golmakani & Alishah, 2008) H. R. Golmakani & Alishah (2008)	-	IA
(Shaw, Liu, & Kopman, 2008) Shaw et al. (2008)	Lagrangian	-
(Branke, Scheckenbach, Stein, Deb, & Schmeck, 2009) Branke et al. (2009)	-	MOEA
(Tun-Jen Chang et al., 2009) T. J. Chang et al. (2009)	-	GA
(Cura, 2009) Cura (2009)	-	PSO
(J.-S. Chen, Hou, Wu, & Chang-Chien, 2009) J. Li (2009)	-	GA

(Soleimani, Golmakani, & Salimi, 2009) Soleimani et al. (2009)	-	GA
(Le Thi, Moeini, & Dinh, 2009) Le Thi et al. (2009)	-	DCA
(Deng & Lin, 2010) Deng & Lin (2010a)	-	ACO
(S.-Y. Lin et al., 2010) Deng & Lin (2010b)	-	PSO
(Ruiz-Torrubiano & Suárez, 2010) Ruiz-Torrubiano & Suarez (2010)	QP	GA, SA, EDA
(Thomaidis, 2010) Thomaidis (2010)	-	SA, GA, PSO
(K. P. Anagnostopoulos & G. Mamanis, 2011a) Anagnostopoulos & Mamanis (2011a)	-	NPGA, NSGA, SPEA, MOEA
(K. P. Anagnostopoulos & G. Mamanis, 2011b) Anagnostopoulos & Mamanis (2011b)		NSGA-II, PESA, SPEA2
(Golmakani & Fazel, 2011) Golmakani & Fazel (2011)	-	PSO
(Mozafari, Jolai, & Tafazzoli, 2011) Mozafari et al. (2011)	-	SA
(Z. Wang, Liu, & Kong, 2012) Wang et al. (2012)	-	ABC
(A. H. Chen, Liang, & Liu, 2012) A. H. L. Chen et al. (2012)	-	ABC
(Deng et al., 2012) Deng, Lin, & Lo (2012)	-	PSO
(Murray & Shek, 2012) Murray and Shek (2012)	Local Relaxation, CPLEX	-
(Sadigh, Mokhtari, Iranpoor, & Ghomi, 2012) Sadigh et al. (2012)	-	PSO,NN
(Sadjadi, Gharakhani, & Safari, 2012) Sadjadi et al. (2012)	-	GA
(Cesarone, Scozzari, & Tardella, 2013) cesarone et al. (2013)	QP, CPLEX	-
(A. H. Chen, Liang, & Liu, 2013) A. H. L. Chen et al. (2013)	-	ABC
(X. Cui, Zheng, Zhu, & Sun, 2013) X. T. Cui et al. (2013)	Convex Relaxation, QP	-
(Fogarasi & Levendovszky, 2013) Fogarasi & Levendovszky (2013)	-	SA
(Hajnoori, Amiri, & Alimi, 2013) Hajnoori et al. (2013)	-	IWO
(Kao & Cheng, 2013) Kao & Cheng (2013)	-	BFO
(Lwin & Qu, 2013) K. Lwin & Qu (2013)	-	PBIL+DE
(Zhang & Zhang, 2014) P. Zhang & W. G. Zhang (2014)	discrete approximate iteration method (NLP)	-
(Ackora-Prah, Gyamerah, Andam, & Gyamfi, 2014) Ackora-Prah et al. (2014)	-	GA
(T. Cui, Cheng, & Bai, 2014) T. Cui et al. (2014)	-	PSO
(Konstantinos Liagkouras & Metaxiotis, 2014) Liagkouras & Metaxiotis (2014)	-	NSGA, SPEA
(Lwin, Qu, & Kendall, 2014) Khin Lwin et al. (2014)	-	MOEA
(Mishra, Panda, & Majhi, 2014) Mishra, et al. (2014)	-	BFO
(Mishra et al., 2014) Sabar & Song (2014)	-	GA
(Tuba & Bacanin, 2014) Tuba & Bacanin (2014a)	-	ABC+FA
(Bacanin & Tuba, 2014) Tuba & Bacanin (2014b)	-	FA
(Bastiani, Cruz-Reyes, Fernandez, Gómez, & Rivera, 2015) Bastiani et al. (2015)	-	ACO
(W. Chen, 2015) W. Chen (2015)	-	ABC
(J. Yu, Ge, Heuveling, Schneider, & Yang, 2015) Ge (2015)	-	ABC
(KAMILI & RIFFI, 2015) Kamili & Riffi (2015)	-	CSO
(Vercher & Bermúdez, 2015) Vercher & Bermúdez (2015)	-	MOEA
(Yin, Ni, & Zhai, 2015) Yin et al. (2015)	-	PSO
(Baykasoğlu et al., 2015) Baykasoğlu et al. (2015)	-	GRASP+QP
(Zhang, 2016) , (Zhang, 2015) P. Zhang (2015), (2016)	CPLEX	-
(Ni, Yin, Tian, & Zhai, 2017) Ni et al. (2016)	-	PSO
(Saborido, Ruiz, Bermúdez, Vercher, & Luque, 2016) Saborido et al. (2016)	-	MOEA
(Suthiwong & Sodanil, 2016) D. Suthiwong & M. Sodanil (2016)	-	iqABC

(K Liagkouras & Metaxiotis, 2016) K. Liagkouras & K. Metaxiotis (2016)	-	EEMPOS
(Jin, Qu, & Atkin) Y. Jin et al. (2016)	CPLEX	-
(Suthiwong & Sodanil, 2016) Suthiwong & Sodanil (2016)	-	ABC
(Jimbo, Ngongo, Andjiga, Suzuki, & Onana, 2017) H.C. Jimbo et al. (2017)	-	GA
(Monge, 2017) J.F. Monge (2017)	QP	-
(Kessaci, 2017) Y. Kessaci (2017)	-	multi-objective continuous GA
(Ahmadi & Davari-Ardakani, 2017) A. Ahmadi et al . (2017)		
(Zhang & Li, 2017) P. Zhang & B. Li (2017)	dynamic programming	-
(Hardoroudi, Keshvari, Kallio, & Korhonen, 2017) N.D. Hardoroudi et al .(2017)	MILP	-
(Can B Kalayci et al., 2017) C.B. Kalayci et al. (2017)	-	ABC
(B. Chen et al., 2017) B. Chen et al. (2017)	-	MOEA
(Can Berk Kalayci, Okkes Ertenlice, Hasan Akyer, & Hakan Aygoren, 2017) C.B. Kalayci et al .(2017)	-	GA
(Ni et al., 2017) Q. Ni et al .(2017)	-	PSO
(K Liagkouras & Metaxiotis, 2018) K. Liagkouras & K. Metaxiotis (2018)	-	MOEA
(Sabar, Turky, Leenders, & Song, 2018) Nasser R. Sabar et al .(2018)	-	MPGA
(Zhang, 2018) P. Zhang (2018)	CPLEX.	-
(Guijarro, 2018) F. Guijarro (2018)	-	GA
(Zhou, Jin, Xiao, Wu, & Liu, 2018) Z. Zhou et al .(2018)	DEA	-
(Branda, Bucher, Červinka, & Schwartz, 2018) M. Branda et al .(2018)	Local Relaxation	-
(Mutunge & Haugland, 2018) P. Mutunge & D. Haugland (2018)	CPLEX	-
(Zhou, Liu, Xiao, Wu, & Liu) Z. Zhou et al. (2018)	DEA	MOEA
(W. Chen, Wang, Gupta, & Mehlawat, 2018) W.chen et al. (2018)	-	FA-GA
(W. Chen & Xu) W.chen et al. (2018)	-	BA–DE
(W. Chen, Li, & Liu, 2018),(W. Chen, Li, Lu, & Liu, 2018) W.chen et al. (2018)	-	ICA-FA, ICA

ABC: Artificial Bee Colony, ACO: Ant Colony Optimization, BFO: Bacterial Foraging Optimization, CSO: Cat Swarm Optimization, DCA: Difference Convex Algorithms, DE: Differential Evolution, EA: Evolutionary Algorithms, EDA: Estimation of Distribution Algorithm, FA: Firefly Algorithm, GA: Genetic Algorithm, GRASP: Greedy Randomized Adaptive Search Procedure, IA: Immune Algorithm IWO: Invasive Weed Optimization, MOEA: Multi-Objective Evolutionary Algorithms, NN: Neural Networks, NPGA: Niched Pareto Genetic Algorithm, NSGA: Non-dominated Sorting Genetic Algorithm, NSGA-II: Non-dominated Sorting Genetic Algorithm II, PBIL: Population Based Incremental Learning, PESA: Pareto Envelope-based Selection Algorithm, PSO: Particle Swarm Optimization, QP: Quadratic Programming, SA: Simulated Annealing, SPEA: Strength Pareto Evolutionary Algorithm, SPEA2: Strength Pareto Evolutionary Algorithm 2, TS: Tabu Search, VEGA: Vector Evaluated Genetic, MPGA: Multi-population Genetic Algorithm, MILP: mixed integer linear programming, DEA: Data Envelopment Analysis, iqABC: improved quick Artificial Bee Colony, EEMPOS: efficiently encoded multi objective portfolio optimization solver, BA: bat algorithm , ICA: imperialist competitive algorithm.

3. Proposed Algorithms

3.1. Solution representation and constraint handling

The feasible solutions are generated by two phases: the selection of k assets from a universe of N available assets and the determination of weights to those selected assets. To select k assets, random real numbers (r_i) between zero and one are generated for each asset. By descending sorting of the value corresponding to each asset, their preference is obtained and k assets which have the highest random real numbers or preference are selected. Eq. (28) shows that after having a set of k assets, we can use their generated random real numbers to compute the corresponding weight values as follows:

$$w_i = \varepsilon_i z_i + \frac{1 - \sum_{i=1}^{N} \varepsilon_i z_i}{\sum_{i=1}^{N} (\delta_i - \varepsilon_i) z_i r_i} (\delta_i - \varepsilon_i) z_i r_i \quad i = 1, \dots, N \tag{28}$$

3.2. Genetic algorithm (GA)

GA is a meta-heuristic algorithm inspired in the Darwin's theory of evolution, using concepts such as reproduction, natural selection, crossover and mutation. An initial random population is first generated. Once the initial population is generated, all the solutions in the current population are ranked based on their objective values and then, the first p_r% of them are transferred to the next iteration. Once the initial population is generated, two solutions as the parents are selected and recombined to generate the offspring by selection and crossover operators respectively. The last operator used in constructing the new generation is the mutation operator. Finally, objective function evaluation of the new generation should be carried out. Fig. 1 presents the pseudo code.

Procedure
Initialization
Fitness evaluation
X_{best} = Update
While *stopping criterion is not met* **do**
  Reproduction (p_r%)
  Crossover (p_c%=1- p_r%)
  Mutation (All offspring produced by *crossover*, p_m)
  Fitness evaluation
  X_{best} = Update
End while

Fig. 1. The pseudo code of GA algorithm.

3.3. Genetic network programming (GNP)

(Katagiri et al., 2002) illustrated that Genetic network programming (GNP) is a development of GP (GP) at the category of meta heuristic algorithms in terms of genotype structures. In the field of meta heuristic algorithms a large number of researches have been made on the evolutionary optimization techniques like Genetic Algorithm and its specially optimization structure by (Holland, 1975), GP with optimization encoding structure by (Koza, 1992) and evolutionary Programming algorithms.

GNP is combined of many and different nodes. The division of nodes is done in two main categories: Judgment node and processing node. Judgment nodes match to some concrete functional nodes which work as judgments. They match almost to GP's elementary functions. Operations are performed by processing nodes. They correspond almost to GP's terminal signs. The individuals are displayed by connecting these nodes with each other. In GNP structure like GP, all tree structure rules are used such as functions and terminal signs which this expression is represented by $J_{1s}, J_{2s}, P_{1s}, P_{2s}$ in the network structure. Both referred nodes (Judgment and Processing nodes) are critical components of GNP. Also, GNP algorithm is beginning with Start node which in turn is important that with the sign *S* is shown which equivalent to root node on GP. GP begins by Start node and then proceeds to assessment and examine the subcategory nodes accordingly. In this article, the rules used for Judgment nodes are if-then kind decision making functions. When GNP is run at first, the operation will begin from the first defined node, and in the following the next node to be performed is specified based on the initial communication from the node that is active in operation. There are two modes, if Judgment node is currently active node; absolutely the next node will be determined by the defined rules at the activated Judgment node. When processing node is performed, surely in these circumstances, the single connection from processing nodes specifies the next nodes. The obvious difference between GNP and other methods is that opposite of their rules, activating nodes on GNP have authority to move to the start node. For example, GP Algorithm doing process review from root node of tree after integrating the interpretation, also PADO algorithm components that illustrated by (Katagiri et al., 2002) have both a Start node and an End node in its structure. When GNP is booted up for the first time; According to the operation that occurs in GNP algorithm, the sequential of GNP system is done according to the circular grid, for this reason, no end nodes are considered in GNP. Genotypic structure of nodes in GNP is schematically shown at Fig. 2. All the variables listed in the genes will be integer. For example this figure introduced gene of node i and more a set of these genes will lead to the introduction of the desired genotype of individuals. In this figure, NT_i Characterizes the Judgment or processing node type, thus $NT_i = 0$ or $NT_i = 1$ respectively represents the node *i* is Judgment or processing node. Also, the parameter ID_i is a unique identification number for nodes. For example, as the parameters NT_i and ID_i are 0 and 1, respectively, it means that the node *i* is J_1. The parameters $C_{i1}, C_{i2}, \dots,$ are defined according to the Judgment rules on node *i*. The genes are connected to each other based on the type of function that the gens are defined in the nodes. The parameter d_i indicated the time required to do processing on the node *i* and d_{ij} is delay time spent at action from node *i* to node C_{ij} .Of course, by adjusting these delay times GNP will have a better performance than the past.

Node 1	Nt_1	ID_1	d_1	C_{11}	d_{11}	C_{12}	d_{12}	...	C_{1n}	d_{1n}
Node 2	Nt_2	ID_2	d_2	C_{21}	d_{21}	C_{22}	d_{22}	...	C_{2n}	d_{2n}
Node 3	Nt_3	ID_3	d_3	C_{31}	d_{31}	C_{32}	d_{32}	...	C_{3n}	d_{3n}
				⋮						
Node i	Nt_i	ID_i	d_i	C_{i1}	d_{i1}	C_{i2}	d_{i2}	...	C_{in}	d_{in}
				⋮						
Node N	Nt_N	ID_N	d_N	C_{N1}	d_{N1}	C_{N2}	d_{N2}	...	C_{Nn}	d_{Nn}

Fig. 2. Genotype structure of node *i*.

This section indicates genetic operators that applied in GNP. At first mutations operator occur in chromosomes, which may result in better future generations. The mutation is that some genes change completely randomly. GNP also solves the problem using this idea So that affects one after another of the genes needs to be change connection randomly from mutation probability of p_m . After that crossover operator must be done. Here, Two-point Crossover is used among existing methods. In this phase; we select two locations of individuals randomly and move the values between these two points. For all of link in two-point crossover, nodes in two positions are randomly selected, and all genes between these two positions are displaced in the parent chromosomes by crossover Probability of p_c between the two parents. According to the presented operators, GNP only uses a number of nodes for evolution; crossover is used to the corresponding nodes chose uniformly in two parents. For this reason, these GNP' operators do not change the functions of any of the nodes; According to their nature, the nodes' connection location is changed in the selected parent by them. In general, GNP focuses only on the connections of evolution and does not carry out any activity on function' nodes.

3.4. Simulated annealing (SA)

SA is a probabilistic meta-heuristic algorithm, proposed by (Černý, 1985; Kirkpatrick, Gelatt, & Vecchi, 1983) , which inspired by the annealing process used in metallurgy based on principles of thermodynamics. It starts the search with an initial solution and iteratively moves towards other existing solutions to find a better solution. The algorithm generates a solution s' using any kind of neighborhood search structure around the candidate solution s . Then, the change between the objective function values, $\Delta_{s,s'} = f(s') - f(s)$, is computed. In case of $\Delta_{s,s'} \le 0$, solution s' is accepted. Otherwise, solution s' is accepted with a probability equal to $exp - \Delta_{s,s'} / T$. However, it is probable to move to an inferior solution, this happens because of reducing the chance of getting stuck or escaping from local optima. This acceptance probability is controlled by $\Delta_{s,s'}$ and temperature (*T*) that is decreased gradually from an initial temperature (T_0) during the SA procedure. Search is carried out for a fixed number of neighborhood searches in each temperature ($n_{\max}$). While the procedure proceeds, the temperature is gradually lowered under the law of cooling schedule. In this study, we use an exponential cooling strategy, $T_{i+1} = \alpha T_i$, where $\alpha \in (0,1)$ is the temperature decreasing rate and T_i represents the temperature of iteration *i* of algorithm. The loop is repeated till a stopping criterion is satisfied. The pseudo-code of the algorithm is presented in Fig. 3.

Initialization: Select an initial solution (s_0), an initial temperature (T_0), Number of neighborhood search in each temperature $n_{\max}$, and termination
Set $T \leftarrow T_0$ and $s \leftarrow s_0$;
Repeat
  Repeat
    Randomly select $s' \in N(s)$;
    Calculate $\Delta_{s,s'} = f(s') - f(s)$
    if $\Delta_{s,s'} \le 0$ then $s \leftarrow s'$;
    else generate random R uniformly in the range (0, 1);
    if $R \le \exp(-\Delta_{s,s'} / T_n)$ then $s \leftarrow s'$;
  Until iteration_countre = *nt*
    Decrease of the temperature *T*;
until the stopping criterion is met

Fig. 3. The pseudo code of SA algorithm.

3.5. Particle swarm optimization (PSO)

PSO is a population based meta-heuristic algorithm, proposed by (Kennedy & Eberhart, 1942), which inspired by the collective behavior of bird flocks. So, the solutions are taken as particles. In each iteration *t*, the ith particle is associated with two vectors, i.e., a position vector or a solution $x_{i,t}$ and a velocity or movement vector$v_{i,t}$. The personal best position of the particle $i$is $Pbest_i$, while $Gbest$ is referred to as global best position The new position, the velocity of the particle and inertia weight at iteration t are given as:

$$v_{i,t+1} = \omega_i . v_{i,t} + c_1 . r_1 . (Pbest_i - x_{i,t}) + c_2 . r_2 . (Gbest - x_{i,t}) \quad (29)$$

$$x_{i,t+1} = x_{i,t} + v_{i,t+1} \quad (30)$$

$$\omega_t = \omega_{max} - \frac{(\omega_{max} - \omega_{min}) * t}{t_{max}} \quad (31)$$

Where c_1 and c_2 are the cognitive, and social weights, respectively; r_1 and r_2 are two random numbers generated in the range (0, 1). The algorithm is illustrated in Fig. 4.

```
Procedure
Initialization
Fitness evaluation
Gbest = Update
Pbest = Update
While stopping criterion is not met do
    Velocity calculation
    Move
    Fitness evaluation
    Gbest = Update
    Pbest = Update
End while
```

Fig. 4. The pseudo code of PSO algorithm.

3.6. Electromagnetism-like algorithm (EM)

The electromagnetism-like algorithm (EM) is a population-based meta-heuristic algorithm proposed by (Birbil & Fang, 2003). EM imitates the attraction-repulsion mechanism of electromagnetic charged particles and, these particles will gradually move towards the global optimal solution through some iterative operations. The particles and their charge are taken as the solutions and objective function value in a population. The magnitude of the attraction or repulsion force among these particles depended to their objective function value. The move of particles is specified by the resultant force which is the summation of all the forces exerted by on the particle by other particles. The resultant force also is based on Coulomb's law, which is directly proportional to the product of their individual charges and inversely proportional to the square distance between them.

There are four phases in EM: initialization, local search, calculation of resultant force and determination of moving distance, which are represented as Initialize (*popsize*), Local search (*LSITER*, δ), Calculate Forces and Move, respectively. The pseudo-code of the EM is presented in Fig. 5.

```
Initialize (popsize)
iteration ← 1
While termination criterion are not satisfied do
    Local search (LSITER, δ)
    Calculate Forces
    Move
    iteration ← iteration + 1
End while
```

Fig. 5. The pseudo code of EM algorithm.

Local search:

The procedure that generates a temporary neighborhood then finds its related objective value. This new neighborhood solution replaced when its objective value is better than the current one. This procedure is shown in Fig. 6.

```
counter ←1
for i = 1 to popsize do
    for k = 1 to n do
        λ1 ← U (0, 1)
        while counter <LSITER do
            Y ← X^i
            λ2 ← U (0, 1)
            if λ1 > 0.5 then
                Yk ← Yk + λ2 δ
            else
                Yk ← Yk − λ2 δ
            end if
            if f (Y) < f (X^i) then
                X^i ← Y
                counter ← LSITER − 1
            end if
            counter ←counter + 1
        end while
    end for
end for
```

Fig. 6. The pseudo code of Local (*LSITER*,δ).

Total forces Computation:

Obviously, q^i is positive and a particles with better objective value has higher charge. In addition, a good solution attracts others in its direction to converge its valley. On the contrary, a bad solution discourages the others to move toward its region by repulsion. The

charge q^i, the components $F_j^i (j \in J)$ of the resultant force exerted on each solution X^i are obtained as follow:

$$q^i = \exp\left(-n \frac{f\left(X^i\right) - f\left(X^{best}\right)}{\sum_{k=1}^{popsize} \left(f\left(X^i\right) - f\left(X^{best}\right)\right)} \right), \quad i = 1, ..., popsize \tag{32}$$

Where, X^{best} is the current best solution in the population.

$$F_j^i = \sum_{\substack{k=1 \\ k \neq i}}^{popsize} \left\{ \begin{array}{ll} \left(x_j^k - x_j^i\right) \dfrac{q^i q^k}{\left\| x^k - x^i \right\|^2} & if \quad f\left(X^k\right) < f\left(X^i\right) \\ \left(x_j^i - x_j^k\right) \dfrac{q^i q^k}{\left\| x^k - x^i \right\|^2} & if \quad f\left(X^k\right) \geq f\left(X^i\right) \end{array} \right\}, \quad i = 1, ..., popsize, \ j \in J \tag{33}$$

Where

$$\left\| x^k - x^i \right\| = \left(\sum_{j \in J} \left(x_j^k - x_j^i \right)^2 \right)^{1/2} \tag{34}$$

Movement procedure:

After evaluating F^i, the all particles but the best are moved in the direction of the force by a random step length λ, uniformly distributed between 0 and 1. The calculation for the movement is as follows:

$$x_j^i = \left\{ \begin{array}{ll} x_j^i + \lambda \dfrac{F_j^i}{\left\| F^i \right\|} \left(u_j - x_j^i \right) & if \quad F_j^i > 0 \\ x_j^i + \lambda \dfrac{F_j^i}{\left\| F^i \right\|} \left(x_j^i - l_j \right) & if \quad F_j^i \leq 0 \end{array} \right\} \quad i = 1, ..., popsize, \ j \in J \tag{35}$$

Where

$$\left\| F^i \right\| = \left(\sum_{j \in J} F_j^{i^2} \right)^{1/2} \tag{36}$$

Where $(u_j - x_j^i)$ and $(x_j^i - l_j)$ denotes the amount of feasible movement toward upper boundary (u_j) and lower boundary (l_j).

3.7. Hybrid algorithms

One of the key issues in developing a hybrid meta-heuristic algorithm that attempt to find optimum solutions is the diversification capability to explore new unvisited regions of the solution space. Some meta-heuristic algorithms suffer from loss of variation results in low exploration, leading the search to converge prematurely to a local optimum. Without such diversification mechanism, these algorithms may become trapped in a local optimum valley, eliminating the chance of exploring the global optimum valleys.

In our procedure the diversification mechanism is applied when the best objective function does not change during the number of pre-specified consecutively iterations (*no change*) in the algorithm. Diversification refers to perturbation of the current population by removing a number of inferior solutions and replacing new randomly generated solutions.

4. Computational Experiments

4.1. Instance

In this part, the computational results on various risk measures are noticed. A historical daily data collected was applied in the Tehran with costs' data of 50 companies. The cardinality constraint k is set from 10 to 30 which

expanded by 10 per time. The problem resolved by using EM, HEM, GA, HGA, PSO, HPSO, GNP, HGNP, SA, HSA and coding in Matlab 2015b and running on a personal computer. The companies of the suggested portfolio optimization for various number of assets ($k = 10\ to\ 30$) are reported at Table 2. As Table 2 indicates this research includes of 50 companies from Tehran stock market. The table consists of two columns; numbers and abbreviated name of company at stock market.

Table 2
The fifty companies selected at Tehran stock market.

Stock. no	Stock. Abbreviation name	Stock. no	Stock. Abbreviation name	Stock. no	Stock. Abbreviation name
s_1	vnowin	s_{18}	Kasafa	s_{35}	sshomal
s_2	vmelli	s_{19}	Fanaval	s_{36}	sshargh
s_3	Vmellat	s_{20}	Falooleh	s_{37}	sroud
s_4	Vmaaden	s_{21}	Fajoosh	s_{38}	sdour
s_5	Vliz	s_{22}	Fajr	s_{39}	skhazar
s_6	Vsana	s_{23}	Fapanta	s_{40}	skhash
s_7	Vsepah	s_{24}	Faahwaz	s_{41}	steranh
s_8	Vpars	s_{25}	Ghadam	s_{42}	steran
s_9	Vbimeh	s_{26}	Ghabehnoosh	s_{43}	sepaha
s_{10}	Vbouali	s_{27}	Ghalber	s_{44}	deyran
s_{11}	Vbeshahr	s_{28}	Shoush	s_{45}	delghamah
s_{12}	Valbarah	s_{29}	Shpars	s_{46}	delghama
s_{13}	Mdaran	s_{30}	Sfars	s_{47}	deler
s_{14}	Lmokhaber	s_{31}	Sfar	s_{48}	dekimi
s_{15}	Lkhazar	s_{32}	Sgharb	s_{49}	dekosar
s_{16}	Kafara	s_{33}	Ssoufi	s_{50}	defara
s_{17}	Kadama	s_{34}	Ssefha	…	…

4.2. Performance metrics

In this paper, four performances metric introduced to compare efficiency and effectiveness of suggested algorithms as bellow:

1) Quality metric (QM): To calculate the value of this criterion, first, the net non-dominated solutions (NDS) are generated by a set of all non-dominated solutions obtained from all algorithms (whose members should be also non dominated in relation to one another) and then the percentage of non-dominated solutions of each algorithm in NDS to the number of NDS is calculated. The larger the number, the better the performance of the algorithm will be.

2) Mean ideal distance (MID): This measure presents the closeness between Pareto solution and ideal point in NDS which can be shown as Eq. (37).

$$MID = \frac{\sum_{i=1}^{n}\sqrt{\left(\frac{f_{1,i}-f_{1,best}}{f_{1,worst}-f_{1,best}}\right)^2+\left(\frac{f_{2,i}-f_{2,best}}{f_{2,worst}-f_{2,best}}\right)^2}}{n} \quad (37)$$

In the above equation, n is the number of Pareto solutions while $f_{j,best}$ and $f_{j,worst}$ are the best and the worst value of the objective function among all the algorithms objective functions values. The less value of MID indicated better algorithm's performance.

3) Spacing metrics (SM): This metrics demonstrate the uniformity of Pareto solutions in solution space:

$$SM = \frac{\sum_{i=1}^{n}|\bar{d}-d_i|}{(n-1)\bar{d}} \quad (38)$$

In Eq. (38) d_i is the Euclidean distance of two neighboring Pareto solutions in solution space, $\bar{d}$ is the average of all d_i. Lower amounts of SM shows more uniform non-dominates solutions and therefore a better algorithm.

4) Diversification Metric (DM): This metric shows the spread of the Pareto solutions set and is measured by Eq. (39).

$$DM = \sqrt{\left(\frac{f_{1,max}-f_{1,min}}{f_{1,worst}-f_{1,best}}\right)^2+\left(\frac{f_{2,max}-f_{2,min}}{f_{2,worst}-f_{2,best}}\right)^2} \quad (39)$$

Where $f_{1,max}$ and $f_{1,min}$ are the maximum and the minimum value of the objective function from an algorithm. The higher value of DM demonstrated better performance.

4.3. Parameter setting

It is absolutely clear that the different levels of the parameters by a random search algorithm strongly affect the quality of the solutions. Most users adjust parameters manually by using the reference values that refers on the previous literatures. So, we investigate the behavior of the suggested algorithms in different levels of parameters and find the best level of these parameters and operators by using trial-and-error method. The optimum levels of parameters and operators used in all algorithms are shown in Table 3.

Table 3
Best level of parameters for the algorithms.

EM & HEM	GA & HGA	GNP & HGNP	PSO & HPSO	SA & HSA
popsize = 30;	*popsize* = 80;	*popsize* = 301;	*popsize* = 100;	T_0 = 1000;
LSITER = 10;	pc = 0.8;	crossover size=120;	ω_{max} = 0.9;	α = 0.98;
δ = 0.05;	pm = 0.4;	mutation size=180;	ω_{min} = 0.4;	$n_{\max}$ = 500;
no_change = 20;	*no_change* = 30;	Pm = 0. 1;	c_1 = 2;	*no_change* = 50;
		Pm = 0. 1;	c_2 = 2;	
		NO of nodes=60;	*no_change* = 50;	
		no_change = 20;		

4.4. Experimental results

To be fair in comparison, we consider the same time as a stop criterion for all algorithms. Also, the search time for all algorithms is considered to be 1500 seconds. As shown in Fig. 7, the convergence is suitable for all algorithms. So, by increasing the time, there is not any improvement in solution will be observed. In other words, each algorithm achieved to the best solution.

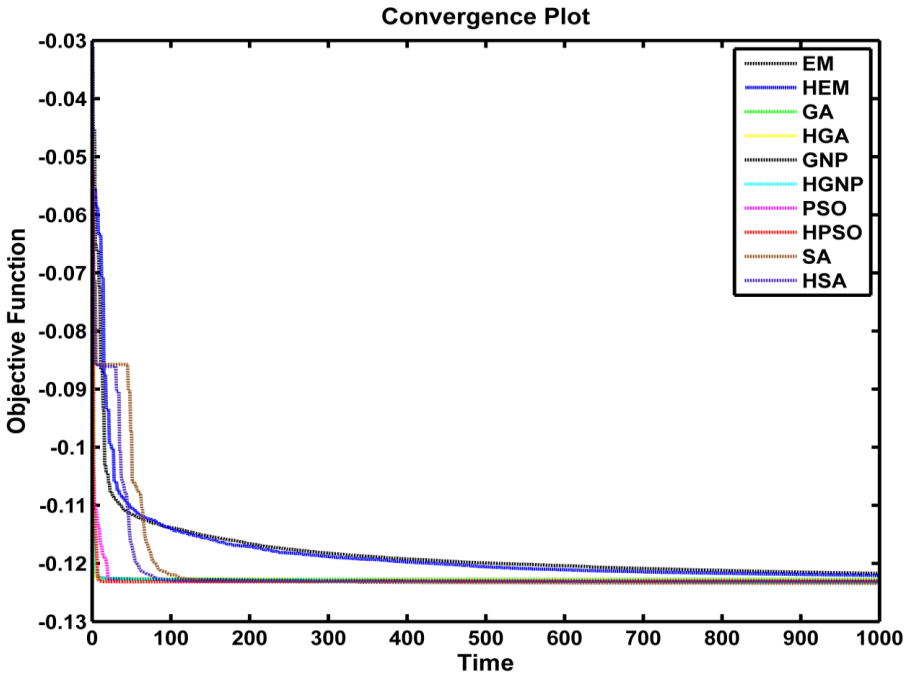

Fig. 7. Convergence plot for the proposed algorithms.

Due to the high competition between algorithms to achieving the optimal solutions, we plot the pareto optimal solutions at Figs. 8-10. Each figure plotted at cost function ($Fcn,1-4$) and *k* number ($k = 10,20,30$), respectively.

As shown in Figs. 8-10, due to existing competition at multi objective problem, visual judgment about the performance of each algorithm is so difficult. For this reason, quantitative and qualitative metrics should be investigated. So in the following, by using the evaluation metrics and analysis of variance, we discuss about performance of the algorithms.

According to the cardinality constraint at different cost functions, Tables 4-15 showed performance evaluation metrics for all algorithms at bellow by different k number ($k = 10, 20, 30$). As previously mentioned, (QM, DM) and (MID, SM) respectively are positive and negative metrics. The optimum number is marked at each of the following tables. As shown in the results, by considering the cost functions and changing the k number, the different metrics indicate the superiority of the algorithms.

So, we find all possible solutions in the optimal portfolio selection based on the level of acceptable risk by investors. For example when f_1 is a cost function and $k = 10$, EM has a good performance by negative metrics. Also in the same conditions, GNP and SA respectively have best performance on QM and DM metrics.

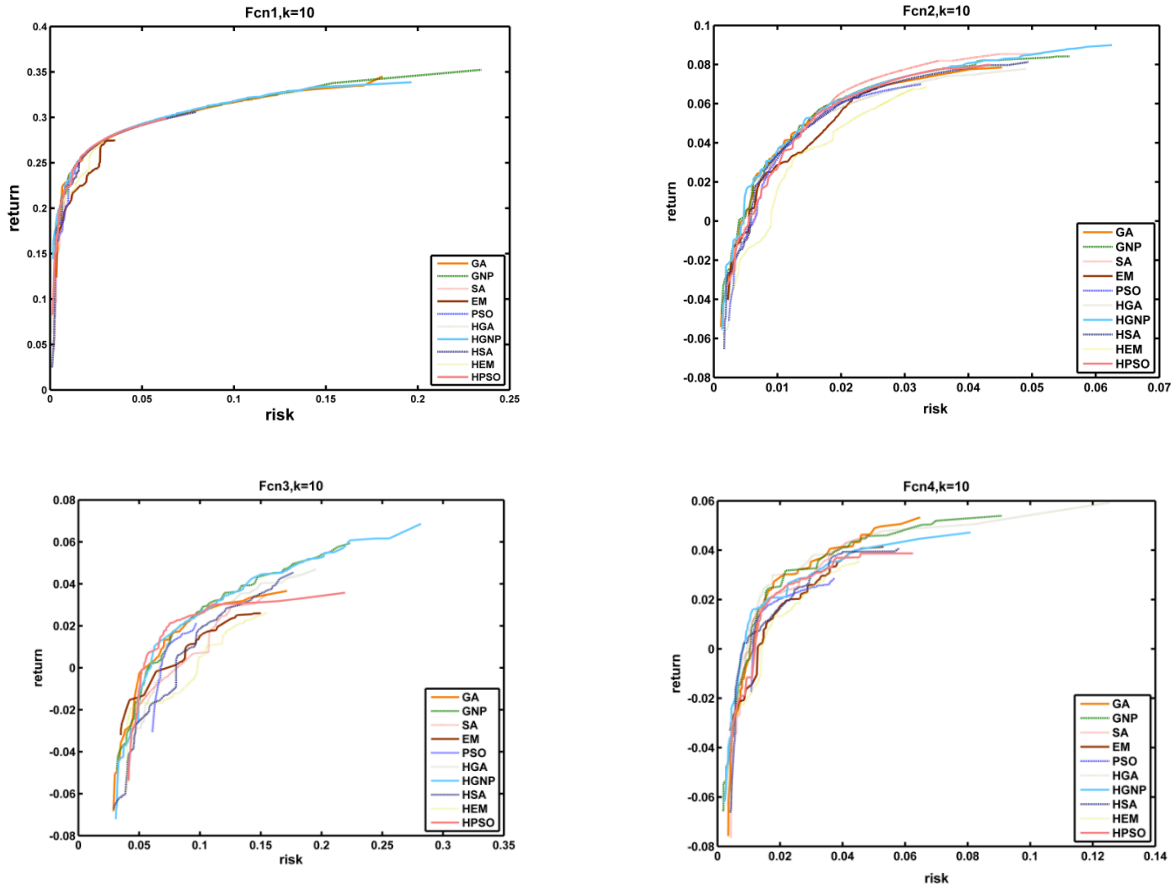

Fig. 8. Comparison of 10 algorithms on efficient frontiers, *Fcn* 1-4, *K*=10.

Table 4
The performance evaluation metrics for different algorithms at f_1 and *K*=10.

Metrics	Algorithms									
	EM	HEM	GA	HGA	GNP	HGNP	PSO	HPSO	SA	HSA
QM	0	0	0.001	0.0003	0.0045	0.0179	0.3777	0.0034	**0.5951**	0.0001
MID	**0.6824**	0.6998	0.79	0.7856	0.7916	0.7952	0.7925	0.774	0.8214	0.7947
SM	**1.2372**	1.306	1.7075	1.7329	1.6716	1.4349	1.4294	1.5273	1.4582	1.4065
DM	0.3392	0.3465	1.0165	1.0312	**1.1571**	1.0253	0.4875	0.7122	0.8776	0.9214

Table 5
The performance evaluation metrics for different algorithms at f_1 and *K*=20.

Metrics	Algorithms									
	EM	HEM	GA	HGA	GNP	HGNP	PSO	HPSO	SA	HSA
QM	**0.7093**	0	0.0116	0.0233	0.25	0	0	0.0058	0	0
MID	0.0207	**0.0003**	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0004	0.0003
SM	1.9342	1.2998	1.6129	1.7135	1.52	1.4455	1.918	1.8693	**1.1606**	1.3637
DM	**1.4141**	0.0003	0.0003	0.0004	0.0003	0.0004	0.0002	0.0003	0.0005	0.0004

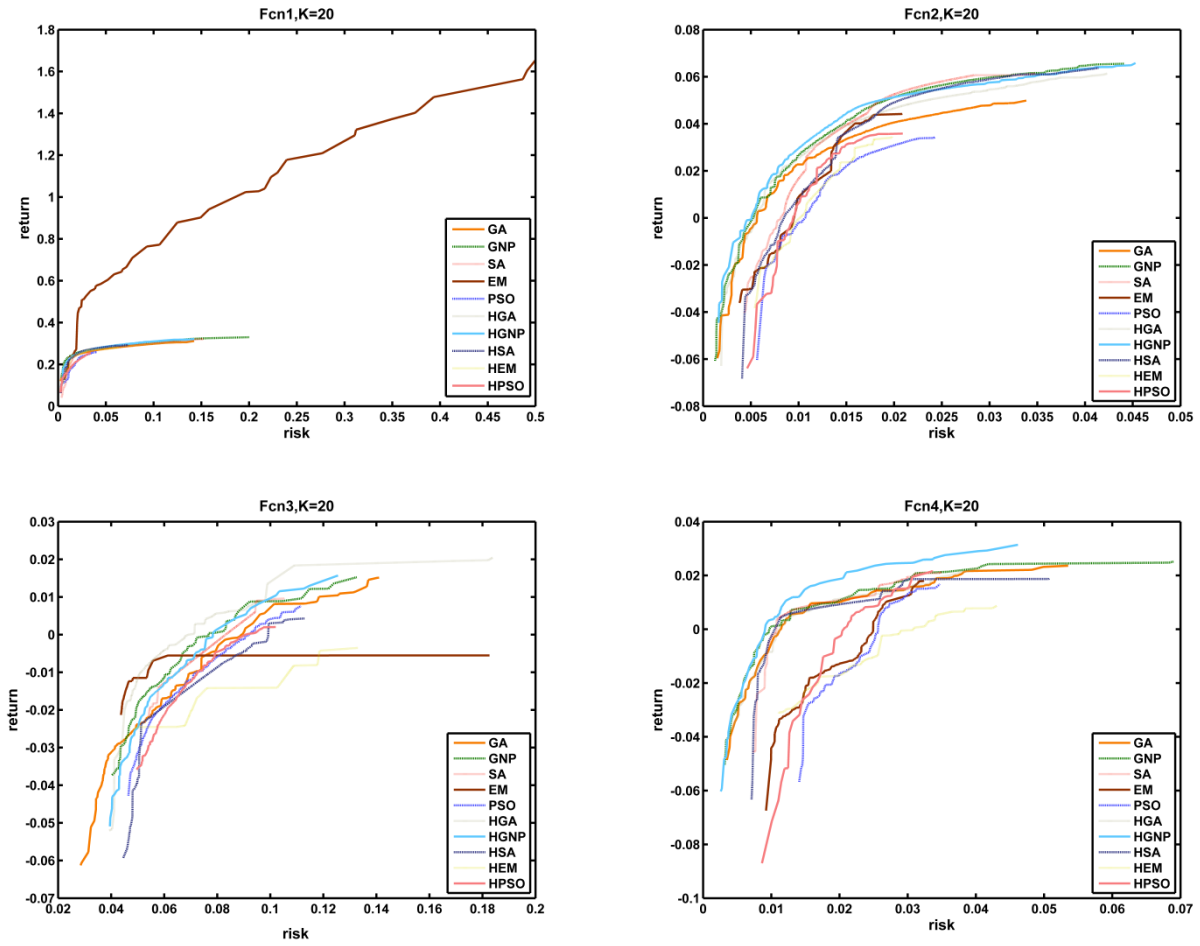

Fig. 9. Comparison of 10 algorithms on efficient frontiers, *Fcn* 1-4, *K*=20.

Table 6
The performance evaluation metrics for different algorithms at f_1 and *K*=30.

Metrics	Algorithms									
	EM	HEM	GA	HGA	GNP	HGNP	PSO	HPSO	SA	HSA
QM	0	0.0241	0.0002	0.0002	0.0389	0.0039	0	0	0	**0.9327**
MID	0.6547	0.639	0.7108	0.715	0.7069	0.7086	**0.5429**	0.6873	0.6897	0.7314
SM	1.4877	1.4336	1.6562	1.5696	1.447	1.4904	1.924	1.763	**1.3138**	1.4238
DM	0.773	0.7414	**1.5061**	1.181	1.1161	1.3233	0.693	0.853	1.0654	1.0122

Table 7
The performance evaluation metrics for different algorithms at f_2 and *K*=10.

Metrics	Algorithms									
	EM	HEM	GA	HGA	GNP	HGNP	PSO	HPSO	SA	HSA
QM	0.0002	0	0.0187	0.0004	0.0056	0.0085	0	0	**0.9665**	0
MID	**0.7535**	0.8515	0.921	0.9063	0.9077	0.9282	0.9	0.9498	0.9042	0.8519
SM	1.2315	1.3806	1.6731	1.6773	1.3473	1.5225	1.7544	1.5718	**1.0705**	1.1073
DM	0.8416	0.8969	1.1678	1.2197	1.2744	**1.4191**	0.9722	1.033	1.1245	1.2813

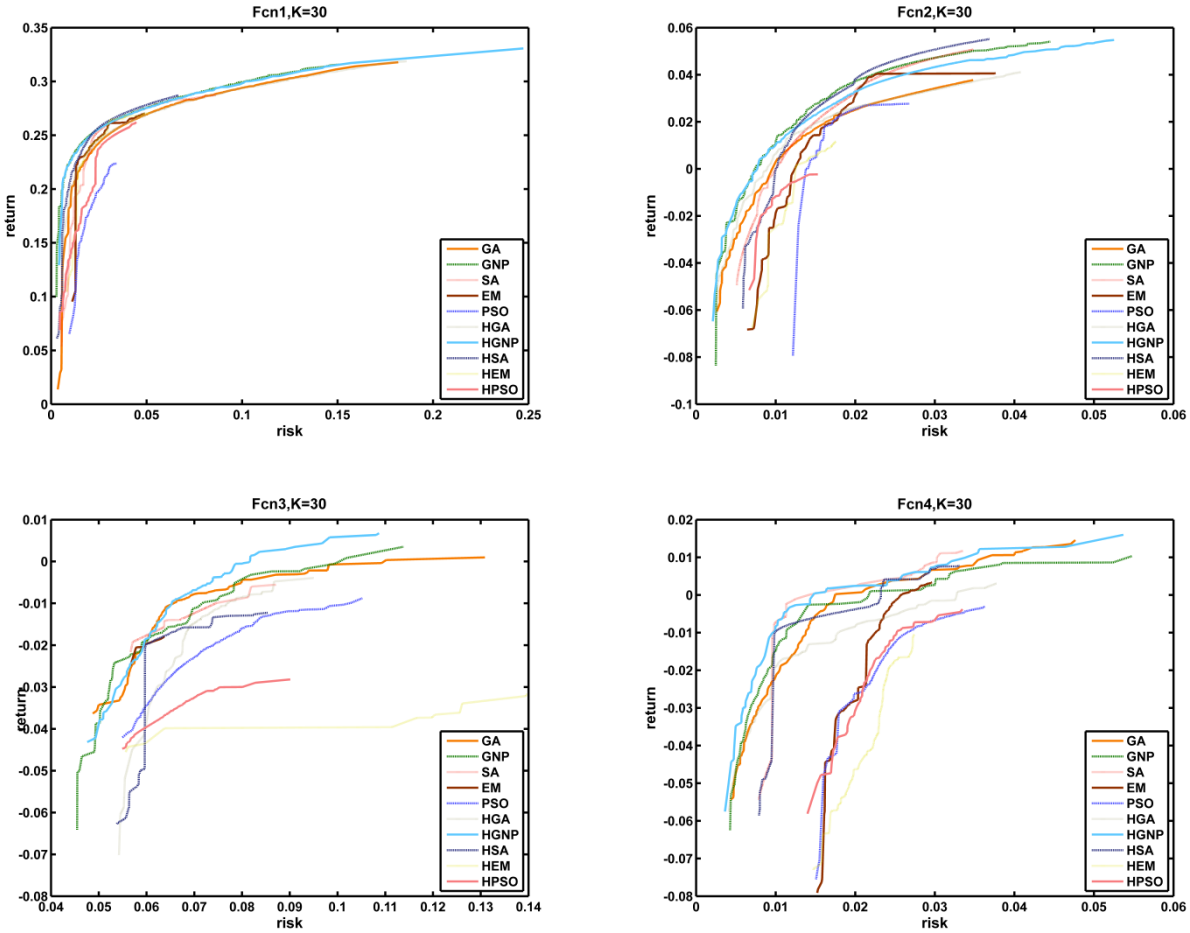

Fig. 10. Comparison of 10 algorithms on efficient frontiers, *Fcn* 1-4, *K*=30.

Table 8
The performance evaluation metrics for different algorithms at f_2 and *K*=20.

Metrics	Algorithms									
	EM	HEM	GA	HGA	GNP	HGNP	PSO	HPSO	SA	HSA
QM	0	0	0	0.0002	0.0064	0.1229	0	0	**0.8692**	0.0012
MID	0.7773	**0.6285**	0.8512	0.9511	0.9501	0.9202	0.8678	0.786	0.9079	0.959
SM	1.3457	1.1882	1.2437	1.352	1.3307	1.2682	1.9198	1.8692	**1.1322**	1.2466
DM	0.7438	0.6613	1.1357	1.3441	**1.3942**	1.3207	0.8599	0.8711	1.0617	1.3444

Table 9
The performance evaluation metrics for different algorithms at f_2 and *K*=30.

Metrics	Algorithms									
	EM	HEM	GA	HGA	GNP	HGNP	PSO	HPSO	SA	HSA
QM	0	0	0	0	0.113	0.0288	0	0	0	**0.8582**
MID	0.9375	0.6833	0.9659	0.9619	1.0964	1.0294	0.9787	**0.5976**	0.8998	0.946
SM	1.4798	1.5886	1.4028	1.3497	1.4943	1.2985	1.9632	1.6386	**1.0056**	1.0927
DM	1.2776	0.7248	1.2375	1.3966	1.6668	**1.7601**	0.9862	0.4791	1.1948	1.3057

Table 10
The performance evaluation metrics for different algorithms at f_3 and *K*=10.

Metrics	Algorithms									
	EM	HEM	GA	HGA	GNP	HGNP	PSO	HPSO	SA	HSA
QM	**0.7409**	0	0.0763	0	0.009	0.1231	0	0.0485	0	0.0023
MID	**0.3989**	0.4051	0.4944	0.6355	0.5495	0.6054	0.5852	0.58	0.5119	0.4248
SM	1.6752	1.5409	1.519	1.7432	1.4127	1.5264	1.321	1.2795	1.6578	**1.2191**
DM	0.6225	0.7708	0.9517	0.8629	1.093	**1.4288**	0.4056	0.9607	0.6488	1.0089

Table 11
The performance evaluation metrics for different algorithms at f_3 and K=20.

Metrics	Algorithms									
	EM	HEM	GA	HGA	GNP	HGNP	PSO	HPSO	SA	HSA
QM	**0.4665**	0	0.1423	0.3841	0.0004	0	0	0	0.0067	0
MID	0.584	0.5413	**0.4403**	0.6781	0.5862	0.5888	0.5945	0.5888	0.7496	0.5724
SM	1.8306	1.7073	1.3791	1.7464	1.4702	1.4989	0.8514	0.827	**0.7664**	1.1722
DM	0.9146	0.6023	1.1824	**1.2852**	0.875	0.9851	0.7446	0.573	0.5042	0.8936

Table 12
The performance evaluation metrics for different algorithms at f_3 and K=30.

Metrics	Algorithms									
	EM	HEM	GA	HGA	GNP	HGNP	PSO	HPSO	SA	HSA
QM	0	0	0.1243	0	0.051	0.1297	0	0	**0.6951**	0
MID	0.651	0.5349	0.8164	0.8006	0.6294	0.835	0.6662	**0.421**	0.7482	0.5026
SM	1.6957	1.6403	1.5958	1.5013	1.2935	1.3107	1.0875	1.3496	**1.0748**	1.3584
DM	0.1492	1.3442	1.4014	1.1349	**1.442**	1.196	0.9228	0.6033	0.532	0.8699

Table 13
The performance evaluation metrics for different algorithms at f_4 and K=10.

Metrics	Algorithms									
	EM	HEM	GA	HGA	GNP	HGNP	PSO	HPSO	SA	HSA
QM	0.05	0	0.0917	**0.3417**	0.0833	0.2667	0	0	0.075	0.0917
MID	0.6131	0.6388	0.6436	0.5948	0.5789	**0.538**	0.6979	0.6835	0.7186	0.6223
SM	1.011	1.1463	1.023	1.0705	0.9995	**0.8884**	0.9816	1.0807	1.6125	1.1894
DM	0.716	0.5926	1.1463	**1.3926**	1.199	1.0822	0.4258	0.8254	1.0441	0.9603

Table 14
The performance evaluation metrics for different algorithms at f_4 and K=20.

Metrics	Algorithms									
	EM	HEM	GA	HGA	GNP	HGNP	PSO	HPSO	SA	HSA
QM	0	0	0	0	0.2619	**0.7381**	0	0	0	0
MID	0.8297	0.7443	0.7947	**0.6418**	0.6818	0.6842	0.7635	0.7584	0.8386	0.854
SM	1.3627	1.0794	0.8329	0.7356	0.7717	**0.6913**	0.9259	0.9836	1.3891	1.592
DM	1.0723	0.8532	1.3925	1.0543	**1.7214**	1.4142	0.9306	1.3185	0.9654	1.3433

Table 15
The performance evaluation metrics for different algorithms at f_4 and K=30.

Metrics	Algorithms									
	EM	HEM	GA	HGA	GNP	HGNP	PSO	HPSO	SA	HSA
QM	0	0	0.0034	0	0	0.0387	0	0	**0.9512**	0.0067
MID	0.7085	**0.5265**	0.6541	0.5614	0.6009	0.6835	0.735	0.6573	0.7091	0.7109
SM	1.225	1.203	0.7274	**0.6145**	0.7308	0.7425	1.2935	0.9641	1.6397	1.2991
DM	1.1582	0.8877	1.2754	1.0637	1.4132	**1.4142**	1.0727	0.8351	1.0354	1.0331

Because the scale of performance evaluation metrics is different, they must be normalized. In order to make the comparison easy and comprehensive, we use the normalized metrics as a common measure of comparing algorithms for each objective function and k as follow:

$$C_1 = QM \tag{40}$$

$$C_{2,j} = \frac{(MID_j)^{-1}}{\sum_{i=1}^{m}(MID_j)^{-1}} \qquad j = 1, \dots, m \tag{41}$$

$$C_{3,j} = \frac{(SM_j)^{-1}}{\sum_{i=1}^{m}(SM_j)^{-1}} \qquad j = 1, \dots, m \tag{42}$$

$$C_{4,j} = \frac{DM_j}{\sum_{i=1}^{m} DM_j} \qquad j = 1, \dots, m \tag{43}$$

Comparison of all algorithms at 4 normalized metrics has been showed in Tables 16-19 for different k number ($k = 10, 20, 30$). Also after each table, mean plot and least significant difference (LSD) intervals at the 95 % confidence level for all the algorithms at every metrics shown in Figs. 12-15.

To verify the statistical validity of the results, we carry out the analysis of variance (ANOVA) technique to accurately analyze the results. For example, the mean plot and LSD intervals at the 95 % confidence level for all the algorithms at c_1 metric are shown in Fig. 12. As it is illustrated, SA provides statistically better results than EM and the others. Also in Table 16, according to the twelve

solutions, EM, HGA, HGNP, SA, HSA respectively have 3, 1, 1, 5, 2 times good performance at c_1 metric. Meanwhile in average mode, SA has a best performance at c_1 metric that showed by mean plot in Fig. 12. Also among of all hybrids, GA and GNP have good performance.

Table 16
Comparison of all algorithms at C_1 metric.

Problems	Algorithms									
	EM	HEM	GA	HGA	GNP	HGNP	PSO	HPSO	SA	HSA
f1k10C1	0	0	0.001	0.0003	0.0045	0.0179	0.3777	0.0034	**0.5951**	0.0001
f1k20 C1	**0.7093**	0	0.0116	0.0233	0.25	0	0	0.0058	0	0
f1k30 C1	0	0.0241	0.0002	0.0002	0.0389	0.0039	0	0	0	**0.9327**
f2k10 C1	0.0002	0	0.0187	0.0004	0.0056	0.0085	0	0	**0.9665**	0
f2k20 C1	0	0	0	0.0002	0.0064	0.1229	0	0	**0.8692**	0.0012
f2k30 C1	0	0	0	0	0.113	0.0288	0	0	0	**0.8582**
f3k10 C1	**0.7409**	0	0.0763	0	0.009	0.1231	0	0.0485	0	0.0023
f3k20 C1	**0.4665**	0	0.1423	0.3841	0.0004	0	0	0	0.0067	0
f3k30 C1	0	0	0.1243	0	0.051	0.1297	0	0	**0.6951**	0
f4k10 C1	0.05	0	0.0917	**0.3417**	0.0833	0.2667	0	0	0.075	0.0917
f4k20 C1	0	0	0	0	0.2619	**0.7381**	0	0	0	0
f4k30 C1	0	0	0.0034	0	0	0.0387	0	0	**0.9512**	0.0067
Average	0.1639	0.002	0.0391	0.0625	0.0687	0.1232	0.0315	0.0048	**0.3466**	0.1577

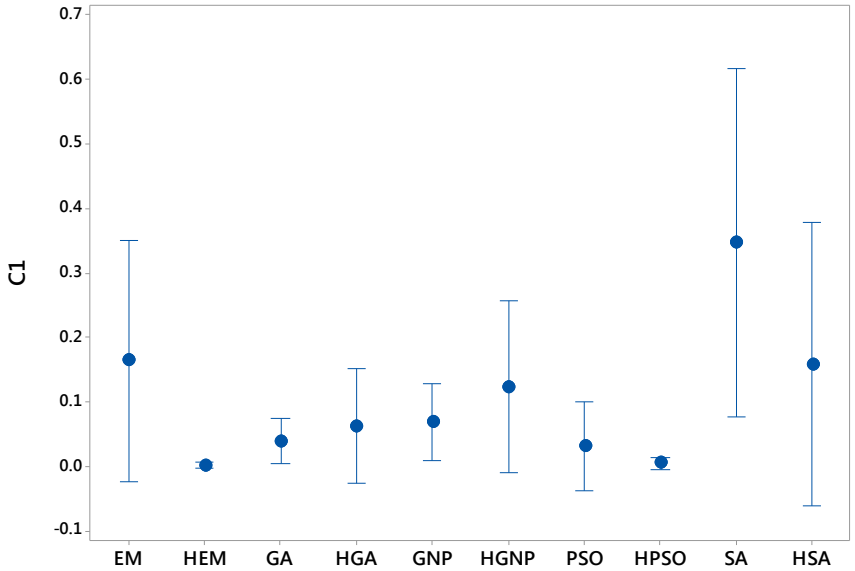

Fig. 11. Mean plot and LSD intervals at the 95 % confidence level for c_1 metric.

Table 17
Comparison of algorithms at c_2 metric.

Problems	Algorithms									
	EM	HEM	GA	HGA	GNP	HGNP	PSO	HPSO	SA	HSA
f1k10C2	**0.1129**	0.1101	0.0975	0.098	0.0973	0.0969	0.0972	0.0995	0.0938	0.0969
f1k20C2	0.0018	**0.1239**	0.1108	0.1095	0.1081	0.1065	0.1151	0.1191	0.0975	0.1077
f1k30C2	0.1029	0.1055	0.0948	0.0943	0.0953	0.0951	**0.1241**	0.0981	0.0977	0.0921
f2k10C2	**0.1173**	0.1038	0.096	0.0975	0.0974	0.0952	0.0982	0.0931	0.0978	0.1038
f2k20C2	0.1089	**0.1347**	0.0995	0.089	0.0891	0.092	0.0976	0.1077	0.0932	0.0883
f2k30C2	0.094	0.1289	0.0912	0.0916	0.0803	0.0856	0.09	**0.1474**	0.0979	0.0931
f3k10C2	**0.1267**	0.1248	0.1022	0.0795	0.092	0.0835	0.0864	0.0872	0.0988	0.119
f3k20C2	0.0997	0.1076	**0.1323**	0.0859	0.0994	0.0989	0.098	0.0989	0.0777	0.1017
f3k30C2	0.0968	0.1178	0.0772	0.0787	0.1001	0.0755	0.0946	**0.1497**	0.0842	0.1254
f4k10C2	0.1025	0.0984	0.0977	0.1057	0.1086	**0.1168**	0.09	0.0919	0.0875	0.101
f4k20C2	0.0907	0.1011	0.0947	**0.1173**	0.1104	0.11	0.0986	0.0993	0.0898	0.0881
f4k30C2	0.0914	**0.123**	0.099	0.1153	0.1077	0.0947	0.0881	0.0985	0.0913	0.0911
Average	0.0955	**0.115**	0.0994	0.0969	0.0988	0.0959	0.0982	0.1075	0.0923	0.1007

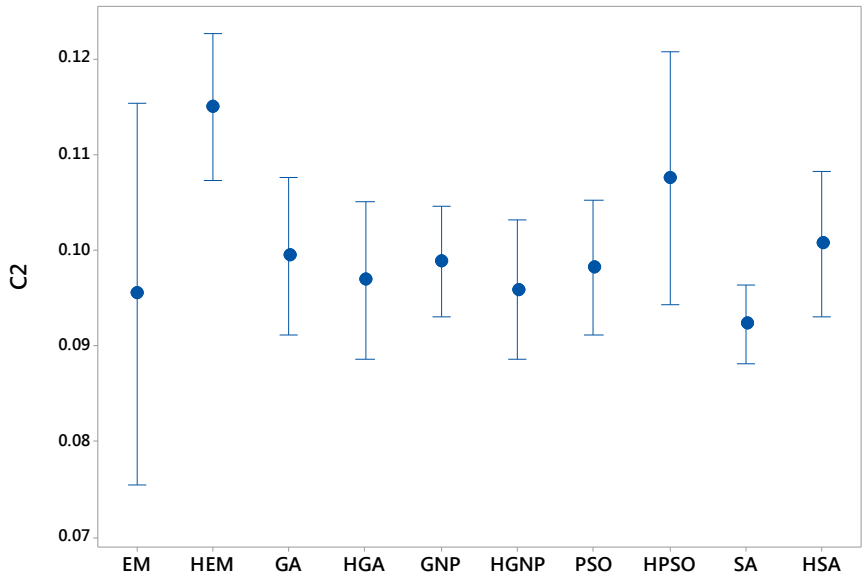

Fig. 12. Mean plot and LSD intervals at the 95 % confidence level for c_2 metric.

Table 18
Comparison of algorithms at c_3 metric.

Problems	Algorithms									
	EM	HEM	GA	HGA	GNP	HGNP	PSO	HPSO	SA	HSA
f1k10C3	**0.1192**	0.1129	0.0863	0.0851	0.0882	0.1027	0.1031	0.0965	0.1011	0.1048
f1k20C3	0.0796	0.1185	0.0955	0.0899	0.1014	0.1066	0.0803	0.0824	**0.1327**	0.113
f1k30C3	0.103	0.1069	0.0926	0.0977	0.1059	0.1029	0.0797	0.087	**0.1167**	0.1077
f2k10C3	0.1132	0.101	0.0833	0.0831	0.1035	0.0916	0.0795	0.0887	**0.1302**	0.1259
f2k20C3	0.1003	0.1136	0.1085	0.0998	0.1014	0.1064	0.0703	0.0722	**0.1192**	0.1083
f2k30C3	0.0935	0.0871	0.0986	0.1025	0.0926	0.1065	0.0705	0.0844	**0.1376**	0.1266
f3k10C3	0.0877	0.0954	0.0968	0.0843	0.1041	0.0963	0.1113	0.1149	0.0887	**0.1206**
f3k20C3	0.0657	0.0704	0.0872	0.0688	0.0818	0.0802	0.1412	0.1454	**0.1568**	0.1026
f3k30C3	0.0802	0.0829	0.0852	0.0906	0.1051	0.1037	0.125	0.1007	**0.1265**	0.1001
f4k10C3	0.1063	0.0938	0.1051	0.1004	0.1075	**0.121**	0.1095	0.0994	0.0667	0.0904
f4k20C3	0.0705	0.089	0.1154	0.1306	0.1245	**0.139**	0.1038	0.0977	0.0692	0.0604
f4k30C3	0.0773	0.0788	0.1302	**0.1542**	0.1297	0.1276	0.0732	0.0983	0.0578	0.0729
Average	0.0914	0.0959	0.0987	0.0989	0.1038	0.107	0.0956	0.0973	**0.1086**	0.1028

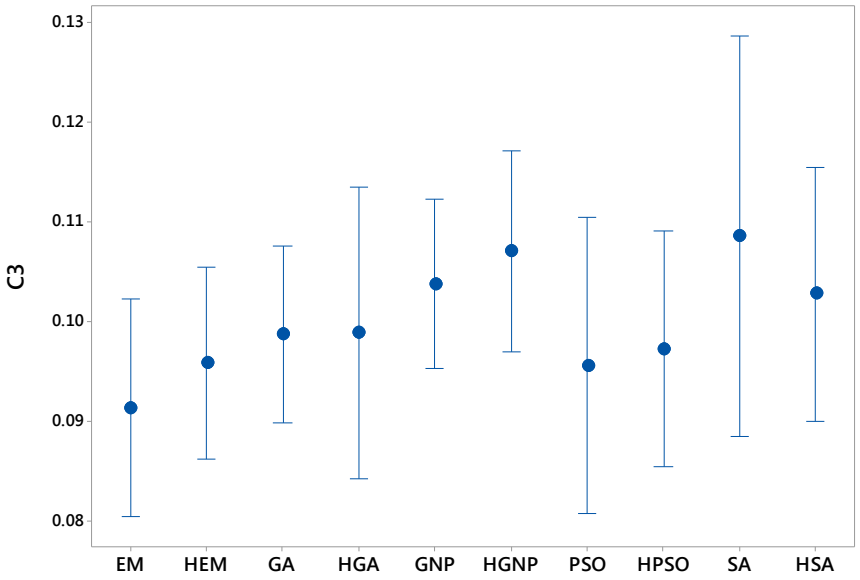

Fig. 13. Mean plot and LSD intervals at the 95 % confidence level for c_3 metric.

Table 19
Comparison of algorithms at c_4 metric.

Problems	Algorithms									
	EM	HEM	GA	HGA	GNP	HGNP	PSO	HPSO	SA	HSA
f1k10C4	0.0429	0.0438	0.1284	0.1303	**0.1462**	0.1295	0.0616	0.09	0.1109	0.1164
f1k20C4	**0.9978**	0.0002	0.0002	0.0003	0.0002	0.0003	0.0002	0.0002	0.0003	0.0003
f1k30C4	0.0753	0.0722	**0.1467**	0.1151	0.1087	0.1289	0.0675	0.0831	0.1038	0.0986
f2k10C4	0.0749	0.0799	0.104	0.1086	0.1135	**0.1264**	0.0866	0.092	0.1001	0.1141
f2k20C4	0.0693	0.0616	0.1058	0.1252	**0.1299**	0.123	0.0801	0.0811	0.0989	0.1252
f2k30C4	0.1062	0.0603	0.1029	0.1161	0.1386	**0.1463**	0.082	0.0398	0.0993	0.1085
f3k10C4	0.0711	0.0881	0.1087	0.0986	0.1249	**0.1632**	0.0463	0.1097	0.0741	0.1153
f3k20C4	0.1068	0.0704	0.1381	**0.1501**	0.1022	0.1151	0.087	0.0669	0.0589	0.1044
f3k30C4	0.0156	0.1401	0.146	0.1183	**0.1503**	0.1246	0.0962	0.0629	0.0554	0.0907
f4k10C4	0.0763	0.0631	0.1221	**0.1484**	0.1278	0.1153	0.0454	0.088	0.1113	0.1023
f4k20C4	0.0889	0.0707	0.1154	0.0874	**0.1427**	0.1172	0.0771	0.1093	0.08	0.1113
f4k30C4	0.1035	0.0793	0.114	0.0951	0.1263	**0.1264**	0.0959	0.0746	0.0925	0.0923
Average	**0.1524**	0.0691	0.111	0.1078	0.1176	0.118	0.0688	0.0748	0.0821	0.0983

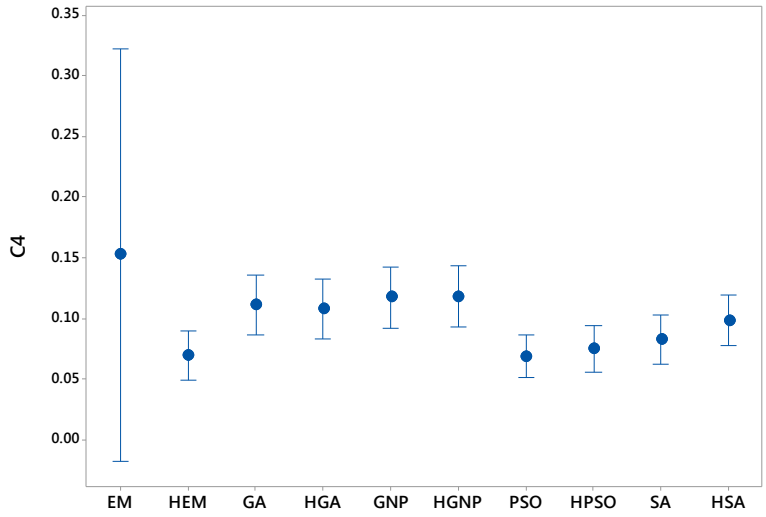

Fig. 14. Mean plot and LSD intervals at the 95 % confidence level for c_4 metric.

Comparison of all algorithms at mentioned normalized metrics respectively at MV, SV, MAD and VWS structure have been showed in Tables 20-23 for different *K* number (*K*=10, 20, 30). Also after each table, mean plot for all of algorithms at every metrics shown in Figs.16-19.

For example as shown in Table 20 (marked number at average row) and Fig.16, EM has better performance rather than other algorithms. Also in Table 20, according to the twelve solutions, EM, HEM, GA, GNP, PSO, SA and HSA respectively have 4, 1, 1, 1, 1, 3, 1 times good performance at mean–variance structure. Meanwhile in average mode, EM has a best performance that showed by mean plot in Fig.16. Also among of all hybrids, HEM and HSA have good performance.

Table 20
Comparison of algorithms at mean–variance structure.

Problems	Algorithms									
	EM	HEM	GA	HGA	GNP	HGNP	PSO	HPSO	SA	HSA
f1k10C1	0	0	0.001	0.0003	0.0045	0.0179	0.3777	0.0034	**0.5951**	0.0001
f1k10C2	**0.1129**	0.1101	0.0975	0.098	0.0973	0.0969	0.0972	0.0995	0.0938	0.0969
f1k10C3	**0.1192**	0.1129	0.0863	0.0851	0.0882	0.1027	0.1031	0.0965	0.1011	0.1048
f1k10C4	0.0429	0.0438	0.1284	0.1303	**0.1462**	0.1295	0.0616	0.09	0.1109	0.1164
f1k20C1	**0.7093**	0	0.0116	0.0233	0.25	0	0	0.0058	0	0
f1k20C2	0.0018	**0.1239**	0.1108	0.1095	0.1081	0.1065	0.1151	0.1191	0.0975	0.1077
f1k20C3	0.0796	0.1185	0.0955	0.0899	0.1014	0.1066	0.0803	0.0824	**0.1327**	0.113
f1k20C4	**0.9978**	0.0002	0.0002	0.0003	0.0002	0.0003	0.0002	0.0002	0.0003	0.0003
f1k30C1	0	0.0241	0.0002	0.0002	0.0389	0.0039	0	0	0	**0.9327**
f1k30C2	0.1029	0.1055	0.0948	0.0943	0.0953	0.0951	**0.1241**	0.0981	0.0977	0.0921
f1k30C3	0.103	0.1069	0.0926	0.0977	0.1059	0.1029	0.0797	0.087	**0.1167**	0.1077
f1k30C4	0.0753	0.0722	**0.1467**	0.1151	0.1087	0.1289	0.0675	0.0831	0.1038	0.0986
Average	**0.1954**	0.0682	0.0721	0.0703	0.0954	0.0743	0.0922	0.0638	0.1208	0.1475

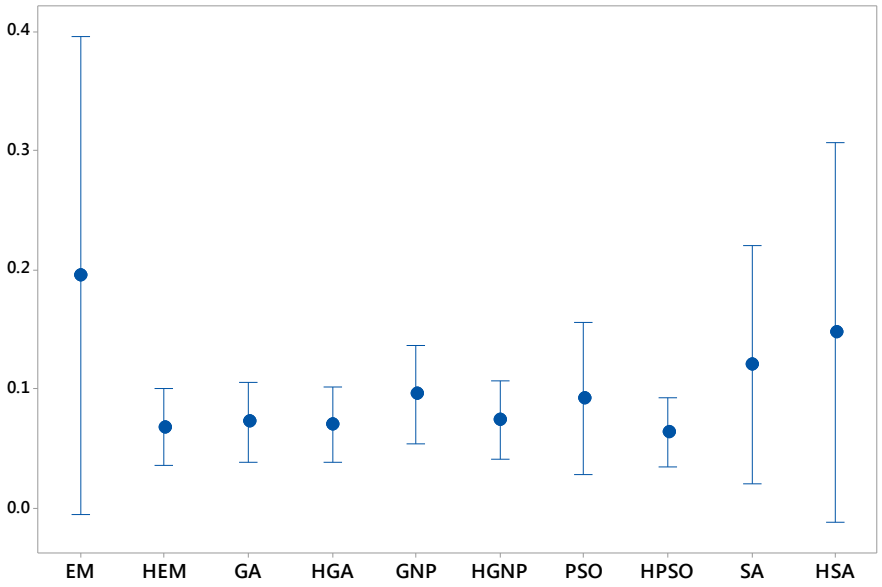

Fig. 15. Comparison of algorithms at mean–variance structure.

Table 21
Comparison of algorithms at SV structure.

Problems	Algorithms									
	EM	HEM	GA	HGA	GNP	HGNP	PSO	HPSO	SA	HSA
f2k10C1	0.0002	0	0.0187	0.0004	0.0056	0.0085	0	0	**0.9665**	0
f2k10C2	**0.1173**	0.1038	0.096	0.0975	0.0974	0.0952	0.0982	0.0931	0.0978	0.1038
f2k10C3	0.1132	0.101	0.0833	0.0831	0.1035	0.0916	0.0795	0.0887	**0.1302**	0.1259
f2k10C4	0.0749	0.0799	0.104	0.1086	0.1135	**0.1264**	0.0866	0.092	0.1001	0.1141
f2k20C1	0	0	0	0.0002	0.0064	0.1229	0	0	**0.8692**	0.0012
f2k20C2	0.1089	**0.1347**	0.0995	0.089	0.0891	0.092	0.0976	0.1077	0.0932	0.0883
f2k20C3	0.1003	0.1136	0.1085	0.0998	0.1014	0.1064	0.0703	0.0722	**0.1192**	0.1083
f2k20C4	0.0693	0.0616	0.1058	0.1252	**0.1299**	0.123	0.0801	0.0811	0.0989	0.1252
f2k30C1	0	0	0	0	0.113	0.0288	0	0	0	**0.8582**
f2k30C2	0.094	0.1289	0.0912	0.0916	0.0803	0.0856	0.09	**0.1474**	0.0979	0.0931
f2k30C3	0.0935	0.0871	0.0986	0.1025	0.0926	0.1065	0.0705	0.0844	**0.1376**	0.1266
f2k30C4	0.1062	0.0603	0.1029	0.1161	0.1386	**0.1463**	0.082	0.0398	0.0993	0.1085
Average	0.0732	0.0726	0.0757	0.0762	0.0893	0.0944	0.0629	0.0672	**0.2342**	0.1544

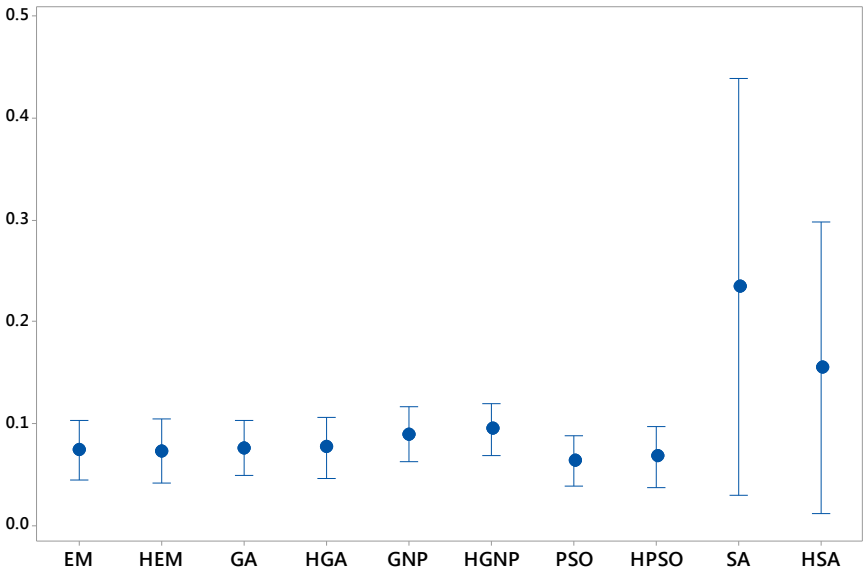

Fig. 16. Comparison of algorithms at SV structure.

Table 22
Comparison of algorithms at MAD structure.

Problems	Algorithms									
	EM	HEM	GA	HGA	GNP	HGNP	PSO	HPSO	SA	HSA
f3k10C1	**0.7409**	0	0.0763	0	0.009	0.1231	0	0.0485	0	0.0023
f3k10C2	**0.1267**	0.1248	0.1022	0.0795	0.092	0.0835	0.0864	0.0872	0.0988	0.119
f3k10C3	0.0877	0.0954	0.0968	0.0843	0.1041	0.0963	0.1113	0.1149	0.0887	**0.1206**
f3k10C4	0.0711	0.0881	0.1087	0.0986	0.1249	**0.1632**	0.0463	0.1097	0.0741	0.1153
f3k20C1	**0.4665**	0	0.1423	0.3841	0.0004	0	0	0	0.0067	0
f3k20C2	0.0997	0.1076	**0.1323**	0.0859	0.0994	0.0989	0.098	0.0989	0.0777	0.1017
f3k20C3	0.0657	0.0704	0.0872	0.0688	0.0818	0.0802	0.1412	0.1454	**0.1568**	0.1026
f3k20C4	0.1068	0.0704	0.1381	**0.1501**	0.1022	0.1151	0.087	0.0669	0.0589	0.1044
f3k30C1	0	0	0.1243	0	0.051	0.1297	0	0	**0.6951**	0
f3k30C2	0.0968	0.1178	0.0772	0.0787	0.1001	0.0755	0.0946	**0.1497**	0.0842	0.1254
f3k30C3	0.0802	0.0829	0.0852	0.0906	0.1051	0.1037	0.125	0.1007	**0.1265**	0.1001
f3k30C4	0.0156	0.1401	0.146	0.1183	**0.1503**	0.1246	0.0962	0.0629	0.0554	0.0907
Average	**0.1631**	0.0748	0.1097	0.1032	0.085	0.0995	0.0738	0.0821	0.1269	0.0818

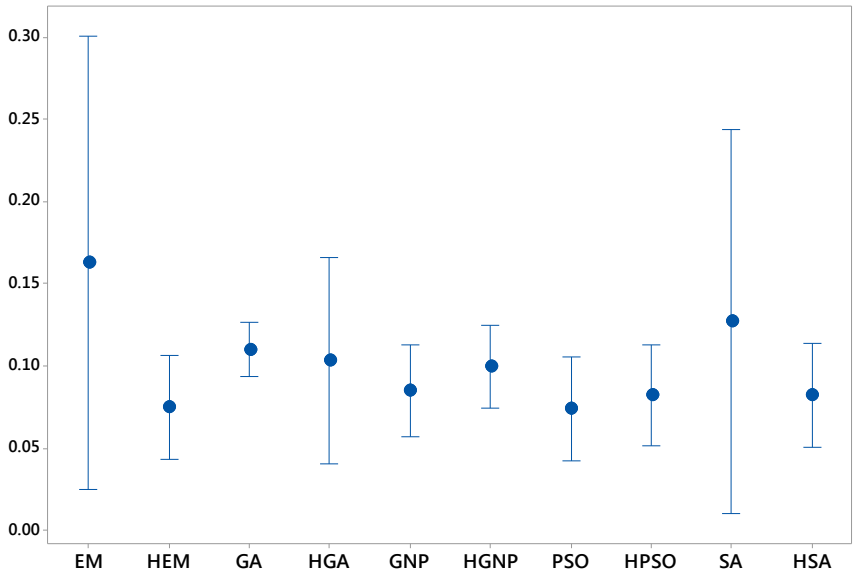

Fig. 17. Comparison of algorithms at MAD structure.

Table 23
Comparison of algorithms at VWS structure.

Problems	Algorithms									
	EM	HEM	GA	HGA	GNP	HGNP	PSO	HPSO	SA	HSA
f4k10C1	0.05	0	0.0917	**0.3417**	0.0833	0.2667	0	0	0.075	0.0917
f4k10C2	0.1025	0.0984	0.0977	0.1057	0.1086	**0.1168**	0.09	0.0919	0.0875	0.101
f4k10C3	0.1063	0.0938	0.1051	0.1004	0.1075	**0.121**	0.1095	0.0994	0.0667	0.0904
f4k10C4	0.0763	0.0631	0.1221	**0.1484**	0.1278	0.1153	0.0454	0.088	0.1113	0.1023
f4k20C1	0	0	0	0	0.2619	**0.7381**	0	0	0	0
f4k20C2	0.0907	0.1011	0.0947	**0.1173**	0.1104	0.11	0.0986	0.0993	0.0898	0.0881
f4k20C3	0.0705	0.089	0.1154	0.1306	0.1245	**0.139**	0.1038	0.0977	0.0692	0.0604
f4k20C4	0.0889	0.0707	0.1154	0.0874	**0.1427**	0.1172	0.0771	0.1093	0.08	0.1113
f4k30C1	0	0	0.0034	0	0	0.0387	0	0	**0.9512**	0.0067
f4k30C2	0.0914	**0.123**	0.099	0.1153	0.1077	0.0947	0.0881	0.0985	0.0913	0.0911
f4k30C3	0.0773	0.0788	0.1302	**0.1542**	0.1297	0.1276	0.0732	0.0983	0.0578	0.0729
f4k30C4	0.1035	0.0793	0.114	0.0951	0.1263	**0.1264**	0.0959	0.0746	0.0925	0.0923
Average	0.0715	0.0664	0.0907	0.1163	0.1192	**0.176**	0.0651	0.0714	0.1477	0.0757

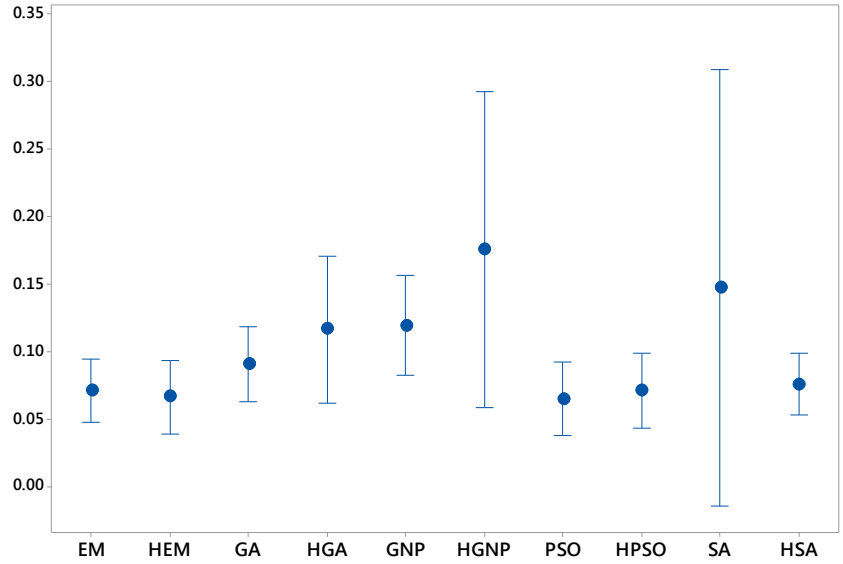

Fig. 18. Comparison of algorithms at VWS structure.

In this section, comparison of all algorithms at mentioned normalized metrics and cost functions by considering different k number ($k = 10, 20, 30$) have been showed in Tables 24-27. Also after each table, mean plot for all of algorithms at every metrics shown in Fig. 20-23.

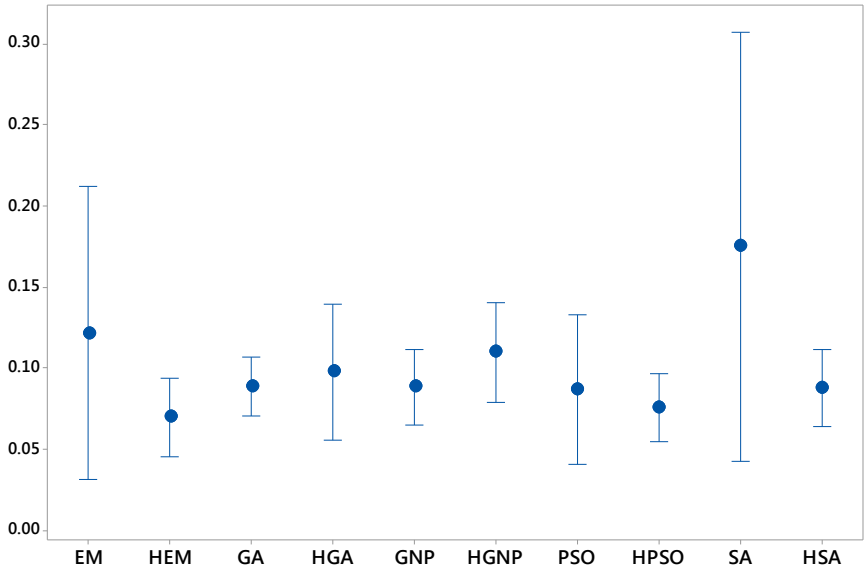

Fig. 19. Comparison of algorithms at *K*=10.

For example as shown in Table 24 (marked number at average row) and Fig. 20, SA has better performance rather than other algorithms. Also in Table 24, according to the sixteen solutions, EM, HGA, GNP, HGNP, SA and HSA respectively have 5, 2, 1, 4, 3, 1 times good performance at $k = 10$. Meanwhile in average mode, SA has a best performance that showed by mean plot in Fig. 20. Also among of all hybrids, HGA, HGNP and HSA have good performance.

Table 24
Comparison of algorithms at *K*=10.

Problems	Algorithms									
	EM	HEM	GA	HGA	GNP	HGNP	PSO	HPSO	SA	HSA
f1k10C1	0	0	0.001	0.0003	0.0045	0.0179	0.3777	0.0034	**0.5951**	0.0001
f1k10C2	**0.1129**	0.1101	0.0975	0.098	0.0973	0.0969	0.0972	0.0995	0.0938	0.0969
f1k10C3	**0.1192**	0.1129	0.0863	0.0851	0.0882	0.1027	0.1031	0.0965	0.1011	0.1048
f1k10C4	0.0429	0.0438	0.1284	0.1303	**0.1462**	0.1295	0.0616	0.09	0.1109	0.1164

f2k10C1	0.0002	0	0.0187	0.0004	0.0056	0.0085	0	0	**0.9665**	0
f2k10C2	**0.1173**	0.1038	0.096	0.0975	0.0974	0.0952	0.0982	0.0931	0.0978	0.1038
f2k10C3	0.1132	0.101	0.0833	0.0831	0.1035	0.0916	0.0795	0.0887	**0.1302**	0.1259
f2k10C4	0.0749	0.0799	0.104	0.1086	0.1135	**0.1264**	0.0866	0.092	0.1001	0.1141
f3k10C1	**0.7409**	0	0.0763	0	0.009	0.1231	0	0.0485	0	0.0023
f3k10C2	**0.1267**	0.1248	0.1022	0.0795	0.092	0.0835	0.0864	0.0872	0.0988	0.119
f3k10C3	0.0877	0.0954	0.0968	0.0843	0.1041	0.0963	0.1113	0.1149	0.0887	**0.1206**
f3k10C4	0.0711	0.0881	0.1087	0.0986	0.1249	**0.1632**	0.0463	0.1097	0.0741	0.1153
f4k10C1	0.05	0	0.0917	**0.3417**	0.0833	0.2667	0	0	0.075	0.0917
f4k10C2	0.1025	0.0984	0.0977	0.1057	0.1086	**0.1168**	0.09	0.0919	0.0875	0.101
f4k10C3	0.1063	0.0938	0.1051	0.1004	0.1075	**0.121**	0.1095	0.0994	0.0667	0.0904
f4k10C4	0.0763	0.0631	0.1221	**0.1484**	0.1278	0.1153	0.0454	0.088	0.1113	0.1023
Average	0.1389	0.0707	0.0919	0.104	0.0898	0.1173	0.0628	0.0761	**0.1581**	0.0905

Table 25
Comparison of algorithms at K=20.

Problems	Algorithms									
	EM	HEM	GA	HGA	GNP	HGNP	PSO	HPSO	SA	HSA
f1k20C1	**0.7093**	0	0.0116	0.0233	0.25	0	0	0.0058	0	0
f1k20C2	0.0018	**0.1239**	0.1108	0.1095	0.1081	0.1065	0.1151	0.1191	0.0975	0.1077
f1k20C3	0.0796	0.1185	0.0955	0.0899	0.1014	0.1066	0.0803	0.0824	**0.1327**	0.113
f1k20C4	**0.9978**	0.0002	0.0002	0.0003	0.0002	0.0003	0.0002	0.0002	0.0003	0.0003
f2k20C1	0	0	0	0.0002	0.0064	0.1229	0	0	**0.8692**	0.0012
f2k20C2	0.1089	**0.1347**	0.0995	0.089	0.0891	0.092	0.0976	0.1077	0.0932	0.0883
f2k20C3	0.1003	0.1136	0.1085	0.0998	0.1014	0.1064	0.0703	0.0722	**0.1192**	0.1083
f2k20C4	0.0693	0.0616	0.1058	0.1252	**0.1299**	0.123	0.0801	0.0811	0.0989	0.1252
f3k20C1	**0.4665**	0	0.1423	0.3841	0.0004	0	0	0	0.0067	0
f3k20C2	0.0997	0.1076	**0.1323**	0.0859	0.0994	0.0989	0.098	0.0989	0.0777	0.1017
f3k20C3	0.0657	0.0704	0.0872	0.0688	0.0818	0.0802	0.1412	0.1454	**0.1568**	0.1026
f3k20C4	0.1068	0.0704	0.1381	**0.1501**	0.1022	0.1151	0.087	0.0669	0.0589	0.1044
f4k20C1	0	0	0	0	0.2619	**0.7381**	0	0	0	0
f4k20C2	0.0907	0.1011	0.0947	**0.1173**	0.1104	0.11	0.0986	0.0993	0.0898	0.0881
f4k20C3	0.0705	0.089	0.1154	0.1306	0.1245	**0.139**	0.1038	0.0977	0.0692	0.0604
f4k20C4	0.0889	0.0707	0.1154	0.0874	**0.1427**	0.1172	0.0771	0.1093	0.08	0.1113
Average	0.1056	0.0683	0.0949	0.1115	0.1042	**0.1536**	0.0711	0.0732	0.1433	0.0743

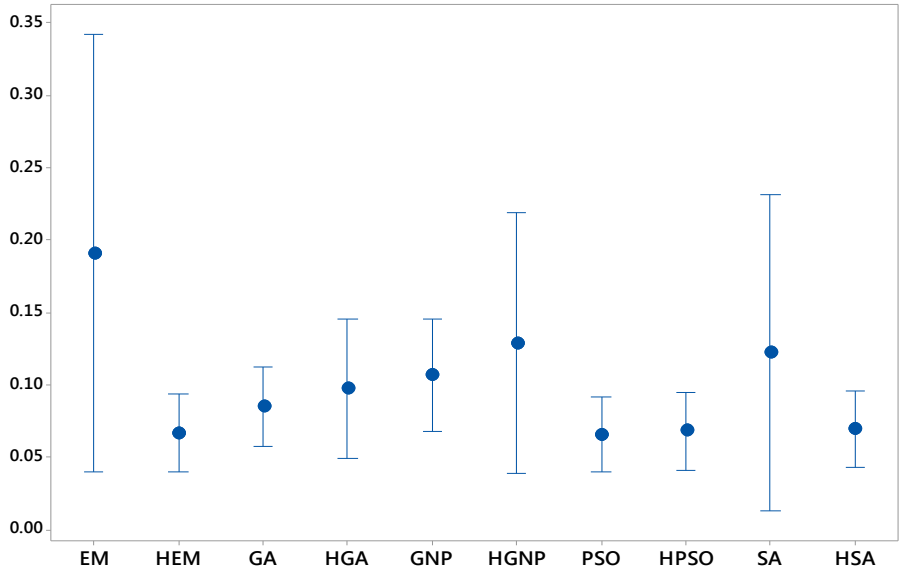

Fig. 20. Comparison of algorithms at K=20.

Table 26
Comparison of algorithms at K=30.

Problems	Algorithms									
	EM	HEM	GA	HGA	GNP	HGNP	PSO	HPSO	SA	HSA
f1k30C1	0	0.0241	0.0002	0.0002	0.0389	0.0039	0	0	0	0.9327
f1k30C2	0.1029	0.1055	0.0948	0.0943	0.0953	0.0951	**0.1241**	0.0981	0.0977	0.0921
f1k30C3	0.103	0.1069	0.0926	0.0977	0.1059	0.1029	0.0797	0.087	**0.1167**	0.1077
f1k30C4	0.0753	0.0722	**0.1467**	0.1151	0.1087	0.1289	0.0675	0.0831	0.1038	0.0986
f2k30C1	0	0	0	0	0.113	0.0288	0	0	0	**0.8582**
f2k30C2	0.094	0.1289	0.0912	0.0916	0.0803	0.0856	0.09	**0.1474**	0.0979	0.0931
f2k30C3	0.0935	0.0871	0.0986	0.1025	0.0926	0.1065	0.0705	0.0844	**0.1376**	0.1266
f2k30C4	0.1062	0.0603	0.1029	0.1161	0.1386	**0.1463**	0.082	0.0398	0.0993	0.1085
f3k30C1	0	0	0.1243	0	0.051	0.1297	0	0	**0.6951**	0
f3k30C2	0.0968	0.1178	0.0772	0.0787	0.1001	0.0755	0.0946	**0.1497**	0.0842	0.1254
f3k30C3	0.0802	0.0829	0.0852	0.0906	0.1051	0.1037	0.125	0.1007	**0.1265**	0.1001
f3k30C4	0.0156	0.1401	0.146	0.1183	**0.1503**	0.1246	0.0962	0.0629	0.0554	0.0907
f4k30C1	0	0	0.0034	0	0	0.0387	0	0	**0.9512**	0.0067
f4k30C2	0.0914	**0.123**	0.099	0.1153	0.1077	0.0947	0.0881	0.0985	0.0913	0.0911
f4k30C3	0.0773	0.0788	0.1302	**0.1542**	0.1297	0.1276	0.0732	0.0983	0.0578	0.0729
f4k30C4	0.1035	0.0793	0.114	0.0951	0.1263	**0.1264**	0.0959	0.0746	0.0925	0.0923
Average	0.0632	0.0749	0.0893	0.0802	0.0996	0.099	0.068	0.0714	**0.2074**	0.1471

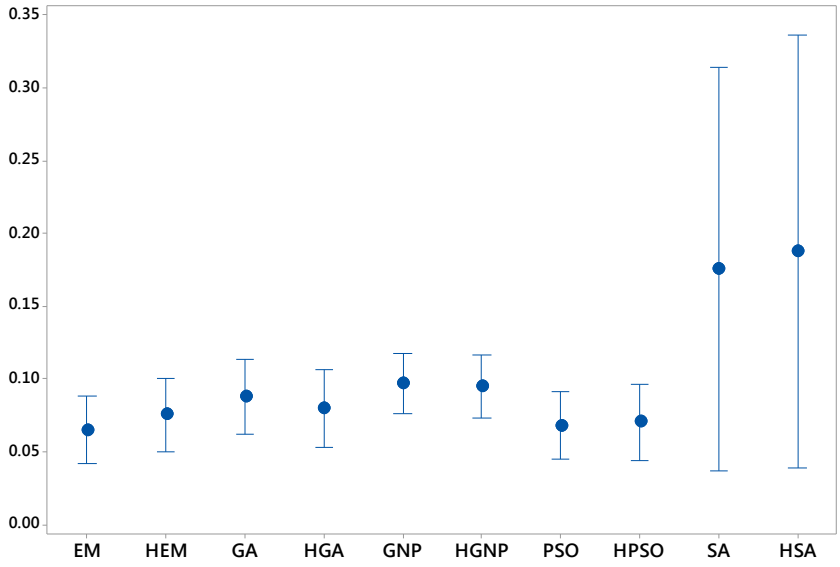

Fig. 21. Comparison of algorithms at K=30.

Table 27 shows final comparison between all algorithms by considering the result of the experiments for all models at 48 normalized metrics per average. As shown in this table and Fig. 19, the best performance is obtained by SA with the average of 0.1574.

Table 27
Final comparison of all algorithms.

EM	HEM	GA	HGA	GNP	HGNP	PSO	HPSO	SA	HSA
0.1258	0.0705	0.0871	0.0915	0.0972	0.111	0.0735	0.0711	**0.1574**	0.1149

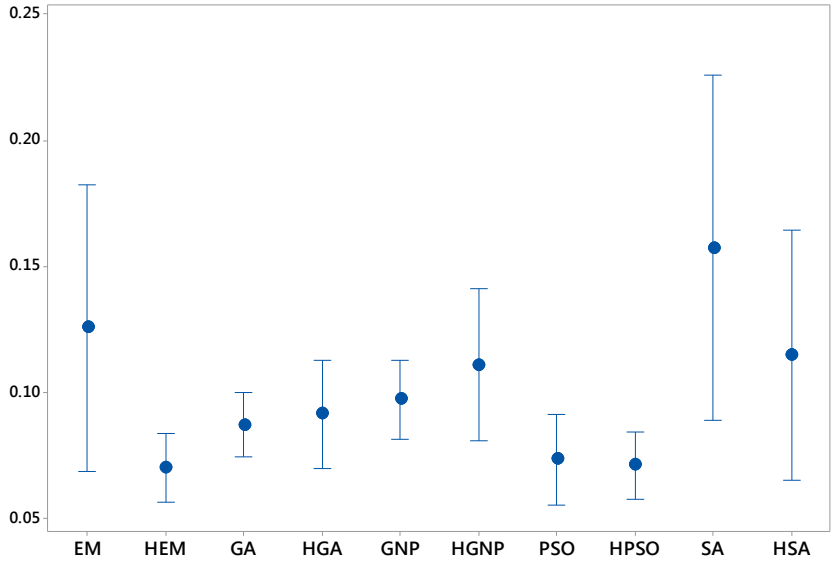

Fig. 22. Final comparison of all algorithms.

5. Conclusions and Future Research Directions

This paper was focused to solve the portfolio optimization problem and tracing out its efficient frontier. The MV, MAD, SV and VWS based cardinality constrained portfolio optimization models that includes cardinality and bounding constraints were used to develop the meta-heuristic algorithms. In order to solve the problems, the EM, GA, GNP, PSO and SA algorithms are proposed and hybridized with a diversification mechanism. Trial-and-error method was offered to set the proper values for the proposed algorithm's parameters. The performance of approaches in the quality of Pareto optimal solutions were evaluated by four measurement metrics. Results showed that SA was better than other algorithms in QM and SM metrics. Comparing the obtained results in terms of MID and DM metrics reveals that the proposed HEM and EM are more effective respectively.

In the future, novel hybrid meta-heuristic algorithms with new definitions of risk or return by different metrics can be useful to solving financial problems, such as Risk Management (RM) and option pricing etc.

References

Ackora-Prah, J., Gyamerah, S. A., Andam, P. S., & Gyamfi, D. (2014). Pattern Search for Portfolio Selection. *Applied Mathematical Sciences, 8*(143), 7137-7147.

Aguilar-Rivera, R., Valenzuela-Rendón, M., & Rodríguez-Ortiz, J. (2015). Genetic algorithms and Darwinian approaches in financial applications: A survey. *Expert Systems with Applications, 42*(21), 7684-7697.

Ahmad, G., Hasan, F., Shahid, M., Imran, M., & Alam, M. (2023). *A Cardinality-Constrained Portfolio Selection Model Using Golden Eagle Optimizer in Stock Markets.* Paper presented at the 2023 4th International Conference on Data Analytics for Business and Industry (ICDABI).

Ahmadi, A., & Davari-Ardakani, H. (2017). A multistage stochastic programming framework for cardinality constrained portfolio optimization. *Numerical Algebra, Control & Optimization, 7*(3), 359-377.

Anagnostopoulos, K., & Mamanis, G. (2011). The mean–variance cardinality constrained portfolio optimization problem: An experimental evaluation of five multiobjective evolutionary algorithms. *Expert Systems with Applications, 38*(11), 14208-14217.

Anagnostopoulos, K. P., & Mamanis, G. (2011a). The mean–variance cardinality constrained portfolio optimization problem: An experimental evaluation of five multiobjective evolutionary algorithms. *Expert Systems with Applications, 38*(11), 14208-14217.

Anagnostopoulos, K. P., & Mamanis, G. (2011b). Multiobjective evolutionary algorithms for complex portfolio optimization problems. *Computational Management Science, 8*(3), 259-279.

Atiya, A. F. (2001). Bankruptcy prediction for credit risk using neural networks: A survey and new results. *IEEE Transactions on neural networks, 12*(4), 929-935.

Bacanin, N., & Tuba, M. (2014). Firefly algorithm for cardinality constrained mean-variance portfolio optimization problem with entropy diversity constraint. *The Scientific World Journal, 2014*.

Bastiani, S. S., Cruz-Reyes, L., Fernandez, E., Gómez, C., & Rivera, G. (2015). An ant colony algorithm for solving the selection portfolio problem, using a quality-assessment model for portfolios of projects expressed by a priority ranking *Design of Intelligent Systems Based on Fuzzy Logic, Neural Networks and Nature-Inspired Optimization* (pp. 357-373): Springer.

Baykasoğlu, A., Yunusoglu, M. G., & Özsoydan, F. B. (2015). A GRASP based solution approach to solve cardinality constrained portfolio optimization problems. *Computers & Industrial Engineering, 90*, 339-351.

Birbil, Ş. İ., & Fang, S.-C. (2003). An electromagnetism-like mechanism for global optimization. *Journal of Global Optimization, 25*(3), 263-282.

Branda, M., Bucher, M., Červinka, M., & Schwartz, A. (2018). Convergence of a Scholtes-type regularization method for cardinality-constrained optimization problems with an application in sparse robust portfolio optimization. *Computational Optimization and Applications, 70*(2), 503-530.

Branke, J., Scheckenbach, B., Stein, M., Deb, K., & Schmeck, H. (2009). Portfolio optimization with an envelope-based multi-objective evolutionary algorithm. *European Journal of Operational Research, 199*(3), 684-693.

Busetti, F. (2006). Heuristic approaches to realistic portfolio optimisation. *WIT Transactions on Modelling and Simulation, 43*.

Canela, M. A., & Collazo, E. P. (2007). Portfolio selection with skewness in emerging market industries. *Emerging Markets Review, 8*(3), 230-250.

Černý, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of optimization theory and applications, 45*(1), 41-51.

Cesarone, F., Scozzari, A., & Tardella, F. (2013). A new method for mean-variance portfolio optimization with cardinality constraints. *Annals of Operations Research, 205*(1), 213-234.

Chang, T.-J., Meade, N., Beasley, J. E., & Sharaiha, Y. M. (2000). Heuristics for cardinality constrained portfolio optimisation. *Computers & Operations Research, 27*(13), 1271-1302.

Chang, T.-J., Yang, S.-C., & Chang, K.-J. (2009). Portfolio optimization problems in different risk measures using genetic algorithm. *Expert Systems with Applications, 36*(7), 10529-10537.

Chen, A. H., Liang, Y.-C., & Liu, C.-C. (2012). *An artificial bee colony algorithm for the cardinality-constrained portfolio optimization problems.* Paper presented at the Evolutionary Computation (CEC), 2012 IEEE Congress on.

Chen, A. H., Liang, Y.-C., & Liu, C.-C. (2013). *Portfolio optimization using improved artificial bee colony approach.* Paper presented at the Computational Intelligence for Financial Engineering & Economics (CIFEr), 2013 IEEE Conference on.

Chen, B., Lin, Y., Zeng, W., Xu, H., & Zhang, D. (2017). The mean-variance cardinality constrained portfolio optimization problem using a local search-based multi-objective evolutionary algorithm. *Applied Intelligence, 47*(2), 505-525.

Chen, J.-S., Hou, J.-L., Wu, S.-M., & Chang-Chien, Y.-W. (2009). Constructing investment strategy portfolios by combination genetic algorithms. *Expert Systems with Applications, 36*(2), 3824-3828.

Chen, W. (2015). Artificial bee colony algorithm for constrained possibilistic portfolio optimization problem. *Physica A: Statistical Mechanics and its Applications, 429*, 125-139.

Chen, W., Li, D., & Liu, Y.-J. (2018). A novel hybrid ICA-FA algorithm for multi-period uncertain portfolio optimization model based on multiple criteria. *IEEE Transactions on Fuzzy Systems*.

Chen, W., Li, D., Lu, S., & Liu, W. (2018). Multi-period mean–semivariance portfolio optimization based on uncertain measure. *Soft Computing*, 1-17.

Chen, W., Wang, Y., Gupta, P., & Mehlawat, M. K. (2018). A novel hybrid heuristic algorithm for a new uncertain mean-variance-skewness portfolio selection model with real constraints. *Applied Intelligence*, 1-23.

Chen, W., & Xu, W. A Hybrid Multiobjective Bat Algorithm for Fuzzy Portfolio Optimization with Real-World Constraints. *International Journal of Fuzzy Systems*, 1-17.

Chen, Y., Mabu, S., & Hirasawa, K. (2010). A model of portfolio optimization using time adapting genetic network programming. *Computers & Operations Research, 37*(10), 1697-1707.

Chen, Y., Mabu, S., Hirasawa, K., & Hu, J. (2007). *Trading rules on stock markets using genetic network programming with sarsa learning.* Paper presented at the Proceedings of the 9th annual conference on Genetic and evolutionary computation.

Chiam, S., Tan, K., & Al Mamum, A. (2008). Evolutionary multi-objective portfolio optimization in practical context. *International Journal of Automation and Computing, 5*(1), 67-80.

Chiam, S. C., Al Mamun, A., & Low, Y. (2007). *A realistic approach to evolutionary multiobjective portfolio optimization.* Paper presented at the Evolutionary Computation, 2007. CEC 2007. IEEE Congress on.

Cui, T., Cheng, S., & Bai, R. (2014). *A combinatorial algorithm for the cardinality constrained portfolio optimization problem.* Paper presented at the Evolutionary Computation (CEC), 2014 IEEE Congress on.

Cui, X., Zheng, X., Zhu, S., & Sun, X. (2013). Convex relaxations and MIQCQP reformulations for a class of cardinality-constrained portfolio selection problems. *Journal of Global Optimization, 56*(4), 1409-1423.

Cura, T. (2009). Particle swarm optimization approach to portfolio optimization. *Nonlinear analysis: Real world applications, 10*(4), 2396-2406.

Deng, G.-F., & Lin, W.-T. (2010). *Ant colony optimization for Markowitz mean-variance portfolio model.* Paper presented at the International Conference on Swarm, Evolutionary, and Memetic Computing.

Deng, G.-F., Lin, W.-T., & Lo, C.-C. (2012). Markowitz-based portfolio selection with cardinality constraints using improved particle swarm optimization. *Expert Systems with Applications, 39*(4), 4558-4566.

Dropsy, V. (2011). Do macroeconomic factors help in predicting international equity risk premia?: Testing the out-of-sample accuracy of linear and nonlinear

forecasts. *Journal of Applied Business Research (JABR), 12*(3), 120-132.

Eguchi, T., Hirasawa, K., Hu, J., & Ota, N. (2006). A study of evolutionary multiagent models based on symbiosis. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 36*(1), 179-193.

Ertenlice, O., & Kalayci, C. B. (2018). A survey of swarm intelligence for portfolio optimization: Algorithms and applications. *Swarm and evolutionary computation, 39*, 36-52.

Esfahanipour, A., & Mousavi, S. (2011). A genetic programming model to generate risk-adjusted technical trading rules in stock markets. *Expert Systems with Applications, 38*(7), 8438-8445.

Feng, Y., Zhang, B., & Peng, J. (2023). Mean-risk model for uncertain portfolio selection with background risk and realistic constraints. *Journal of Industrial & Management Optimization, 19*(7).

Fernández, A., & Gómez, S. (2007). Portfolio selection using neural networks. *Computers & Operations Research, 34*(4), 1177-1191.

Fieldsend, J. E., Matatko, J., & Peng, M. (2004). *Cardinality constrained portfolio optimisation.* Paper presented at the International Conference on Intelligent Data Engineering and Automated Learning.

Fogarasi, N., & Levendovszky, J. (2013). Sparse, mean reverting portfolio selection using simulated annealing. *Algorithmic Finance, 2*(3-4), 197-211.

Frajtova-Michalikova, K., Spuchľakova, E., & Misankova, M. (2015). Portfolio Optimization. *Procedia Economics and Finance, 26*, 1102-1107.

Golberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning. *Addion wesley, 1989*, 102.

Golmakani, H. R., & Alishah, E. J. (2008). *Portfolio selection using an artificial immune system.* Paper presented at the Information Reuse and Integration, 2008. IRI 2008. IEEE International Conference on.

Golmakani, H. R., & Fazel, M. (2011). Constrained portfolio selection using particle swarm optimization. *Expert Systems with Applications, 38*(7), 8327-8335.

Guijarro, F. (2018). A similarity measure for the cardinality constrained frontier in the mean–variance optimization model. *Journal of the Operational Research Society, 69*(6), 928-945.

Gunjan, A., & Bhattacharyya, S. (2023). A brief review of portfolio optimization techniques. *Artificial Intelligence Review, 56*(5), 3847-3886. doi:10.1007/s10462-022-10273-7

Hajnoori, A., Amiri, M., & Alimi, A. (2013). Forecasting stock price using grey-fuzzy technique and portfolio optimization by invasive weed optimization algorithm. *Decision Science Letters, 2*(3), 175-184.

Hardoroudi, N. D., Keshvari, A., Kallio, M., & Korhonen, P. (2017). Solving cardinality constrained mean-variance portfolio problems via MILP. *Annals of Operations Research, 254*(1-2), 47-59.

Hirasawa, K., Eguchi, T., Zhou, J., Yu, L., Hu, J., & Markon, S. (2008). A double-deck elevator group supervisory control system using genetic network programming. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 38*(4), 535-550.

Hirasawa, K., Okubo, M., Katagiri, H., Hu, J., & Murata, J. (2001). *Comparison between genetic network programming (GNP) and genetic programming (GP).* Paper presented at the Evolutionary Computation, 2001. Proceedings of the 2001 Congress on.

Holland, J. H. (1975). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*: U Michigan Press.

Huang, H.-H., & Wang, C.-P. (2013). Portfolio selection and portfolio frontier with background risk. *The North American Journal of Economics and Finance, 26*, 177-196.

Huang, X. (2008). Portfolio selection with a new definition of risk. *European Journal of Operational Research, 186*(1), 351-357.

Jaaman, S. H. H., Lam, W. H., & Isa, Z. (2011). Different Downside Risk Approaches in Portfolio Optimisation. *Journal of Quality Measurement and Analysis JQMA, 7*(1), 77-84.

Jalota, H., & Thakur, M. (2018). Genetic algorithm designed for solving portfolio optimization problems subjected to cardinality constraint. *International Journal of System Assurance Engineering and Management, 9*(1), 294-305.

Jawad, M., Naz, M., & Muqaddus, H. (2024). A multi-criteria decision-making approach for portfolio selection by using an automatic spherical fuzzy AHP algorithm. *Journal of the Operational Research Society, 75*(1), 85-98.

Jia, J., & Dyer, J. S. (1996). A standard measure of risk and risk-value models. *Management Science, 42*(12), 1691-1705.

Jiang, C., Ma, Y., & An, Y. (2016). Portfolio selection with a systematic skewness constraint. *The North American Journal of Economics and Finance, 37*, 393-405.

Jimbo, H. C., Ngongo, I. S., Andjiga, N. G., Suzuki, T., & Onana, C. A. (2017). Portfolio Optimization under Cardinality Constraints: A Comparative Study. *Open Journal of Statistics, 7*(04), 731.

Jin, Y., Qu, R., & Atkin, J. Constrained portfolio optimisation: the state-of-the-art Markowitz models. In: The 2016 International Conference on Operations Research and Enterprise Systems, 23-25 February 2016, Rome, Italy.

Kalayci, C. B., Ertenlice, O., Akyer, H., & Aygoren, H. (2017). An artificial bee colony algorithm with feasibility enforcement and infeasibility toleration procedures for cardinality constrained portfolio optimization. *Expert Systems with Applications, 85*, 61-75.

Kalayci, C. B., Ertenlice, O., Akyer, H., & Aygoren, H. (2017). A review on the current applications of genetic algorithms in mean-variance portfolio

optimization Ortalama-varyans portföy optimizasyonunda genetik algoritma uygulamaları üzerine bir literatür araştırması.

KAMILI, H., & RIFFI, M. E. (2015). PORTFOLIO SELECTION USING THE CAT SWARM OPTIMIZATION. *Journal of Theoretical & Applied Information Technology*, *74*(3).

Kandakoglu, M., Walther, G., & Ben Amor, S. (2024). The use of multi-criteria decision-making methods in project portfolio selection: a literature review and future research directions. *Annals of Operations Research*, *332*(1), 807-830.

Kao, Y., & Cheng, H.-T. (2013). Bacterial foraging optimization approach to portfolio optimization. *Computational Economics*, *42*(4), 453-470.

Katagiri, H., Hirasawa, K., Hu, J., Murata, J., & Kosaka, M. (2002). Network structure oriented evolutionary model: Genetic network programming. *Transactions of the Society of Instrument and Control Engineers*, *38*(5), 485-494.

Kennedy, J., & Eberhart, R. (1942). Particle Swarm Optimization: IEEE International Conference on Neural Networks, 1995. *Perth: IEEE, 19951*.

Kessaci, Y. (2017). *A multi-objective continuous genetic algorithm for financial portfolio optimization problem.* Paper presented at the Proceedings of the Genetic and Evolutionary Computation Conference Companion.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*(4598), 671-680.

Konno, H., & Yamamoto, R. (2005). A mean-variance-skewness model: algorithm and applications. *International Journal of Theoretical and Applied Finance*, *8*(04), 409-423.

Konno, H., & Yamazaki, H. (1991). Mean-absolute deviation portfolio optimization model and its applications to Tokyo stock market. *Management science*, *37*(5), 519-531.

Koza, J. R. (1992). *Genetic programming: on the programming of computers by means of natural selection* (Vol. 1): MIT press.

Lam, M. (2004). Neural network techniques for financial performance prediction: integrating fundamental and technical analysis. *Decision support systems*, *37*(4), 567-581.

Le Thi, H. A., Moeini, M., & Dinh, T. P. (2009). Portfolio selection under downside risk measures and cardinality constraints based on DC programming and DCA. *Computational Management Science*, *6*(4), 459-475.

Liagkouras, K., & Metaxiotis, K. (2014). A new probe guided mutation operator and its application for solving the cardinality constrained portfolio optimization problem. *Expert Systems with Applications*, *41*(14), 6274-6290.

Liagkouras, K., & Metaxiotis, K. (2016). A new efficiently encoded multiobjective algorithm for the solution of the cardinality constrained portfolio optimization problem. *Annals of Operations Research*, 1-39.

Liagkouras, K., & Metaxiotis, K. (2018). Handling the complexities of the multi-constrained portfolio optimization problem with the support of a novel MOEA. *Journal of the Operational Research Society*, 1-19.

Lim, K. (2024). Covariance Matrix Analysis for Optimal Portfolio Selection. *Available at SSRN 4874008*.

Lin, C.-C., & Liu, Y.-T. (2008). Genetic algorithms for portfolio selection problems with minimum transaction lots. *European Journal of Operational Research*, *185*(1), 393-404.

Lin, D., Wang, S., & Yan, H. (2001). A multiobjective genetic algorithm for portfolio selection problem.

Lin, S.-Y., Horng, S.-J., Kao, T.-W., Huang, D.-K., Fahn, C.-S., Lai, J.-L., . . . Kuo, I.-H. (2010). An efficient bi-objective personnel assignment algorithm based on a hybrid particle swarm optimization model. *Expert Systems with Applications*, *37*(12), 7825-7830.

Liu, S.-T. (2011). The mean-absolute deviation portfolio selection problem with interval-valued returns. *Journal of computational and applied mathematics*, *235*(14), 4149-4157.

Lwin, K., & Qu, R. (2013). A hybrid algorithm for constrained portfolio selection problems. *Applied Intelligence*, *39*(2), 251-266.

Lwin, K., Qu, R., & Kendall, G. (2014). A learning-guided multi-objective evolutionary algorithm for constrained portfolio optimization. *Applied Soft Computing*, *24*, 757-772.

Ma, X., Gao, Y., & Wang, B. (2012). Portfolio optimization with cardinality constraints based on hybrid differential evolution. *AASRI Procedia*, *1*, 311-317.

Mabu, S., Hirasawa, K., & Hu, J. (2004). *Genetic network programming with reinforcement learning and its performance evaluation.* Paper presented at the Genetic and Evolutionary Computation Conference.

Mabu, S., Hirasawa, K., & Hu, J. (2007). A graph-based evolutionary algorithm: genetic network programming (GNP) and its extension using reinforcement learning. *Evolutionary Computation*, *15*(3), 369-398.

Mabu, S., Hirasawa, K., Hu, J., & Murata, J. (2002). *Online learning of genetic network programming (gnp).* Paper presented at the CEC.

Macedo, L. L., Godinho, P., & Alves, M. J. (2017). Mean-semivariance portfolio optimization with multiobjective evolutionary algorithms and technical analysis rules. *Expert Systems with Applications*, *79*, 33-43.

Maringer, D., & Kellerer, H. (2003). Optimization of cardinality constrained portfolios with a hybrid local search algorithm. *Or Spectrum*, *25*(4), 481-495.

Markowitz, H. (1952). Portfolio selection. *The journal of finance*, *7*(1), 77-91.

Mishra, S. K., Panda, G., & Majhi, B. (2016). Prediction based mean-variance model for constrained portfolio

assets selection using multiobjective evolutionary algorithms. *Swarm and evolutionary computation, 28*, 117-130.

Mishra, S. K., Panda, G., & Majhi, R. (2014). Constrained portfolio asset selection using multiobjective bacteria foraging optimization. *Operational Research, 14*(1), 113-145.

Monge, J. F. (2017). Cardinality constrained portfolio selection via factor models. *arXiv preprint arXiv:1708.02424*.

Moral-Escudero, R., Ruiz-Torrubiano, R., & Suárez, A. (2006). *Selection of optimal investment portfolios with cardinality constraints.* Paper presented at the Evolutionary Computation, 2006. CEC 2006. IEEE Congress on.

Mozafari, M., Jolai, F., & Tafazzoli, S. (2011). A new IPSO-SA approach for cardinality constrained portfolio optimization. *International Journal of Industrial Engineering Computations, 2*(2), 249-262.

Murray, W., & Shek, H. (2012). A local relaxation method for the cardinality constrained portfolio optimization problem. *Computational Optimization and Applications, 53*(3), 681-709.

Mutunge, P., & Haugland, D. (2018). Minimizing the tracking error of cardinality constrained portfolios. *Computers & Operations Research, 90*, 33-41.

Ni, Q., Yin, X., Tian, K., & Zhai, Y. (2017). Particle swarm optimization with dynamic random population topology strategies for a generalized portfolio selection problem. *Natural Computing, 16*(1), 31-44.

Oh, K. J., Kim, T. Y., Min, S.-H., & Lee, H. Y. (2006). Portfolio algorithm based on portfolio beta using genetic algorithm. *Expert Systems with Applications, 30*(3), 527-534.

Ruiz-Torrubiano, R., & Suárez, A. (2010). Hybrid approaches and dimensionality reduction for portfolio selection with cardinality constraints. *IEEE Computational Intelligence Magazine, 5*(2), 92-107.

Sabar, N. R., Turky, A., Leenders, M., & Song, A. (2018). *Multi-population Genetic Algorithm for Cardinality Constrained Portfolio Selection Problems.* Paper presented at the International Conference on Computational Science.

Saborido, R., Ruiz, A. B., Bermúdez, J. D., Vercher, E., & Luque, M. (2016). Evolutionary multi-objective optimization algorithms for fuzzy portfolio selection. *Applied Soft Computing, 39*, 48-63.

Sadigh, A. N., Mokhtari, H., Iranpoor, M., & Ghomi, S. (2012). Cardinality constrained portfolio optimization using a hybrid approach based on particle swarm optimization and Hopfield neural network. *Advanced Science Letters, 17*(1), 11-20.

Sadjadi, S. J., Gharakhani, M., & Safari, E. (2012). Robust optimization framework for cardinality constrained portfolio problem. *Applied Soft Computing, 12*(1), 91-99.

Samuelson, P. (1958). The fundamental approximation theorem of portfolio analysis in terms of means variances and higher moments. *Review of Economic Studies, 25*, 65-86.

Shaw, D. X., Liu, S., & Kopman, L. (2008). Lagrangian relaxation procedure for cardinality-constrained portfolio optimization. *Optimisation Methods & Software, 23*(3), 411-420.

Skolpadungket, P., Dahal, K., & Harnpornchai, N. (2007). *Portfolio optimization using multi-obj ective genetic algorithms.* Paper presented at the Evolutionary Computation, 2007. CEC 2007. IEEE Congress on.

Soleimani, H., Golmakani, H. R., & Salimi, M. H. (2009). Markowitz-based portfolio selection with minimum transaction lots, cardinality constraints and regarding sector capitalization using genetic algorithm. *Expert Systems with Applications, 36*(3), 5058-5063.

Streichert, F., & Tanaka-Yamawaki, M. (2006). *The effect of local search on the constrained portfolio selection problem.* Paper presented at the Evolutionary Computation, 2006. CEC 2006. IEEE Congress on.

Suthiwong, D., & Sodanil, M. (2016). *Cardinality-constrained portfolio optimization using an improved quick artificial bee colony algorithm.* Paper presented at the Computer Science and Engineering Conference (ICSEC), 2016 International.

Thakur, M., Meghwani, S. S., & Jalota, H. (2014). A modified real coded genetic algorithm for constrained optimization. *Applied Mathematics and Computation, 235*, 292-317.

Thomaidis, N. S. (2010). *Active portfolio management from a fuzzy multi-objective programming perspective.* Paper presented at the European Conference on the Applications of Evolutionary Computation.

Tian, F., Li, D. J., Fu, Q., Zhu, Z. F., Fu, Y. C., Wang, X. K., & Sun, C. Q. (2006). Construction of introgression lines carrying wild rice (Oryza rufipogon Griff.) segments in cultivated rice (Oryza sativa L.) background and characterization of introgressed segments associated with yield-related traits. *Theoretical and Applied Genetics, 112*(3), 570-580.

Tuba, M., & Bacanin, N. (2014). Artificial bee colony algorithm hybridized with firefly algorithm for cardinality constrained mean-variance portfolio selection problem. *Applied Mathematics & Information Sciences, 8*(6), 2831.

Vercher, E., & Bermúdez, J. D. (2015). Portfolio optimization using a credibility mean-absolute semi-deviation model. *Expert Systems with Applications, 42*(20), 7121-7131.

Wang, X., Wang, B., Li, T., Li, H., & Watada, J. (2023). Multi-criteria fuzzy portfolio selection based on three-way decisions and cumulative prospect theory. *Applied Soft Computing, 134*, 110033.

Wang, Z., Liu, S., & Kong, X. (2012). Artificial bee colony algorithm for portfolio optimization problems. *International Journal of Advancements in Computing Technology, 4*(4), 8-16.

Woodside-Oriakhi, M., Lucas, C., & Beasley, J. E. (2011). Heuristic algorithms for the cardinality

constrained efficient frontier. *European Journal of Operational Research, 213*(3), 538-550.

Yang, S.-C., Lin, T.-L., Chang, T.-J., & Chang, K.-J. (2011). A semi-variance portfolio selection model for military investment assets. *Expert Systems with Applications, 38*(3), 2292-2301.

Yin, X., Ni, Q., & Zhai, Y. (2015). *A novel PSO for portfolio optimization based on heterogeneous multiple population strategy*. Paper presented at the Evolutionary Computation (CEC), 2015 IEEE Congress on.

Yu, J., Ge, J., Heuveling, J., Schneider, E., & Yang, M. (2015). Structural basis for substrate specificity of an amino acid ABC transporter. *Proceedings of the National Academy of Sciences*, 201415037.

Yu, L., Wang, S., & Lai, K. K. (2008). Neural network-based mean–variance–skewness model for portfolio selection. *Computers & Operations Research, 35*(1), 34-46.

Yuen, S. Y., Chow, C. K., Zhang, X., & Lou, Y. (2016). Which algorithm should I choose: an evolutionary algorithm portfolio approach. *Applied Soft Computing, 40*, 654-673.

Zhang, P. (2015). Multi-period possibilistic mean semivariance portfolio selection with cardinality constraints and its algorithm. *Journal of Mathematical Modelling and Algorithms in Operations Research, 14*(2), 239-253.

Zhang, P. (2016). An interval mean–average absolute deviation model for multiperiod portfolio selection with risk control and cardinality constraints. *Soft Computing, 20*(3), 1203-1212.

Zhang, P. (2018). Chance-constrained multiperiod mean absolute deviation uncertain portfolio selection. *Journal of Industrial & Management Optimization*, 14208-14217.

Zhang, P., & Li, B. (2017). The Admissible Multiperiod Mean Variance Portfolio Selection Problem with Cardinality Constraints. *Industrial Engineering & Management Systems, 16*(1), 118-128.

Zhang, P., & Zhang, W.-G. (2014). Multiperiod mean absolute deviation fuzzy portfolio selection model with risk control and cardinality constraints. *Fuzzy Sets and Systems, 255*, 74-91.

Zhou, Z., Jin, Q., Xiao, H., Wu, Q., & Liu, W. (2018). Estimation of cardinality constrained portfolio efficiency via segmented DEA. *Omega, 76*, 28-37.

Zhou, Z., Liu, X., Xiao, H., Wu, S., & Liu, Y. A DEA-based MOEA/D algorithm for portfolio optimization. *Cluster Computing*, 1-10.