Engineering Applications of Artificial Intelligence 94 (2020) 103763

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

A new Kho-Kho optimization Algorithm: An application to solve combined emission economic dispatch and combined heat and power economic dispatch problem

Abhishek Srivastava, Dushmanta Kumar Das *

Department of Electrical and Electronics Engineering, National Institute of Technology Nagaland, India

ARTICLE INFO

Keywords:
Combined Emission Economic Dispatch problem (CEED)
Combined Heat and Power Economic Dispatch problem (CHPED)
Optimization technique

ABSTRACT

In this article, a new optimization technique known as Kho-Kho optimization (KKO) algorithm is presented. This proposed technique is a population based meta-heuristic method which is inspired from the strategies used by players in a well known tag-team game played in India, i.e. Kho-Kho. The performance and superiority of the proposed method with respect to other existing methods is evaluated using twenty nine benchmark functions and real-time optimization problems related to power system i.e. combined emission economic dispatch and combined heat and power economic dispatch problem.

1. Introduction

Optimization algorithms have wide applications in almost all the disciplines such as engineering, science, economics, etc. Researchers are always trying to develop new optimization algorithms so that these large and complex problems can be solved effectively. Numerical methods have difficulties in global optimality (Storn and Price, 1997). Nowadays, population-based heuristic and meta-heuristic algorithms have special attention. Many such algorithms are inspired by nature, animal behavior, etc.

The classical algorithms such as Quadratic Programming (QP) (Frank and Wolfe, 1956), Dynamic Programming (DP) (Bellman, 2013) and Lagrangian approach (Bertsekas, 1999) fail to search the global optima. These algorithms use to stuck in local optima. The drawbacks of the numerical methods are the main source of inspiration to develop some alternatives and many nature-inspired heuristic and meta-heuristic optimization algorithms. These are based on different natural processes and intelligence of animals in food searching, prey hunting, etc. A population-based lion optimization algorithm (LOA) is introduced in Yazdani and Jolai (2016), which is based on the lifestyle of lions and their co-operation characteristics. A meta-heuristic whale optimization algorithm (WOA) is presented in Mirjalili and Lewis (2016), which is inspired by the social behavior of humpback whales. In Zheng (2015), a water wave optimization (WWO) is introduced which is inspired by the shallow water wave theory. The herding behavior of elephants group is taken as an inspiration source to develop a new search method, i.e elephant herding optimization (EHO) in Deb et al. (2015). In Meng et al. (2014), the hierarchal order and behaviors of the chicken swarms are the source of developing new chicken swarm optimization (CSO) algorithm. It is also evident that the natural processes can also be a good source for developing new optimization techniques. A new algorithm named as mine blast optimization (MBO) based on mine bomb explosion concept is developed by the researches in Sadollah et al. (2013). In Yang (2012), a flower pollination algorithm is introduced where the source of inspiration is the pollination process of the flowers.

In recent year, many optimization techniques have been developed and used to solve many optimization problems. Based on the source of inspiration, they can be classified into different groups. One of the most interesting and widely used optimization group is population-based optimization algorithms (Karaboga and Akay, 2009). This group can be further divided into two groups which are evolutionary algorithms (Eiben et al., 2003) and swarm intelligence (Eberhart et al., 2001). Evolutionary algorithm include techniques such as genetic algorithm (GA) (Whitley, 1994), genetic programming (GP) (Banzhaf et al., 1998), evolutionary programming (EP) (Yao et al., 1999), etc. In recent years, swarm intelligence have also influenced many researchers. According to Bonabeau et al. (1999), swarm intelligence algorithms can be defined as the designing of intelligent agent algorithm taking into the account of the collective behavior of insects or animals. Several techniques based on swarm intelligence have been developed which include particle swarm optimization (PSO) inspired by social behavior of bird flocking (Kennedy and Eberhart, 1995), artificial bee colony (ABC) inspired by the cooperative behavior of bees (Karaboga and Akay, 2009), krill herd (KH) inspired by the herding behavior of krill individuals (Gandomi and Alavi, 2012), social spider optimization (SSO) inspired by social behavior of spiders (Cuevas et al., 2013), Firefly

* Corresponding author.
E-mail addresses: abhishek91026@gmail.com (A. Srivastava), dushmantakumardas29@gmail.com (D.K. Das).

https://doi.org/10.1016/j.engappai.2020.103763
Received 5 January 2020; Received in revised form 23 April 2020; Accepted 13 June 2020
Available online 17 June 2020

(FF) method inspired by the mating behavior of firefly inspects (Yang, 2010), Locust search (LS) inspired by the behavior of swarm of locusts (Cuevas et al., 2016), etc. Another interesting optimization group is music-based meta-heuristic optimization (Yang, 2009). This group includes technique such as harmony search (HS) which is inspired by the observation that aims to find the best music by a prefect combination of harmony (Yang, 2009), an improved harmony search (IHS) which is developed to improve the fine tuning characteristic of HS (Mahdavi et al., 2007), etc.

2. Novelty of the paper

In this paper, the authors have proposed a new population-based meta-heuristic optimization algorithm known as Kho-Kho optimization (KKO) algorithm. The proposed KKO algorithm is inspired by a well-known tag-team game played in India, i.e. Kho-Kho. The game is played between two teams, one is known as chasing team and other one is known as running team. During a match, a set of three players from the running team runs within the field and try to avoid being touched by the chasing team. These three players of the running team are termed as 'runners'. The players of the chasing team try to touch the runners within the permissible duration of time. For this, a unique strategy is used by the chasing team. Among the players of the chasing team, one player is selected as a chaser who tries to touch the runners. Remaining players of the team sit in rows with alternate players facing the opposite side. A chaser selects a new chaser from the sitting players so that the runner is close to him and it can be touched. Once the allotted time is over or all the players of the running team are touched, the second inning begins. In the second inning of the game, the role of chasing team and running team are interchanged. At the end of the second inning, who so ever score more points is declared as the winner. The proposed KKO algorithm mimics the unique strategy used by the chasing team to touch the runners (global best solution).

The performance of the proposed KKO algorithm is tested with twenty-nine benchmark functions. Next, the technique is used to solve large and highly non-linear problem related to power system i.e. a combined emission economic dispatch (CEED) and combined heat and power economic power dispatch (CHPED) problem.

3. Kho-Kho optimization (KKO) algorithm

3.1. Game information

Kho-Kho is a well-known tag-team game played in India. The game is played between the two teams, a chasing team and a running team. Both the team includes twelve players each. Among which only nine players from both the team play this game. The game includes two sections known as innings. During one inning, one team acts as a chasing team and other as a running team. During the second inning, the roles are interchanged i.e. running team chases and changing team runs.

During an inning, a set of three players from the running team occupy random positions on the field. These players are termed as runners and their role is to avoid being touched by the opposition team. These players can move freely within the field in any direction. On the other hand, one player is selected as a chaser from the chasing team and rest players sit in rows on the field as shown in Fig. 1. This sitting position is such that two adjacent players of the chasing team face opposite to each other. This unique sitting position helps the chasing team in switching the side and direction of movement as the chaser cannot switch his direction and side. Chaser can select any sitting player as the new chaser if he finds that a sitting player faces opposite to the chaser at a more optimal position concerning the runner. If this happens, the chaser takes the position of the new chaser on the field and the new chaser can decide to switch direction based on the position of the runners. The new chaser has to move his facing direction. This selection of new chaser is important and helps in touching the runners. Once a chaser touches a runner, the runner is replaced by a new runner. Once all the runners are touched or the chasing team runs out of time, the inning is stopped. At the end of two-innings, the team with less taken to out all the players or less number of players out in a fixed time wins the game.

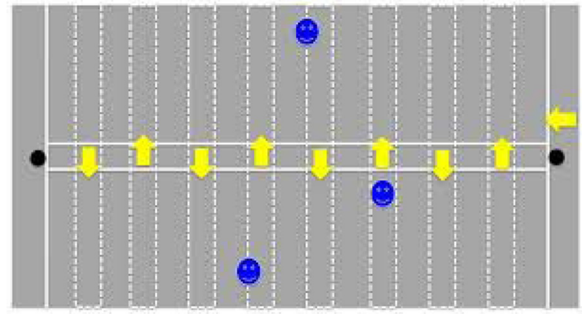

Fig. 1. Facing of chaser's.

The proposed KKO mimics the strategies used by the chasing team to touch a runner (global optimal solution).

3.2. Explanation to chasing strategy

In this section, the strategy used by the chasers to catch the runner is explained. According to the game, the chasers used to settle down in a straight line of chasing blocks in the center of the court in which two neighboring chasers are facing towards the two opposite sidelines. The chasers sitting at an even-numbered sitting block face towards one side of the center line and the chasers sitting at an odd-numbered sitting block face towards the other side of the centerline. This facing property categorizes the chasers in two categories: even position chasers and odd position chasers. The chasers which are facing towards the runner are the possible candidate to be selected as a chaser. The centerline divides the playing court into two halves. Throughout the game, the runner keeps moving randomly in the court. The even position chasers manage to chase and touch the runners in one half of the court while on the other half is managed by the odd position chasers. The interesting rule of the game is that only one chaser can stand up and attempt to chase and catch only when it satisfies the following conditions: 1. The chaser must face towards the current position of the runner, 2. The chaser must be sitting closest to a runner position.

Once a chaser is selected, he tries to chase and touch the runner. As per rules, a runner can move on the other side to avoid touching. In the case of a chaser, this cannot happen. Therefore, the present chaser passes its activeness to a sitting player-facing towards the other side of the court. For the selection of new chaser, the criterion remains the same. Once a new chaser is selected, the old chaser replaces the sitting position of the new chaser. This process is continued till a chaser touch the runner. Therefore, the chasers keep on changing their sitting blocks.

3.3. Mathematical model of chasing strategy

The behavior of both runners and a chaser on the court and strategies followed by the chaser to chase and touch out a runner can be mathematically modeled in this section.

3.3.1. Identification of runner position and type of the sitting players

To start the chase, the position of the runner should be known exactly. In the proposed algorithm, the fittest candidate serves as the runner position ($\overrightarrow{X}_r(t)$). In the beginning, all the chasers occupy space in a line of sitting blocks in which two consecutive players of the chasing team face opposite to each other. This opposite facing of the chasers generates two sets of the chasers. The classification of players in two sets is based upon a number at which the player is sitting in the line of chasing blocks. Therefore, the chasers are classified according to an even or an odd-numbered position. The set of even-numbered chasers is represented as:

$$C_j\,(j = 2, 4, 6....N) \subset C_i, (i = 1, 2, 3....N), \tag{1}$$

and the set of odd numbered chasers is represented as:

$$C_j\,(j = 1, 3, 5....N-1) \subset C_i, (i = 1, 2, 3....N), \tag{2}$$

3.3.2. Selection of a chaser

A chaser is one who stands up and chases a runner. To execute the chasing, a chaser must be selected among the players which are facing towards the runner's current position. In the proposed algorithm, a chaser ($\vec{C}_b(t)$) is the fittest candidate which virtually represent the runner position ($\vec{X}_r(t)$) on the court.

3.3.3. Chase and touch out (exploration and exploitation)

In the proposed algorithm, we have classified players in two sets which are even and odd-numbered position set. When a runner randomly moves in the court, both set of chasers reacts with different strategies. The selection of a chaser's strategy is totally dependent upon the position of a runner. If a runner is in front of any set, a player (chaser) out of the players of this set chases the runner. This process is named as chasing strategy. On the other hand, the players of the other set remain settled down. This strategy of settling down is named unmoved strategy. Both the strategies i.e. chasing and unmoved strategy happens simultaneously. Depending on the position of a runner, two possibilities can happen:

Possibility 1: The odd-numbered position chasers may follow the strategy of chasing and the even-numbered position chasers follow the strategy of settling down.

Possibility 2: The vice-versa of the possibility 1.

To select randomly anyone out of the two possibilities, an operator m is defined. Depending on the value of m, strategy for both the sets is decided. If m is less than or equal to 0.5, even-set follows the strategy of chasing and odd-set follows strategy of unmoved. If m is greater than, 0.5, odd-set follows the strategy of chasing and even-set follows strategy of unmoved.

The updating rule according to the strategy of chasing is mathematically represented as follows:

$$\vec{v}\,(t+1) = \vec{v_0}\,(t) + \left(\vec{C}_b(t) - \vec{C}_b(t)_{old}\right), \tag{3}$$

$$\vec{C}_f\,(t+1) = \vec{C}_b(t) - [g \cdot [e \cdot \vec{C}_b(t) - \vec{C}_f\,(t)]] + \vec{v}\,(t+1), \tag{4}$$

where $\vec{C}_f\,(t+1)$ represents the updated position of the chaser which are facing towards runner, $\vec{C}_f\,(t)$ is the position vector of the chasers facing towards the runner, $\vec{C}_b\,(t)$ is the position vector for active chaser or best chaser which is also representing the runner's position, $v(t)$ represents a feedback position of the runner from previous time instant, $v(t+1)$ is the updated feedback position, t is the present instant of time, $t+1$ is the next instant of time, scalar coefficient (g) is a gap randomizing operator and the operator $e \in [2, 0]$, an endurance factor indicating the runner endurance level.

The updating rule according to the strategy of unmoved is mathematically represented as follows:

$$\vec{C}_o(t+1) = \vec{C}_o + g \cdot [e \cdot \vec{C}_b(t) - \vec{C}_o\,(t)], \tag{5}$$

where $\vec{C}_o\,(t)$ are the position vectors for the chaser which are facing opposite to the runner.

The gap randomizing operator (g) is dependent on the endurance factor (e) of the chaser. This can be modeled as follows:

$$g = [(2 \cdot e \cdot rnd) - e], \tag{6}$$

Kho-Kho Optimization (KKO) Algorithm

Generate initial chasers population $X_i(i = 1, 2, ...N)$
Chasers settling in Kho-Kho formation
Odd position chasers, $C_i\,(i = 1, 3, 5...N-1)$
Even position chasers, $C_i\,(i = 2, 4, 6...N)$
Calculate the fitness of each chaser
Initialize runner position $\left(\vec{X_r}(t)\right)$, emdurance (e) and gap randomizing factor (g)
Initialize the best chaser position $\left(\vec{C_b}\right)$
while $t <$ max iteration
  for each solution
    Update $\left(\vec{X_r}(t)\right)$, (e), (g) and (m)
    if $(m \le 0.5)$
      Runner is in front of even position chasers
      Runner is in opposite of odd position chasers
      Update **the positions and speed** of the current **even position** chasers
      Update the **positions Only** of the current **odd position** chasers
      else
        Runner is in front of odd position chasers
        Runner is in opposite of even position chasers
        Update the **positions and speed** of the current **odd position** chasers
        Update the **positions only** of the current **even position** chasers
    end if
  end for
  Check if any solution goes beyond the search space and reject it.
  Calculate the fitness of the solution.
  Update $\left(\vec{C_b}\right)$, if there is a more satisfactory solution
  $t = t + 1$
end while
return $\vec{X_1}$.

Fig. 2. Algorithm of proposed KKO algorithm.

where 'rnd' is a random number within the range [0 , 1]

$$e = [e_T - \delta_e \cdot e_T], \tag{7}$$

where the operator e_T is the total endurance of the runner, which a fresh runner possesses in the beginning of its run. The rate of declination in the endurance level of a runner $\delta_e(t)$ is represented as:

$$\delta_e(t) = (t/t_R)^2. \tag{8}$$

where t represents the present instant of time and t_R is the run-time of the runner.

3.4. Endurance factor

The game of Kho-Kho is about running and chasing. The endurance factor is a very crucial physical factor which affects a runner performance. As the game progresses, there comes fatigue which leads to a declination in the endurance of a runner which affects its position on the court. Therefore, an endurance factor $e \in [2, 0]$ is associated with the runner position. The endurance level is considered as high for $e = 2$ and zero for zero levels. The endurance factor values with 1 to 2 represent moderate-high and moderate low for 0 to 1 value. As the runner starts running on the court, it is a natural phenomenon that the endurance level of the runner keeps on reducing at an increasing rate till he\she quits running and $\delta(e)$ is the rate of declination in endurance. During any chase, there exists a certain separating distance (gap) between the runner and a chaser. The chase is successful only when this separating distance minimizes towards zero. This gap keeps on varying throughout the game and it depends upon a few physical factors such as fatigue, obstacles, speed variation, decreased endurance etc. Therefore, an operator named as gap randomizing factor (g) is introduced and the value of g depends on the endurance (e) of the runner. All these above mention operators are quite necessary to mimic the situation of the game and behavior of the players.

3.5. Discussion for global optimality

The optimization procedure of proposed KKO algorithm begins with the selection of a runner (best particle) and chasers (candidate particles). The basic idea of this algorithm is to chase the best particle and identify the global best solution as the termination criteria satisfy. Here, chasing the best particle implies that keeping a track of movement (variations) of the best particle and improving the other candidate particles according to it. In a good optimization algorithm, the variations in the parent particle (best particle) should reflect effectively into the other candidate particles. In the proposed KKO, there are two cases of the candidate particles improvement (learning) are present. In both the case, candidate particles are improving themselves according to the movement of the best particle (chaser) but in the first case where the runner is in the front half of the court, an additional velocity improvement is also there. The velocity improvement is the variations in the best particle position concerning the old position of the best particle itself, whereas the primary improvement is the variations in the best particle concerning the other candidate particles. Therefore, the knowledge of both variations assists the other candidate particle vary effectively to improve. Now, in an optimization algorithm, the variations or improvement should be random within a defined search space and should be diminishing in nature. The constant variations will never lead the optimization algorithm to the global solution. In the proposed KKO, the random nature of gap varying factor (g) and the diminishing nature of endurance factor (e) leads the algorithm to achieve the global best solution without getting stuck into the local optimal solutions. In this algorithm, more emphasis is on exploration and less number of iterations are available for the exploitation as compared to the exploration. If the gap varying operator $(|g|) < 1$, the search is converging towards the solution and search is diverging from the solution if $(|g|) \geq 1$. More emphasis is given to the condition $(|g|) \geq 1$ as it makes the algorithm capable of utilizing the search space very efficiently. One last important thing is the termination criterion. The termination criterion or condition is the stage which declares any optimal solution as a global optimal. As the proposed algorithm is based on the chasing and making the runner touch out. Therefore, the termination criterion for the proposed algorithm is the condition when an active chaser chases the runner until its endurance reduces to nil and the chaser makes the runner touch out. The zero endurance (e) is the terminating criterion for the proposed KKO algorithm. This is the real game condition when the runner becomes exhausted due to continuous running and the runner has no endurance to run further and then the chaser makes him out.

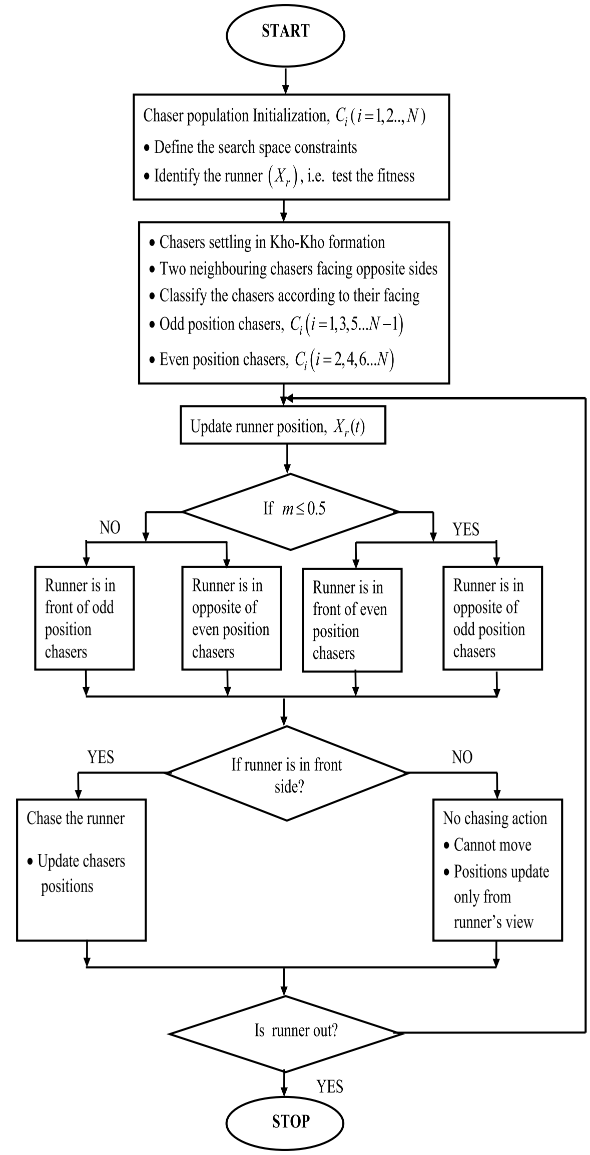

Fig. 3. Flowchart of proposed KKO algorithm.

4. Validation with benchmark functions

4.1. Introduction

The performance of a new optimization algorithm can be judged from its exploitation, exploration, global convergence and local optima avoidance capabilities. Therefore, to validate these capabilities of the proposed KKO algorithm, twenty-nine benchmark functions (BF's) (García et al., 2009) are used. These functions comprise of three groups (i) uni-modal, (ii) multi-modal and (iii) composite function. Uni-modal functions include seven different functions (F_1 to F_7) which have unique global optimal points. Using these functions, the exploitation capabilities of the proposed KKO algorithms can be evaluated. Next group i.e. multi-modal group includes sixteen different function (F_8 to F_{23}). These functions have a single global optimal point along with multiple local minima. These functions are therefore used to evaluate the exploration capabilities of the proposed algorithm. At last, six composite functions (F_{24} to F_{29}) are used to test the balance between exploration and exploitation. This also gives information about the local minima avoidance capability.

4.2. Result and discussions

According to Zhang et al. (2019), optimization algorithms may not provide a globally optimal solution every time because of the initial value of random population (candidate solution). But, they can provide a sub-optimal solution close to an optimal solution in a short computational time. Therefore, each function is simulated for fifteen times. This helps in computation of important parameters such as minimum value (best result), maximum value (worst result), average value (AVG), standard deviation (STD) and p-value. Average value provides the average results for the fifteen simulations while standard deviation helps to analyze the deviation in the results. As the objective while solving these functions is to minimize the function, it is better to get smaller average values. Also, the deviation for every case should be as small as possible. All fifteen simulations are simulated with an iteration count of 500. Player population is considered to be 50. The result obtained for the three groups of benchmark functions is discussed as follows.

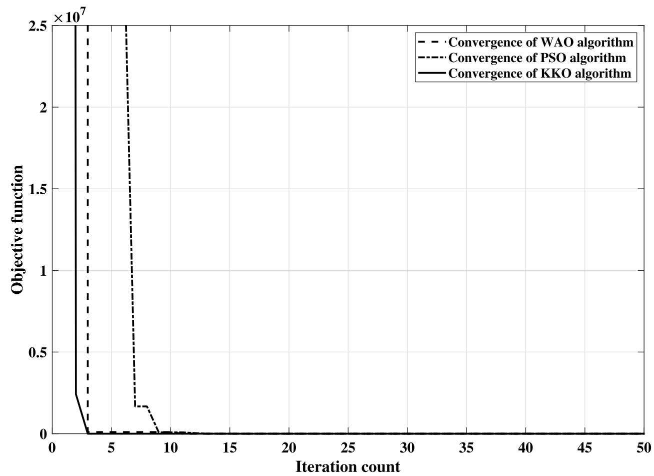

Fig. 4. Convergence profile of F_2.

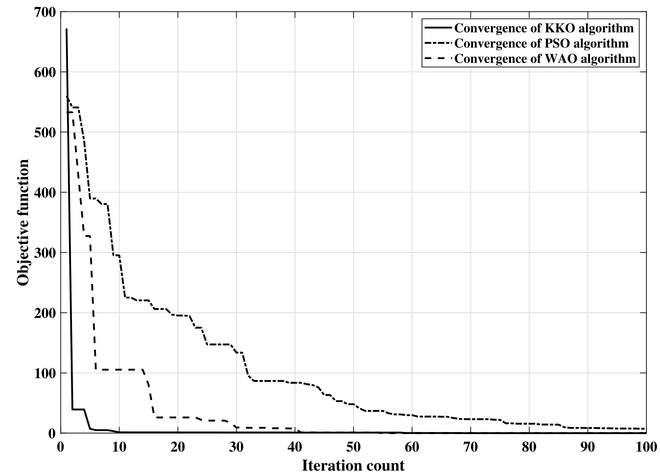

Fig. 5. Convergence profile of F_{11}.

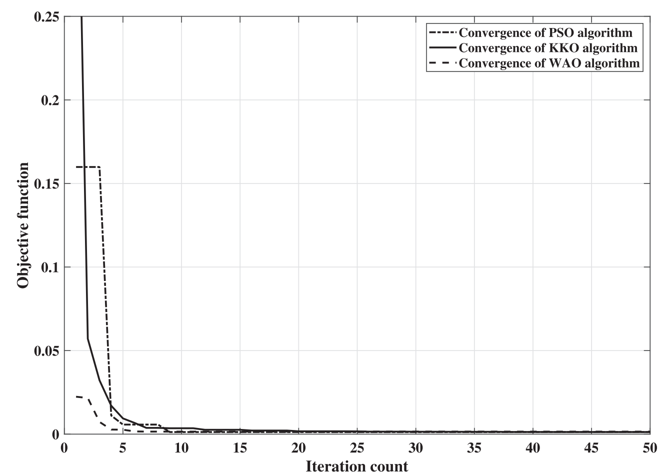

Fig. 6. Convergence profile of F_{15}.

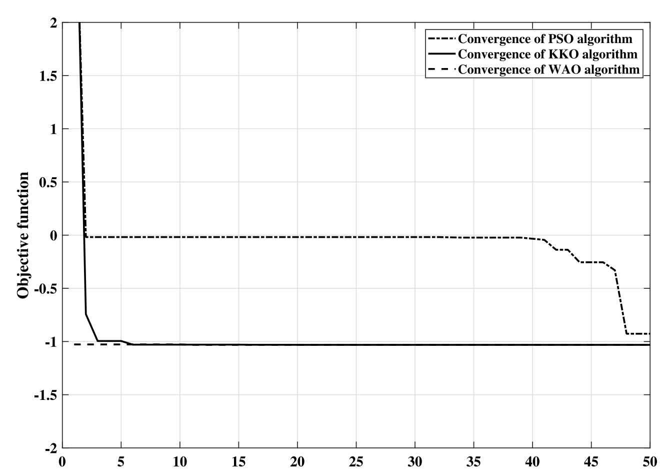

Fig. 7. Convergence profile of F_{16}.

4.2.1. Test with uni-modal benchmark functions

The average and standard deviation obtained for the uni-modal functions is presented in Table 2 along with the existing results for DE, WAO, GSA, PSO, etc. It is observed that for this group of functions, KKO provides the best result for functions F_1, F_2, F_3 and F_7. The deviation in the result is also small as compared to other algorithms used for comparison for functions F_1, F_2, F_3 and F_7. This proves that the proposed KKO exhibits better exploitation capabilities.

4.2.2. Test with multi-modal benchmark functions

Table 2 also presents the average and standard deviation value of the sixteen multi-modal functions along with existing results for DE (Storn and Price, 1997), WAO (Mirjalili and Lewis, 2016), GSA (Rashedi et al., 2009), and PSO (Kennedy and Eberhart, 1995). It is observed that the proposed KKO provide the optimal results for functions F_9, F_{11}, and F_{17}. KKO provides third optimal results for functions F_{10} and forth best for other functions. From this analysis, it can be inferred that the proposed algorithm exhibits good exploration capabilities. (See Table 1.)

4.2.3. Test with composite functions

Composite functions are used to test the balance between exploration and exploitation. The average and standard deviation obtained for the six composite functions is presented in Table 2. For comparison, existing results of DE (Storn and Price, 1997), WAO (Mirjalili and Lewis, 2016), GSA (Rashedi et al., 2009), and PSO (Kennedy and Eberhart, 1995) are also presented in Table 2. It is observed that for functions F_{24} to F_{29} the proposed KKO provides better result compared to existing techniques. The deviation for these functions is also less. These test dues prove that the proposed technique exhibits a good balance between exploration and exploitation.

4.2.4. Convergence analysis for KKO algorithm

To analyze the convergence behavior of the proposed KKO algorithm, convergence profile of the functions F_2, F_{11}, F_{15}, F_{16} and F_{24} are examined. The convergence profile of WAO and PSO is used for the comparison. The convergence plot for these functions are presented in Figs. 4–8 respectively. These figures show that the proposed algorithm exhibits a faster and better convergence profile that the existing techniques WAO and PSO.

5. Asymptotic analysis of the proposed algorithm

In the asymptotic analysis, the run time performance of an optimization algorithm is computed in terms of input parameter size. The proposed KKO algorithm includes some input parameters which

Table 1
Statistical data for benchmark function from fifteen simulation.

BF's	Best result	Worst result	AVG	STD	p-value (WAO)	p-valve (PSO
F_1	6.83E−55	1.33E−33	1.30E−35	3.74E−35	0.0271	0.6627
F_2	6.12E−29	1.48E−22	3.60E−23	6.41E−23	0.0271	0.1858
F_3	3.15E−72	1.90E−34	1.58E−35	5.48E−35	0.7618	0.1858
F_4	4.81E−33	7.29E−18	6.05E−19	2.01E−18	0.3790	0.008
F_5	28.8566	28.9732	28.90336	0.038939	0.0905	0.0024
F_6	4.38972	7.1798	5.788447	0.911463	0.0014	4.9752E−11
F_7	1.486E−06	0.00025862	8.77456E−05	8.29E−05	0.8766	1
F_8	−4176.42	−2935.58	−3758.63	408.5173	0.0064	0.0064
F_9	0	0	0	0	1.1148E−14	1.6853E−14
F_{10}	0	20.4577	11.24042	9.234612	0.0773	1
F_{11}	0	0	0	0	0.4204	0.3790
F_{12}	−0.8471	0.160425	−0.30359	0.360173	0.1023	0.4825
F_{13}	2.461	4.22067	3.302702	0.729056	0.4290	8.1014E−10
F_{14}	1.01987	1.99837	1.28144	0.478666	0.333	0.333
F_{15}	0.001069	0.003337	0.002303	0.001076	0.8456	0.1463
F_{16}	1.03136	−0.99652	−1.02271	0.01186	0.6667	0.6667
F_{17}	0.39868	0.440983	0.412356	0.017307	0.4655	0.4655
F_{18}	3.01404	3.63975	3.206278	0.227625	0.6667	0.6667
F_{19}	−3.7312	−3.42052	−3.59924	0.12549	0.7	1
F_{20}	−2.51678	−1.57962	−2.13964	0.31007	0.132	0.156
F_{21}	−4.35158	−1.05939	−2.90419	1.482463	0.0286	0.9408
F_{22}	−5.67592	−2.90984	−4.14019	−1.093177	0.0286	0.8857
F_{23}	−4.55522	−2.62289	−3.97328	0.674575	0.0154	0.0286
F_{24}	0.295574	41.4243	3.94335	8.39575	0.8494	0.9448
F_{25}	0.096249	54.1581	4.4041	8.91397	0.6548	0.3484
F_{26}	0.005098	124.0245	6.3271	23.8971	0.1456	0.7461
F_{27}	0.82401	84.2746	8.76504	30.4813	0.4286	0.1876
F_{28}	2.4954	73.2484	9.8846	36.2415	0.4481	0.8951
F_{29}	1.21806	44.01587	4.13161	36.2415	0.8453	0.9842

Table 2
Comparative analysis of AVG and STD for benchmark function.

BF's	DE (Storn and Price, 1997)		PSO (Kennedy and Eberhart, 1995)		GSA (Rashedi et al., 2009)		WAO (Mirjalili and Lewis, 2016)		KKO	
	AVG	STD	AVG	AVG	AVG	AVG	AVG	STD	AVG	STD
F_1	8.2E−14	5.9E−14	0.000136	0.000202	2.53E−16	9.67E−17	1.41E−30	4.91E−30	**1.30E−35**	**3.74E−35**
F_2	1.5E−09	9.9E−10	0.042144	0.045421	0.055655	0.194074	1.06E−21	2.39E−21	**3.60E−23**	**6.41E−23**
F_3	6.8E−11	7.4E−11	70.12562	22.11924	896.5347	318.9559	5.39E−07	2.93E−06	**1.58E−35**	**5.48E−35**
F_4	**0**	**0**	1.086481	0.317039	7.35487	1.741452	0.072581	0.39747	6.05E−19	2.01E−18
F_5	**0**	**0**	96.71832	60.11559	67.54309	62.22534	27.86558	0.763626	28.90336	0.038939
F_6	**0**	**0**	0.000102	8.28E−05	2.5E−16	1.74E−16	3.116266	0.532429	5.788447	0.911463
F_7	0.00463	0.0012	0.122854	0.044957	0.089441	0.04339	0.001425	0.001149	**8.77456E−05**	**8.29E−05**
F_8	**−11080.1**	574.7	−4841.29	1152.814	−2821.07	493.0375	−5080.76	695.7968	−3758.63	408.5173
F_9	69.2	38.8	46.70423	11.62938	25.96841	7.470068	**0**	**0**	**0**	**0**
F_{10}	9.7E−08	4.2E−08	0.276015	0.50901	0.062087	0.23628	7.4043	9.897572	11.24042	9.234612
F_{11}	**0**	**0**	0.009215	0.007724	27.70154	5.040343	0.000289	0.001586	**0**	**0**
F_{12}	**7.9E−15**	**8.E−15**	0.006917	0.026301	1.799617	0.95114	0.339676	0.214864	−0.30359	0.360173
F_{13}	**5.1E−14**	**4.8E−14**	0.00675	0.008907	8.899084	7.126241	1.889015	0.266088	3.302702	0.729056
F_{14}	**0.998004**	**3.3E−16**	3.627168	2.560828	5.859838	3.831299	2.111973	2.498594	1.28144	0.478666
F_{15}	**4.5E−14**	**0.00033**	0.000577	0.000222	0.003673	0.001647	0.000572	0.000324	0.002303	0.001076
F_{16}	−1.03163	3.1E−13	−1.03163	6.25E−16	**−1.03163**	**4.88E−16**	−1.03163	4.2E−07	−1.02271	0.01186
F_{17}	0.397887	9.4E−09	**0.397887**	**0**	**0.397887**	**0**	0.397914	2.7E−05	0.412356	0.017307
F_{18}	3	2E−15	3	1.33E−15	3	4.17E−15	3	4.22E−15	**3**	**0**
F_{19}	NA	NA	**−3.86278**	2.58E−15	−3.86278	**2.29E−15**	−3.85616	0.002706	−3.59924	0.12549
F_{20}	NA	NA	**−3.26634**	0.060516	−3.31778	0.023081	−2.98105	0.376653	−2.13964	0.31007
F_{21}	**−10.1532**	**0.0000025**	−6.8651	3.019644	−5.95512	3.737079	−7.04918	3.629551	−2.90419	1.482463
F_{22}	**−10.4029**	**3.9E−07**	−8.45653	3.087094	−9.68447	2.014088	−8.18178	3.829202	−4.14019	−1.093177
F_{23}	−10.5364	1.9E−07	−9.95291	1.782786	**−10.5364**	**2.6E−15**	−9.34238	2.414737	−3.97328	0.674575
F_{24}	6.75E−2	1.11E−1	100	81.65	**6.63E−17**	**2.78E−17**	0.568846	0.505946	3.94335	8.39575
F_{25}	28.759	**8.6277**	155.91	13.176	200.6202	67.72087	75.30874	43.07855	**4.4041**	8.91397
F_{26}	144.41	**19.401**	172.03	32.769	180	91.89366	55.65147	21.87944	**6.3271**	23.8971
F_{27}	324.86	**14.784**	314.3	20.066	170	82.32726	53.83778	21.621	**8.76504**	30.4813
F_{28}	10.789	**2.604**	83.45	101.11	200	47.14045	77.8064	52.02346	**9.8846**	36.2415
F_{29}	490.94	39.461	861.42	125.81	142.0906	88.87141	57.88445	**34.44601**	**4.13161**	36.2415

influences the running time performance. These input parameters in KKO algorithm includes number of iteration (T_{max}), number of players (N_{max}) and position of the players (P_{max}). Depending on these input parameters, the operations to be executed by the proposed algorithm are presented as follows:

1. $T_{max} + 1$
2. $T_{max} \times N_{max} \times P_{max}$
3. $T_{max} \times (N_{max} + 1)$
4. $T_{max} \times (N_{max} + 1)$
5. $T_{max} \times N_{max} \times P_{max}$

The total number of operations (O_{total}) can be computed using the above operations as follows:

$$O_{total} = (T_{max} + 1) + (T_{max} \times N_{max} \times P_{max}) + (T_{max} \times (N_{max} + 1))$$

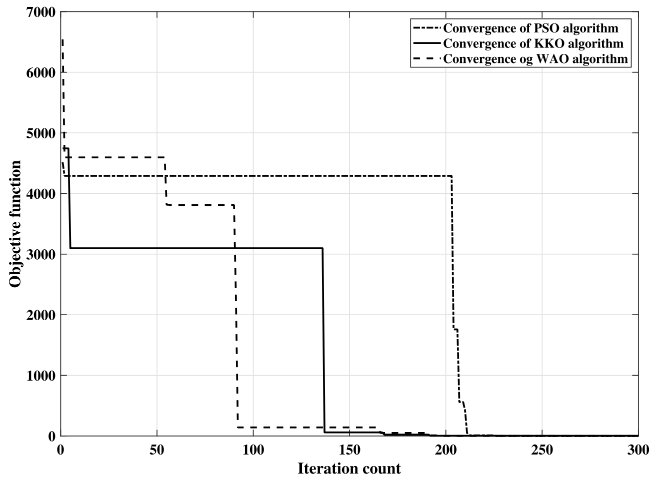

Fig. 8. Convergence profile of F_{24}.

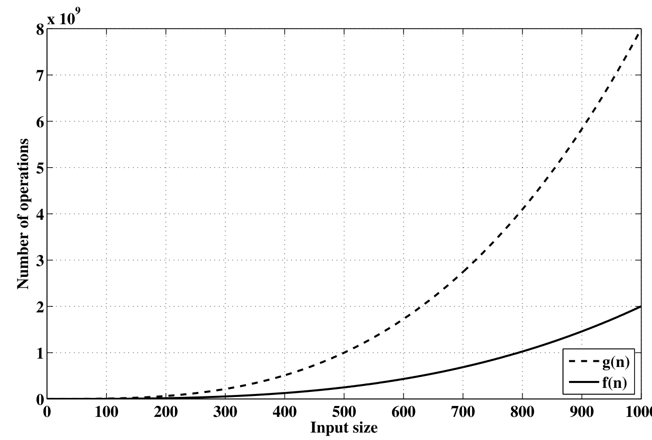

Fig. 9. Asymptotic behavior of KKO.

Table 3
Comparative time complexity analysis.

Technique	Time complexity (under worst case)
PSO (Kennedy and Eberhart, 1995)	$O(n^4)$
MBCO (Das et al., 2018a)	$O(n^4)$
CTO (Das et al., 2018b)	$O(n^4)$
WAO (Mirjalili and Lewis, 2016)	$O(n^3)$
KKO	$O(n^3)$

$$+(T_{max}\times(N_{max}+1)) \\ +(T_{max}\times N_{max}\times P_{max}). \tag{9}$$

$$O_{total} = 2\times(T_{max}\times N_{max}\times P_{max}) \\ +2\times(T_{max}\times(N_{max}+1)) \\ +3\times(T_{max})+1. \tag{10}$$

Now, for the run time performance analysis, all the input parameters are considered to be equal for the worst case scenario. Hence, (10) can be written as a function of n as

$$f(n) = 2\times n^3 + 2\times n^2 + 3\times n + 1 \tag{11}$$

Here, we assume that $f(n)$ is big-O of n^3, i.e. $g(n) = n^3$. Hence, $f(n) = O(g(n))$ if and only if, there exists two positive constants c and n_0 such that

$$|f(n)| \le c|g(n)| \tag{12}$$

for all n $\ge n_0$ and $f(n)$ is non-negative. Therefore, (12) can be written as

$$0 \le f(n) \le cg(n) \tag{13}$$

for all n $\ge n_0$. c can be computed by adding all coefficients of (14) as (8). Then, if n $\ge n_0 = 1$,

$$2\times n^3 + 2\times n^2 + 3\times n + 1 \le 2\times n^3 + 2\times n^3 + 3\times n^3 + n^3 \tag{14}$$

thus $2\times n^3 + 2\times n^2 + 3\times n + 1 = O(n^3)$. Therefore, $f(n) = O(g(n)) = O(n^3)$. By this it can be concluded that as n is increased, $f(n)$ will always be less than or equal to $g(n)$. For further understanding, Fig. 9 shows that as n is increased, $f(n)$ is either equal or less than to $g(n)$. To further analyze the performance of KKO with respect to other existing technique on basic of time complexity analysis, Table 3 is presented. It is observed that the proposed KKO requires less number of operation and hence less time to compute the results.

6. Convergence analysis of KKO

In this section, convergence of KKO is analyzed mathematically using differential equation. For this analysis some assumptions are taken into consideration. Let us consider that the problem space is one dimensional and the number player is one. Next, it is also assumed that endurance factor is 1. Now, the odd and even conditions are analyzed one by one.

Considering the odd condition along with the above assumptions, (5) can be simplified as:

$$C_o(t+1) = C_o(t) + g\cdot[C_b(t) - C_o(t)]. \tag{15}$$

Let $y(t) = C_b(t) - C_o(t)$,

$$C_o(t+1) = C_o(t) + g\cdot y(t). \tag{16}$$

At $(t+1)$ instant, (16) can be written as:

$$C_o(t+2) = C_o(t+1) + g\cdot y(t+1). \tag{17}$$

Considering (16) and the assumption $y(t) = C_b(t) - C_o(t)$, $y(t+1)$ can be written as:

$$y(t+1) = (1-g)\left(\frac{C_o(t+1) - C_o(t)}{g}\right). \tag{18}$$

Replacing $y(t+1)$ from (18) to (17), (17) can be rewritten as:

$$C_o(t+2) + (2-g)\cdot C_o(t+1) + (g-1)\cdot C_o(t) = 0. \tag{19}$$

Making (19) to be continuous and a second order differential equation is derived as:

$$\frac{d^2C_o}{dt^2} + \ln(e_1.e_2)\frac{dC_o}{dt} + \ln(e_1).\ln(e_2)C_o = 0. \tag{20}$$

where e_1 and e_2 are the roots of the quadratic equation (19) which can be further solved to $e_1 = (g-2)/2 + \sqrt{(2-g)^2 + 4(1-g)}/2$ and $e_2 = (g-2)/2 - \sqrt{(2-g)^2 + 4(1-g)}/2$. Hence, the general solution of the differential equation (20) can be written as follows:

$$C_o(t) = K_1\cdot e_1^t + K_2\cdot e_2^t. \tag{21}$$

Further $y(t)$ can be thus written as:

$$y(t) = \frac{K_1\cdot e_1^t(e_1-1) + K_2\cdot e_2^t(e_2-1)}{g}. \tag{22}$$

where $K_1 = \frac{C_o(0)(e_2+1)+g.y(0)}{e_1+e_2}$ and $K_2 = \frac{C_o(0)(e_1-1)+g.y(0)}{e_1+e_2}$

In a similar way, the even condition along with the assumptions can be solved and (4) can be simplified to a quadratic equation which is represented as follows:

$$C_f(t+2) + (2+g)C_f(t+1) - (1+g)C_f(t) - (2+g)v(t) = 0. \tag{23}$$

Here, for simplicity $v(t)$ is assumed to be zero then (23) can be rewritten as:

$$C_f(t+2) + (2+g)C_f(t+1) - (1+g)C_f(t) = 0. \tag{24}$$

Making (24) to be continuous and a second order differential equation is derived as:

$$\frac{d^2C_f}{dt^2} + \ln(p_1.p_2)\frac{dC_f}{dt} + \ln(p_1).\ln(p_2)C_f = 0. \tag{25}$$

where p_1 and p_2 are the roots of the quadratic equation (24) which can be further solved to $p_1 = (g+2)/2 + \sqrt{(2+g)^2 + 4(1+g)}/2$ and $p_2 = (g+2)/2 - \sqrt{(2+g)^2 + 4(1+g)}/2$. Hence, the general solution of (20) can be written as follows:

$$C_f(t) = L_1 \cdot p_1^t + L_2 \cdot p_2^t. \tag{26}$$

Further $y(t)$ can be thus written as:

$$y(t) = \frac{L_1 \cdot p_1^t(p_1 - 1) + L_2 \cdot p_2^t(p_2 - 1)}{g}. \tag{27}$$

where $L_1 = \frac{-C_f(0)(e_2-1)+g.y(0)}{e_1-e_2}$ and $L_2 = \frac{C_f(0)(e_1-1)-g.y(0)}{e_1-e_2}$

Now, from (21), (22), (26) and (27), if t tends to infinity, then $C_o(t)$, $C_f(t)$ and $y(t)$ will tend to zero. This shows that KKO convergence to a global optima.

7. Advantage and disadvantage of proposed KKO

The uniqueness of the proposed KKO optimization technique with respect to other existing optimization scheme (Storn and Price, 1997; Mirjalili and Lewis, 2016; Rashedi et al., 2009; Kennedy and Eberhart, 1995; Gandomi and Alavi, 2012; Cuevas et al., 2013; Yang, 2010) is the use of two groups for the search of the global optimal solution. In KKO, the population of the players are divided into two groups. Players of each group update their position depending on the position of the runner (optimal solution) using different update rules. This increases the diversity of the search and improves the ability of finding the global optimal solution. This can also be observed from the study with the benchmark functions presented in Section 4. The proposed KKO technique provides the best average solution in twelve functions out of twenty-nine functions which surpasses the results of other obtained using other techniques (Storn and Price, 1997; Mirjalili and Lewis, 2016; Rashedi et al., 2009; Kennedy and Eberhart, 1995). This proves that the searching diversity of KKO is better in comparison to other techniques apart from good exploitation, exploitation, convergence capabilities. It is also observed that this approach also improves the convergence profile of the KKO techniques which can be concluded from the Figs. 4–8.

Further, it is also observed that the standard deviation for the twenty-nine benchmark functions obtained for KKO is not the best. In case of standard deviation, differential evolution provides a much stable result in comparison to KKO technique. This concludes that KKO has more deviation in the results. But, still the this is better compared to PSO, WAO etc.

8. KKO to solve combined emission economic dispatch (CEED) problem

8.1. Introduction

Increasing environmental problem due to the burning of fossil fuel in thermal power plants is a big concern for the power industries. Burning of fossil fuels in these industries emits harmful gases like carbon dioxide, sulfur dioxide, nitrogen dioxide, etc. These gases are not only harmful to the environment but also for the human being. To handle this problem power industries changed their objective from an economic load dispatch problem to a CEED. The objective while solving a CEED problem is to minimize the overall generation cost as well as minimize the emission rate. Along with these objectives, different constraints should also be fulfilled.

Numerous methodologies have been developed to solve this highly non-linear and complex multi-objective problem more effectively. Some of the recent works in this domain are lightning flash algorithm (Kheshti et al., 2017), stochastic fractal search algorithm (Alomoush and Oweis, 2018), interior search algorithm (Karthik et al., 2019), parallel hurricane optimization algorithm (Rizk-Allah et al., 2018), multi-objective hybrid bat algorithm (Liang et al., 2018), enhanced moth-flame optimizer (Elsakaan et al., 2018), bio-inspired algorithms (Dey et al., 2019), chaotic improved harmony search algorithm (Rezaie et al., 2019), reinforcement learning based on non-dominated sorting genetic algorithm (Bora et al., 2019), Quantum-behaved bat algorithm (Mahdi et al., 2018), Double weighted particle swarm optimization (Kheshti et al., 2018), Floating search space (Amiri et al., 2018), Chaotic self-adaptive interior search algorithm (Rajagopalan et al., 2019), modified shuffle frog leaping algorithm (Elattar, 2019), improved TLBO (Joshi and Verma, 2019), new sine cosine algorithm (Gonidakis and Vlachos, 2019). Though many optimization algorithms are developed and used to solve the CEED problem. However, new techniques are always entertained to have more efficient and effective results.

In this section, the proposed KKO algorithm is tested with five test system of a CEED problem. Before moving towards the test systems, the mathematical model of the CEED problem is discussed in the next section.

8.2. Problem formulation

As stated earlier, the objective for a CEED problem is to optimize power generation of generating units such that the overall fuel cost and emission rates should be minimum. As there are two objective, CEED problem is considered as a multi-objective problem.

The overall fuel cost for the generating units is computed using the following expression:

$$F_c = \sum_{i=1}^{m} (a_i G_i^2 + b_i G_i + c_i) + |g_i \times \sin(h_i \times (G_i^{\min} - G_i))|, \tag{28}$$

where F_C is the total fuel cost; a_i, b_i c_i are fuel cost coefficients of the ith generating unit; g_i and h_i are coefficients of the ith generator due to valve point effect; G_i is the real active power generated from the ith generating units and m is total number of generating units in the power system.

Similarly, for the evaluation of the emission rate for the m generating units, the mathematical equation used is given as.

$$F_E = \sum_{i=1}^{m} \alpha_i G_i^2 + \beta_i G_i + \gamma_i + \eta_i \times exp(\delta_i \times G_i), \tag{29}$$

where E_C is the total emission, α_i, β_i, γ_i, η_i, δ_i are the emission coefficients of the ith generating units.

Hence, by combining the two objective functions, the objective for CEED problem is formulated as:

$$C = F_C + p * F_E, \tag{30}$$

where C is the objective function for a CEED problem to be minimized and p is referred as a penalty factor.

The penalty factor provides a ratio between the average fuel cost computed for maximum power capacity for each generating unit and average emission computed for maximum power capacity for the same. The computation of penalty factor p for a given load can be done as present in Abdelaziz et al. (2016a). It is used to convert a dual objective CEED problem to a single objective CEED problem.

8.3. Constraints

8.3.1. Generation operating limits

According to this, all the generating units should operate within a prescribed limit of generation. Mathematically, it is given as:

$$G_i^{\min} \leq G_i \leq G_i^{\max}, i = 1, 2, 3...., n, \tag{31}$$

where $G_i^{\min}$ is the minimum power output of ith generating unit (in MW) and $G_i^{\max}$ is the maximum output power of ith generating unit (in MW).

8.3.2. Power balance criterion

During transmission and generation of power, losses are always associated. For this reason, it is important that the total power generation is generated in such a way that the following equation is fulfilled.

$$\sum_{i=1}^{m} G_i = D_P + L_P, \tag{32}$$

where D_P is the total power demand and L_P represents the power loss in transmission.

The total transmission loss is computed by:

$$L_P = \sum_{i=1}^{m} \sum_{j=1}^{m} G_i B_{ij} G_j, \tag{33}$$

where B_{ij} denotes the element in loss coefficient matrix.

8.4. Results and discussions

The proposed KKO algorithm is tested with five test systems. These test systems are used to evaluate the performance and effectiveness of proposed KKO to solve the CEED problem. The characteristics of these five systems are presented in Table 4. Each system is simulated for fifteen times with an iteration count of 500 and players population of 150. The statistical data obtained form the fifteen individual runs are presented in Table 5.

For test system V.i, the best result obtained is presented in Table 6. For a comparative analysis of the result, some existing results of well-known techniques like FA (Gherbi et al., 2016), BA (Gherbi et al., 2016), HYB (Gherbi et al., 2016), GA (Gherbi et al., 2016), PSO (Elattar, 2019), FPA (Abdelaziz et al., 2016b), and MSFLA (Elattar, 2019) are also presented in Table 6. It is observed that the proposed technique performs better than the existing techniques. Proposed KKO optimized the generator units to provide better fuel cost and emission rates with reduced losses. The convergence behavior of the objective function in presented in Fig. 10.

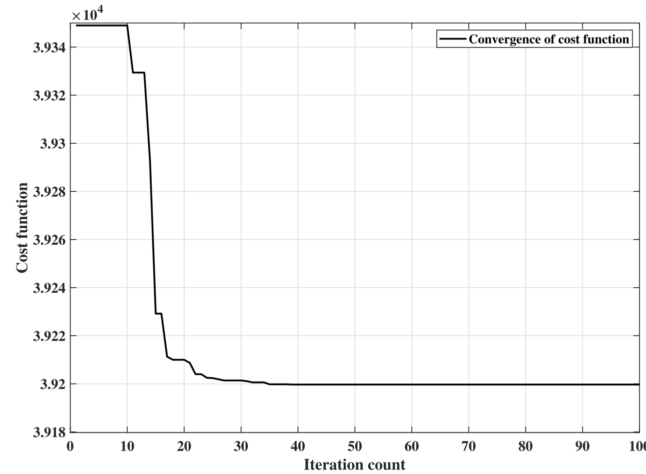

Fig. 10. Convergence profile for test system V.i.

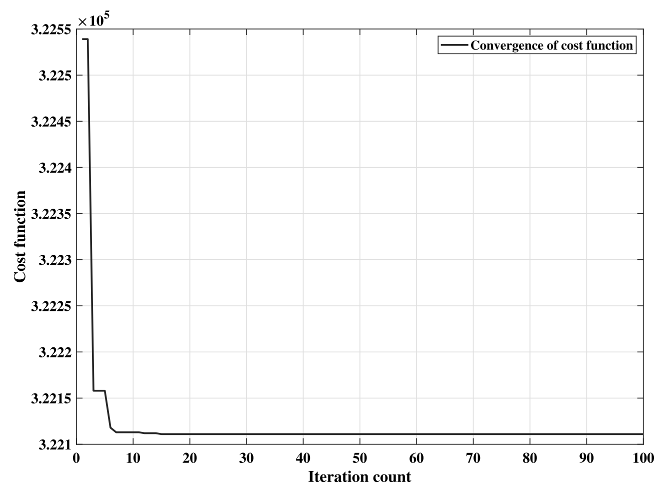

Fig. 11. Convergence profile for test system V.ii.

For test system V.ii, the optimal fuel cost and emission rate obtained using the proposed KKO algorithm is presented in Table 7. For comparative analysis of the result obtained, existing results of NSGA-II (Basu, 2011), PDE (Basu, 2011), SPEA-2 (Basu, 2011), GSA (Güvenç et al., 2012), ABC-PSO (Manteaw and Odero, 2012), EMOCA (Zhang et al., 2013), FPA (Abdelaziz et al., 2016a) and LFA (Kheshti et al., 2017) are used. These results are also presented in Table 7. On comparing the above mentioned techniques, it is observed that the proposed KKO provides the best emission rate where as LFA provides the best fuel cost. The convergence behavior of the objective is presented in Fig. 11.

Test system V.iii is simulated under two simulation condition. In first condition, generators are optimized to provide best fuel cost only. Under this condition, the optimal power allocation that provides best fuel cost is presented in Table 8. For comparative analysis results of existing techniques like MHBA (Liang et al., 2018), FSBF (Liang et al., 2018), NSBF (Liang et al., 2018) are also presented in Table 8. It is observed that proposed KKO algorithm optimizes the generator units more effectively with reduced losses. For the second condition, generators are optimized to provide best emission rate. For this condition, the optimal power allocation that provides better emission is presented in Table 9. Results for MHBA (Liang et al., 2018), FSBF (Liang et al., 2018), NSBF (Liang et al., 2018) is also used in this case for comparative analysis. It is clear from the analysis that proposed KKO provides the best emission rate for this case. The convergence behavior of the objective function in case of best fuel and emission is presented in Figs. 12 and 13 respectively.

For test system V.iv, the results of existing technique like MODE (Basu, 2011), PDE (Basu, 2011), NSGA-II (Basu, 2011), SPEA-2 (Basu, 2011), GSA (Güvenç et al., 2012), MABC/D/C (Secui, 2015), MABC/D/Log (Secui, 2015), FPA (Abdelaziz et al., 2016b) and ISA (Karthik et al., 2019) along with the proposed KKO technique is presented in Table 10. It is observed that ISA provides the best result for this system whereas the proposed CSOA provides third best result for this case. The convergence of the objective function for this case is presented in Fig. 14.

For test system V.v, the optimal power allocation obtained for the fourteen generators is presented in Table 11. Techniques such as CSAISA (Rajagopalan et al., 2019), ISA (Rajagopalan et al., 2019), HSA (Jeddi and Vahidinasab, 2014), DE (Rajagopalan et al., 2019), PSO (Rajagopalan et al., 2019) and GA (Rajagopalan et al., 2019) are used for a comparative analysis. All the results are presented in Table 11. It is clear from the comparative analysis that the proposed KKO provides the best fuel cost and emission rate. Further, the convergence behavior of the objective function is presented in Fig. 15.

Table 4
Characteristic of the ten test systems.

Problem	System	No. of units	Power demand	Heat demand	VPE	Loss
CEED	V.i (Gherbi et al., 2016)	3	500	–	✓	✓
	V.ii (Abdelaziz et al., 2016b)	10	2000	–	✓	✓
	V.iii (Liang et al., 2018)	IEEE 30 Bus	2.834	–	✓	✓
	V.iv (Kheshti et al., 2017)	40	10500	–	✓	–
	V.v (Rajagopalan et al., 2019)	IEEE 118 Bus	950	–	✓	✓
CHPED	VI.i (Sun and Li, 2019)	4	200	115	✓	–
	VI.ii (Haghrah et al., 2016)	5	250	175	✓	
	VI.iii (Sun et al., 2017)	7	600	150	✓	✓
	VI.iv (Nazari-Heris et al., 2019)	24	2350	1250	✓	–
	VI.v (Basu, 2016)	48	4700	2500	✓	–

Table 5
Statistical data for the ten test systems.

System	Method	Best	Worst	AVG	STD
Case V.i	KKO	39199.7	39200	39199.77	0.121106
	PSO	39210.20	39261	39229.02	23.29
	FPA	39210.15	39244.9	39216.49	15.48465
Case V.ii	KKO	311995	322464	322218.8	177.5696
	FPA	321927	322604	322201	356.4
	LFA	320914	322087	321278.66	701.15
Case V.iii	KKO	605.68	610.364	606.8468	2.106488
	MHBA	607.3897	607.5692	607.4492	–
Case V.iv	KKO	10056	10185.2	10127.02	60.77781
	FPA	121074.5	121095.7	121196.3	–
	ISA	120385.47	120403.21	120478.62	–
Case V.v	KKO	251245	251345	251281.2	21.70714
	PSO	251268	251456	251298.1	35.6122
Case VI.i	KKO	9217.03	9527.75	9300.26	113.6207
	IGA-NCM	9257.075	9257.9014	9257.1553	0.1752
	TSCO	9257.07	9261.32	9258.09	–
	CSA	9257.07	9327.72	9259.165	9.886
Case VI.ii	KKO	12113.4	12419.3	10798.44	942.7646
	IGA-NCM	12117.17	12117.1725	12117.1704	0.006
Case VI.iii	KKO	10045.4	10063.5	10049.62	9.640211
	IGA-NCM	10107.9071	10108.6241	10108.04	0.1804
	EMA	10111.0732	–	10111.0932	–
	GWO	10111.24	10452.12	10194.91	–
Case VI.iv	KKO	57735.6	60646.6	58491.46	1259.745
	IGA-NCM	57826.0802	57875.3621	57843.5046	10.4849
	GSO	58122.7	–	–	–
	IGSO	58049.019	58219.141	58156.519	–
	MPHS	57845.639	–	–	–
	RCGA-IMM	57927.69	58301.901	58066.635	–
Case VI.V	KKO	115422	117172	116332	797.3646
	IGA-NCM	115685.1807	115828.4495	115767.6079	30.8627
	GSO	117098.4186	117109.97	117103.02	–
	OGSO	116678.1987	116695.7408	116684.819	–

Table 6
Comparison analysis for test system V.i.

Method	G_1	G_2	G_3	C	F_C	F_E	L_P
FA (Gherbi et al., 2016)	128.8249	192.5856	190.2825	39209.93	–	311.15	11.6936
BA (Gherbi et al., 2016)	128.8280	192.5792	190.2858	39209.94	–	311.15	11.6936
HYB (Gherbi et al., 2016)	128.8343	192.5670	190.2918	39209.96	-	311.15	11.6936
GA (Gherbi et al., 2016)	128.997	192.683	190.11	39220	-	311.27	11.6964
PSO (Elattar, 2019)	128.984	192.645	190.063	39210.20	–	311.150	11.6919
FPA (Abdelaziz et al., 2016b)	128.8074	192.5906	190.2958	39210.15	–	311.155	11.6938
MSFLA (Elattar, 2019)	128.338	191.964	191.389	39209.81	–	311.1638	11.6927
KKO	129.011	192.303	190.274	**39199.7**	**25490.5**	**311.013**	**11.6874**

9. KKO to solve combined heat and power dispatch problem

9.1. Introduction

In past few decades, combined heat and power units have drawn attention of researchers. These units are quite popular in the field of energy saving and environmental protection. This is because the efficiency of these units are high which can be up to 90% (Alipour et al., 2014; Haghrah et al., 2016). They are capable enough to reduce the emission rate which can vary between 13%–18% (Vasebi et al., 2007). Unlike a economic load dispatch problem or a CEED problem the objective while solving a combined heat and power dispatch (CHPED) problem is same i.e. operate the units to fulfill the required demand

Table 7
Comparison analysis for test system V.ii.

Outputs	NSGA-II (Basu, 2011)	PDE (Basu, 2011)	SPEA-2 (Basu, 2011)	GSA (Güvenç et al., 2012)	ABC-PSO (Manteaw and Odero, 2012)	EMOCA (Zhang et al., 2013)	FPA (Abdelaziz et al., 2016a)	LFA (Kheshti et al., 2017)	KKO
G_1	51.9515	54.9853	52.9761	54.9992	55	55	53.188	54.9920	54.9923
G_2	67.2584	79.3803	72.813	79.9586	80	80	79.975	78.7689	78.8914
G_3	73.6879	83.9842	78.1128	79.4341	81.14	83.5594	78.105	87.7168	78.7946
G_4	91.3554	86.5942	83.6088	85	84.213	84.6031	97.119	78.1055	88.7479
G_5	134.0522	144.4386	137.2432	142.1063	138.3377	146.5632	152.74	140.6272	159.814
G_6	174.9504	165.7756	172.9188	166.5670	167.5086	169.2481	163.08	157.0936	160.555
G_7	289.4350	283.2122	287.2023	292.8749	296.8338	300	258.61	299.9954	262.174
G_8	314.0556	312.7709	326.4023	313.2387	311.5824	317.3496	302.22	309.2219	308.857
G_9	455.6978	440.1135	448.8814	441.1775	420.3363	412.9183	433.21	439.32434	430.307
G_{10}	431.8054	432.6783	423.9025	428.6306	449.1598	434.3133	466.07	438.6947	461.039
F_C	1.13539	1.1351	1.1352	1.1349	1.1342	1.13445	1.1337	**1.13246**	1.13481
F_E	4130.2	4111.4	4109.1	4111.4	4120.1	4113.98	3997.7	4139.89	**3982.85**
L_P	84.25	83.9	84.1	83.9869	84.1736	**83.56**	84.3	84.37	84.17

Table 8
Comparison of C_{Fuel} for test system V.iii.

Units	MHBA (Liang et al., 2018)	FSBF (Liang et al., 2018)	NSBF (Liang et al., 2018)	KKO
G_1	10.94	19.43	17.80	0.129546
G_2	29.85	37.26	33.66	0.322445
G_3	58.29	68.57	72.92	0.541935
G_4	99.48	59.19	59.08	0.969029
G_5	51.81	60.85	57.66	0.529524
G_6	36.20	40.61	44.74	0.36548
F_C	607.39	619.3679	619.6086	**605.68**
F_E	0.2208	**0.2015**	0.2027	0.217897
L_P	3.204	2.51	2.46	**0.02396**

Table 9
Comparison of $C_{Emission}$ for test system V.iii.

Units	MHBA (Liang et al., 2018)	FSBF (Liang et al., 2018)	NSBF (Liang et al., 2018)	KKO
G_1	40.94	41.19	40.47	0.412812
G_2	45.15	46.62	45.33	0.461706
G_3	53.30	54.21	54.39	0.545594
G_4	40.51	38.48	39.21	0.386772
G_5	54.25	54.31	54.54	0.545754
G_6	52.14	51.60	52.46	0.516297
F_C	**643.3760**	645.6193	644.4141	646.457
F_E	0.1942	0.1942	0.1942	**0.194185**
L_P	**2.92**	3.01	3	0.03493

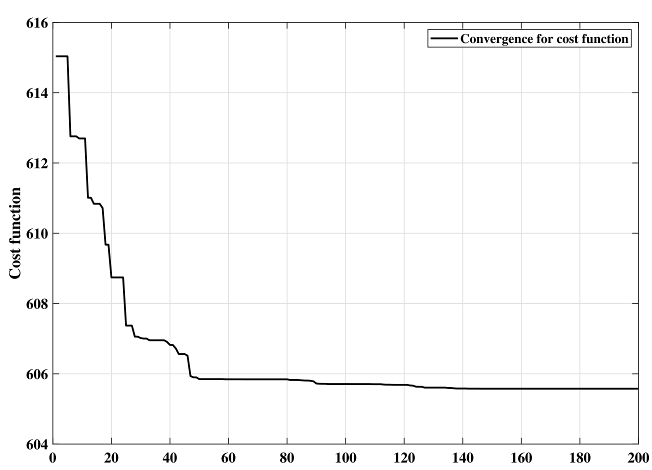

Fig. 12. Convergence profile of C_{Fuel} for test system V.iii.

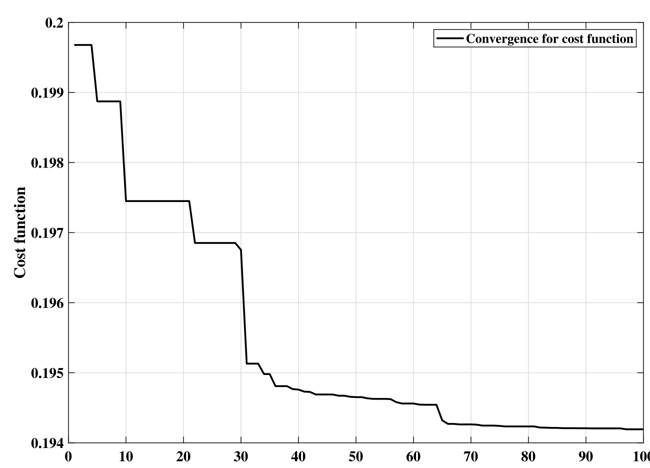

Fig. 13. Convergence profile of $C_{Emission}$ for test system V.iii.

and at the same time all constraints should be met. But the complexity for these system is high as more constraints are to be handled.

Considering the complexity and non-linearity of a CHPED problem, many techniques have been proposed to solve this problem more effectively and obtain more optimized results. Some of the techniques include improved genetic algorithm with new constraints handling strategy (IGA-NCM) (Zou et al., 2019), group search optimization (GSO) (Davoodi et al., 2017), hybrid bat and artificial bee colony (Murugan et al., 2018), hybrid gravitational search algorithm and particle swarm optimization (Beigvand et al., 2017), integrated civilized swarm

Table 10
Comparison analysis for test system V.iv.

	NSGA-II (Basu, 2011)	SPEA-2 (Basu, 2011)	GSA (Güvenç et al., 2012)	MABC/D/C (Secui, 2015)	MABC/D/Log (Secui, 2015)	FPA (Abdelaziz et al., 2016b)	ISA (Karthik et al., 2019)	KKO
G_1	113.8685	113.9694	113.9989	110.7998	110.7998	43.405	43.567	114
G_2	113.6381	114	113.9896	110.7998	110.7998	113.95	113.56	113.045
G_3	120	119.8719	119.9995	97.3999	97.3999	105.86	105.76	119.744
G_4	180.7887	179.9284	179.7857	174.5504	174.5486	169.65	169.43	181.102
G_5	97	97	97	87.7999	97	96.659	96.62	96.5081
G_6	140	139.2721	139.0128	105.3999	105.3999	139.02	139.23	139.796
G_7	300	300	299.9885	259.5996	259.5996	273.28	273.36	299.686
G_8	299.0084	298.2706	300	284.5996	284.5996	285.17	285.15	298.619
G_9	288.8890	290.5228	296.2025	284.5996	284.5996	241.96	241.54	289.447
G_{10}	131.6132	131.4832	130.3850	130	130	131.26	131.26	131.386
G_{11}	246.5128	244.6704	245.4775	318.1921	318.2129	312.13	312.12	247.114
G_{12}	318.8748	317.2003	318.2101	243.5996	243.5996	362.58	362.45	318.381
G_{13}	395.7224	394.7357	394.6257	394.2793	394.2793	346.24	346.34	395.689
G_{14}	394.1369	394.6223	395.2016	394.2793	394.2793	306.06	306.06	393.82
G_{15}	305.5781	304.7271	306.0014	394.2793	394.2793	358.78	358.54	305.891
G_{16}	394.6968	394.7289	395.1005	394.2793	394.2793	260.68	260.23	394.283
G_{17}	489.4234	487.9857	489.2569	399.5195	399.5195	415.19	415.26	489.706
G_{18}	488.2701	488.5321	488.7598	399.5195	399.5195	423.94	423.56	487.897
G_{19}	500.8	501.1683	499.2320	506.1985	506.1716	549.12	549.03	500.104
G_{20}	455.2006	456.4324	455.2821	506.1985	506.2206	496.7	496.74	455.719
G_{21}	434.6639	434.7887	433.4520	514.1472	514.1105	539.17	538.76	434.334
G_{22}	434.15	434.3937	433.8125	514.1455	514.1472	546.46	546.46	434.86
G_{23}	445.8385	445.0772	445.5136	514.5237	514.5664	540.06	540.56	446.6
G_{24}	450.7509	451.8970	452.0547	514.5386	514.4868	514.5	514.55	451
G_{25}	491.2745	492.3946	492.8864	433.5196	433.5195	453.46	453.67	491.259
G_{26}	436.3418	436.9926	433.3695	433.5195	433.5196	517.31	516.891	435.771
G_{27}	11.2457	10.7784	10.0026	10	10	14.881	14.345	11.079
G_{28}	10	10.2955	10.0246	10	10	18.79	18.64	10.3466
G_{29}	12.0714	13.7018	10.0125	10	10	26.611	26.578	12.2337
G_{30}	97	96.2431	96.9125	97	87.8042	59.581	59.565	96.6001
G_{31}	189.4826	190.0000	189.9689	159.733	159.733	183.48	183.36	189.436
G_{32}	174.7971	174.2163	175	159.733	159.7331	183.39	182.87	175.188
G_{33}	189.2845	190	189.0181	159.733	159.733	189.02	189.22	189.992
G_{34}	200	200	200	200	200	198.73	198.65	199.679
G_{35}	199.9138	200	200	200	200	198.77	198.76	199.89
G_{36}	199.5066	200	199.9978	200	200	182.23	182.45	199.905
G_{37}	108.3061	110	109.9969	89.1141	89.1141	39.673	39.635	108.554
G_{38}	110	109.6912	109.0126	89.1141	89.1141	81.596	81.525	109.71
G_{39}	109.7899	108.5560	109.4560	89.1141	89.1141	42.96	42.91	108.639
G_{40}	421.5609	421.8521	421.9987	506.1879	506.1951	537.17	537.15	421.912
F_C	1.2583	1.2581	1.2578	1.24490903	1.24491161	1.23170	**1.23034**	1.25852
F_E	2.1095	2.1110	2.1093	2.56560267	2.56560267	2.0846	**2.0643**	2.10837

Table 11
Comparison analysis for test system V.v.

	CSAISA (Rajagopalan et al., 2019)	ISA (Rajagopalan et al., 2019)	HSA (Jeddi and Vahidinasab, 2014)	DE (Rajagopalan et al., 2019)	PSO (Rajagopalan et al., 2019)	GA (Rajagopalan et al., 2019)	KKO
G_1	102.6468	100.3485	100.3839	100.5473	100.7363	100.8578	65.43
G_2	59.1816	58.8270	58.6583	58.5372	58.2314	58.3547	75.2154
G_3	50.0599	50.8309	50.8302	50.8474	50.5242	50.9943	69.7721
G_4	70.3498	73.3932	73.5292	73.0932	73.3238	73.4352	76.7522
G_5	63.1042	59.1153	59.1846	59.9323	59.7272	59.4636	78.5975
G_6	51.0080	50.1468	50.9231	50.7397	50.2726	50.6254	74.8284
G_7	50.0000	50.7470	50.2832	50.5360	50.8362	50.5363	64.9154
G_8	51.0166	53.2494	53.0220	53.2324	53.5242	53.1321	64.005
G_9	83.9497	85.1551	85.8231	85.4235	85.4355	85.3546	73.8251
G_{10}	98.1409	92.1870	92.6484	92.5426	92.0388	92.7522	66.2354
G_{11}	58.6019	61.3470	61.9233	61.5243	61.6493	61.4368	68.8125
G_{12}	119.1476	121.8597	121.4353	121.6357	121.7468	121.2463	65.6357
G_{13}	50.0000	50.0000	50.4352	50.5367	50.6484	50.7468	64.8102
G_{14}	50.0000	50.0000	50.5327	50.6382	50.8202	50.4373	50
L_P	7.2070	**7.2069**	7.3536	7.6388	7.8447	7.4357	8.854
F_C	4352.39	4353.57	4366.27	4387.44	4427.26	4453.85	**4304.62**
F_E	135.23	136.46	144.85	153.42	176.75	184.38	**108.185**

optimization with Powell pattern search method (Narang et al., 2017), cuckoo optimization algorithm (Mehdinejad et al., 2017), social cognitive optimization (SCO) (Sun and Li, 2019), SCO with tent map (Sun and Li, 2019), real coded genetic algorithm with improved Muhlenbein mutation (Haghrah et al., 2016), effective cuckoo search algorithm (ECSA) (Nguyen et al., 2018), multi-player harmony search method (MPHS) (Nazari-Heris et al., 2019).

9.2. Problem formulation

For a CHPED problem, the objective is to optimize the heat and power units such that the operational cost is minimum. It is also necessary that the heat and power demands should satisfy. Taking these objectives into consideration, the objective function for a CHPED

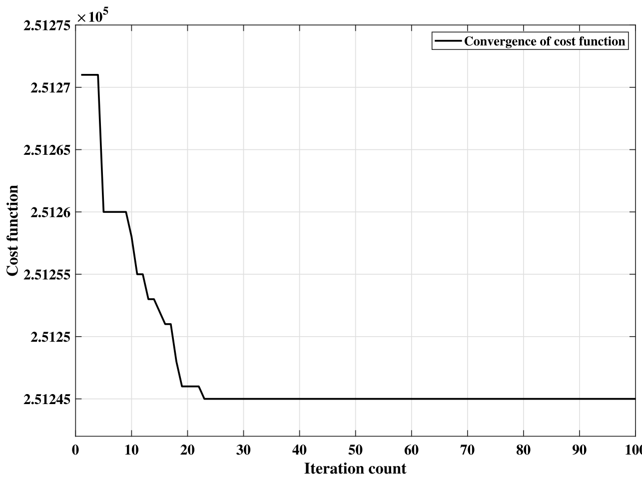

Fig. 14. Convergence profile for test system V.iv.

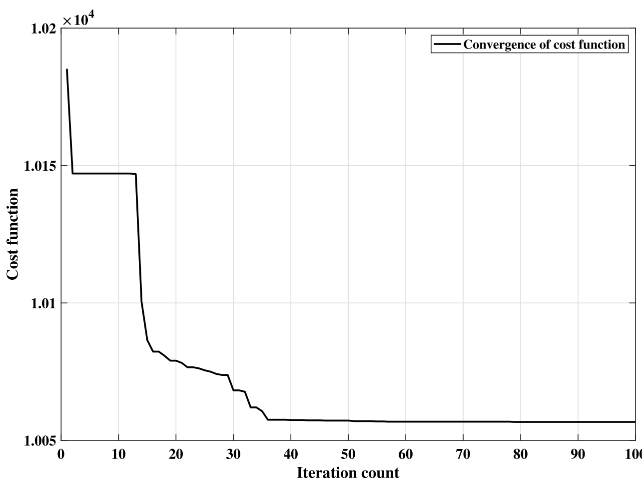

Fig. 15. Convergence profile for test system V.v.

problem is formulated as follows:

$$C = F_P + F_{(P,H)} + F_H \tag{34}$$

where F_P is the production cost for conventional thermal units; $F_{(P,H)}$ is the production cost for co-generation units and F_H is the production cost for heat units. Production cost for these three generating units are given as follows:

The production cost for the conventional thermal units are computed as:

$$F_P = \sum_{i=1}^{n} (a_i G_i^2 + b_i G_i + c_i + |g_i \times \sin(h_i \times (G_i^{\min} - G_i)|); \tag{35}$$

where G_i is the active power generated by ith generating unit; a_i, b_i and c_i are the fuel coefficients for ith generating unit; g_i and h_i are coefficients of the ith generator due to valve point effect; G_i^{min} is the minimum generation capacity of ith generating unit.

The production cost for the co-generation units are calculated as follows.

$$F_{(P,H)} = \sum_{i=1}^{m} (a_i G_i^2 + b_i G_i + c_i + d_i H_i^2 + e_i H_i + f_i H_i Pi); \tag{36}$$

where P_i and H_i are the power and heat generated by ith co-generating unit respectively; a_i, b_i, c_i, d_i, e_i and , f_i are the cost parameters of ith co-generating unit.

The production cost for heat thermal units are computed as follows.

$$F_H = \sum_{i=1}^{o} (a_i H_i^2 + b_i H_i + c_i); \tag{37}$$

where H_i is heat generated by ith heat unit; a_i, b_i, c_i, d_i, e_i and f_i are the cost parameters of ith heat unit.

9.3. Constraints

9.3.1. Capacity limits

It is important that all the conventional thermal units, co-generating units and heat units operate in a prescribed generating limits.

For ith generating unit in a conventional thermal unit, this limits is given as follows:

$$P_i^{min} \le P_i \le P_i^{max}; \tag{38}$$

where P_i^{min} is the minimum generating capacity and P_i^{max} is the maximum generating capacity of ith generating unit.

Similarly, for a ith heat unit, the prescribed generating limits is given as follows:

$$H_i^{min} \le H_i \le H_i^{max}; \tag{39}$$

where H_i^{min} is the minimum generating capacity and H_i^{max} is the maximum generating capacity of ith heat unit.

In case of a ith co-generating unit, prescribed limits is given as

$$P_i^{min}(H_i) \le P_i \le P_i^{max}(H_i); \tag{40}$$

$$H_i^{min}(P_i) \le H_i \le H_i^{max}(P_i); \tag{41}$$

where $P_i^{min}(H_i)$ and $P_i^{max}(H_i)$ are the functions of generated heat H_i and represent the maximum and minimum power limits of ith co-generating unit; $H_i^{min}(P_i)$ and $H_i^{max}(P_i)$ are the functions of generated heat P_i and represent the maximum and minimum power limits of ith co-generating unit.

9.3.2. Heat balance

It is important that the total heat produced by the heat and co-generating unit is equal to the heat demand. Mathematically, this is represented as follow:

$$\sum_{i=1}^{n} H_i + \sum_{i=1}^{n} H_i = H_D. \tag{42}$$

9.3.3. Power balance

Similar to the heat balance, it is important that the total power generated by the thermal unit and co-generating unit is equal to the total power demand. This is represented as follows:

$$\sum_{i=1}^{n} P_i + \sum_{i=1}^{n} P_i = P_D \tag{43}$$

9.4. Results and discussions

To test the effectiveness of KKO algorithm to solve CHPED problem, five test systems are considered. The description of these five test systems are presented in Table 4. All the five systems are simulated for fifteen times with an iteration count and player population of 500 and 200 respectively. The statistical analysis of these simulation is presented in Table 5.

The power and heat scheduling for the test system VI.i is presented in Table 12. A comparative result analysis with existing techniques like RCGA-IMM (Haghrah et al., 2016), EMA (Ghorbani, 2016), GWO (Jayakumar et al., 2016), MCSA (Nguyen et al., 2016), CSA (Nguyen et al., 2016), CSO (Meng et al., 2015), SCO (Sun and Li, 2019) and TSCO (Sun and Li, 2019) is presented in Table 12. The analysis shows that the proposed KKO technique schedules the units

Table 12
Comparison analysis for test system VI.i.

Method	G_1	G_2	G_3	H_2	H_3	H_4	C
RCGA-IMM (Haghrah et al., 2016)	0	160	40	40	75	0	9257.075
EMA (Ghorbani, 2016)	0	160	40	40	75	0	9257.07
GWO (Jayakumar et al., 2016)	0	160	40	40	75	0	9257.07
MCSA (Nguyen et al., 2016)	0	160	40	40	75	0	9257.07
CSA (Nguyen et al., 2016)	0	160	40	40	75	0	9257.07
CSO (Meng et al., 2015)	0	160	40	40	75	0	9257.07
SCO (Sun and Li, 2019)	0	160	40	40	75	0	9257.07
TSCO (Sun and Li, 2019)	0	160	40	40	75	0	9257.07
IGA-NCM (Zou et al., 2019)	0	160	40	40	75	0	9257.075
KKO	0.0282	155.015	44.9568	18.1301	96.8699	0	9217.03

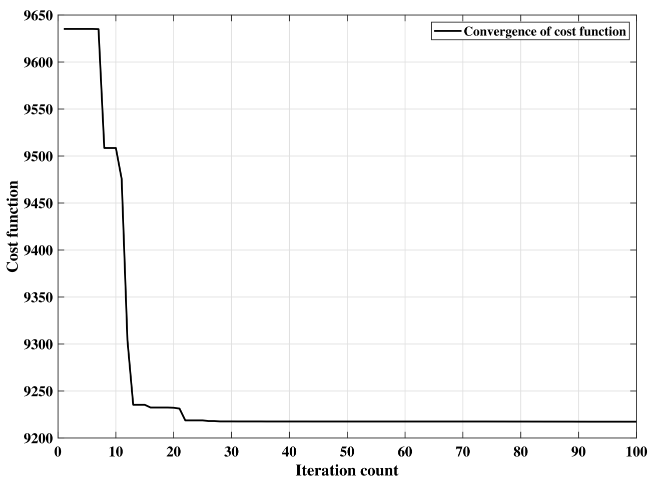

Fig. 16. Convergence profile for test system VI.i.

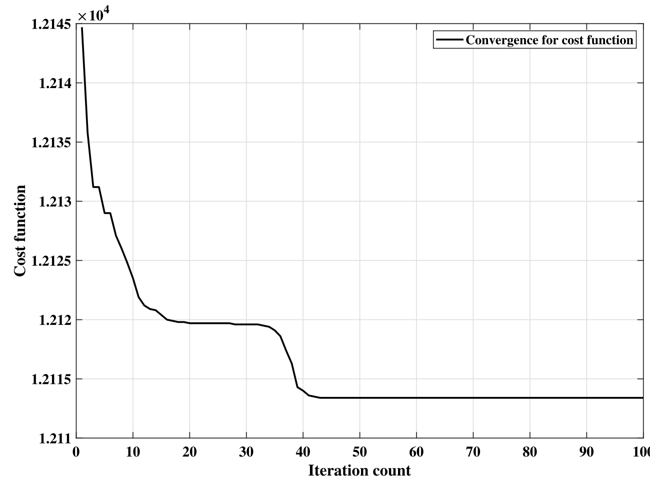

Fig. 17. Convergence profile for test system VI.ii.

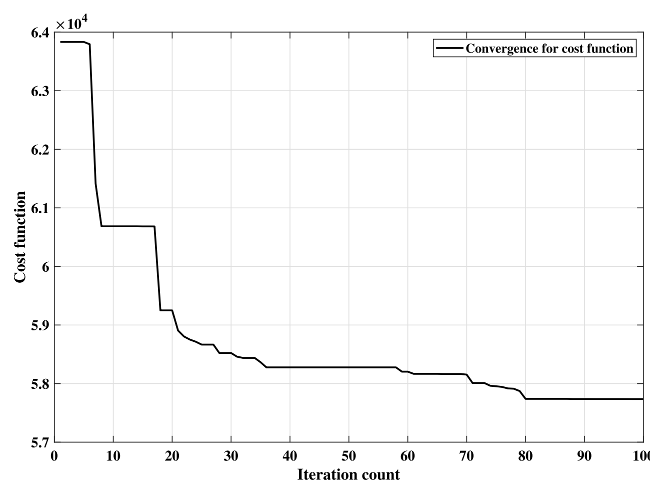

Fig. 18. Convergence profile for test system VI.iii.

more effectively and thus provides the best results a compared to existing techniques. The convergence behavior of the objective function is presented in Fig. 16.

For test system VI.ii, the result obtained using KKO is compared with existing results of RCGA-IMM (Haghrah et al., 2016), MGSO (Davoodi et al., 2017), GSO (Davoodi et al., 2017), COA (Mehdinejad et al., 2017), BD (Abdolmohammadi and Kazemi, 2013), TVAC-PSO (Mohammadi-Ivatloo et al., 2013), GSA (Beigvand et al., 2016), PFCOA (Mellal and Williams, 2015), IWO (Jayabarathi et al., 2014) and IGA-NCM (Zou et al., 2019). All the results along with power and heat scheduling are presented in Table 13. It is observed that the KKO provides the best fuel cost. The convergence profile of the objective function is presented in Fig. 17.

Table 14 presents the best result obtained for KKO and some other existing techniques like RCGA (Haghrah et al., 2016), AIS (Jordehi, 2015), ECSA (Nguyen et al., 2018), EMA (Ghorbani, 2016), TLBO (Roy et al., 2014), CSO (Meng et al., 2015), SCO (Sun and Li, 2019), TSCO (Sun and Li, 2019) for test system VI.iii. It is observed that the proposed KKO provides a fuel cost of 10045.4 which is better as compared to other techniques. This proves that KKO can also effectively solve this test system. The convergence profile for this system is presented in Fig. 18.

Best results of the proposed KKO for test system VI.iv is presented in Table 15. This table also includes the result for techniques like CPSO (Mohammadi-Ivatloo et al., 2013), TVAC-PSO (Mohammadi-Ivatloo et al., 2013), GSO (Hagh et al., 2014), IGSO (Hagh et al., 2014), OTLBO (Roy et al., 2014), GWO (Jayakumar et al., 2016), RCGA-IMM (Haghrah et al., 2016) and MPHS (Nazari-Heris et al., 2019) which are used for comparison. It is observed that the proposed algorithm optimizes the generating units more effectively as the fuel cost obtained using the proposed KKO is less as compared to other techniques. The convergence behavior of the objective function for this test system is presented in Fig. 19.

For the last test system VI.v considered to test effectiveness of the proposed KKO algorithm, the optimal result obtained are presented in Table 16. The optimal results obtained using existing techniques like OGSO (Basu, 2015a), MPSO (Basu, 2015b), GSO (Basu, 2016) and IGA-NCM (Zou et al., 2019) are also presented in this table. From the comparison, it is clear that the proposed technique provides the best result as the fuel cost obtained using the KKO algorithm is minimum. The convergence behavior of the objective function is presented in Fig. 20.

Table 13
Comparison analysis for test system VI.ii.

Method	G_1	G_2	G_3	G_4	H_2	H_3	H_4	H_5	C
RCGA-IMM (Haghrah et al., 2016)	135.0000	40.0000	10.0000	65.0000	75.0000	40.0000	14.0595	45.9405	12104.868
MGSO (Davoodi et al., 2017)	135	39.9999	9.9996	64.9003	75.0008	40.0002	14.4379	45.5609	12113.6041
GSO (Davoodi et al., 2017)	134.9953	40.2832	10.0962	64.6251	71.7131	39.8592	6.1571	57.2704	12128.7805
COA (Mehdinejad et al., 2017)	135	40	10	64991	75	40	14.4001	45.6	12116.6
BD (Abdolmohammadi and Kazemi, 2013)	135	40	10	65	75	40	14.3226	45.6774	12116.601231
TVAC-PSO (Mohammadi-Ivatloo et al., 2013)	135	40.0118	10.0391	64.9491	74.8263	39.8443	16.1867	44.1428	12117.3895
GSA (Beigvand et al., 2016)	135	39.9998	10	64.9807	74.9844	40	17.8939	42.1095	12117.37
PFCOA (Mellal and Williams, 2015)	134.96	40	10	65	75	40	14.13	45.85	12115.91
IWO (Jayabarathi et al., 2014)	134.59	40	10.94	64.47	75	38.98	8.81	52.21	12134.33
IGA-NCM (Zou et al., 2019)	135	40	10	65	75	40	14.0595	45.9405	12117.1701
KKO	135	40.0087	10	64.9913	78.0441	50.9347	2.92276	43.0985	12113.4

Table 14
Comparison analysis for test system VI.iii.

Method	RCGA (Haghrah et al., 2016)	AIS (Jordehi, 2015)	ECSA (Nguyen et al., 2018)	EMA (Ghorbani, 2016)	TLBO (Roy et al., 2014)	CSO (Meng et al., 2015)	SCO (Sun and Li, 2019)	TSCO (Sun and Li, 2019)	GWO (Zou et al., 2019)	IGA-NCM (Zou et al., 2019)	KKO
G_1	74.6834	50.3610	53.7610	52.6847	45.266	45.4909	58.7268	45.5231	52.8074	45.155	45.4842
G_2	97.9578	95.5552	98.5039	98.5398	98.5479	98.5398	98.5398	98.5385	98.5398	98.5398	99.384
G_3	167.2308	110.7515	112.5996	112.6734	112.6786	112.6734	112.6735	112.5798	112.6735	112.6735	113.646
G_4	124.9079	208.7688	209.7993	208.8158	209.8284	209.8158	209.8158	209.8159	209.8158	209.8158	209.029
G_5	98.8008	98.8	93.0872	93.8341	94.4121	94.1838	81	94.2261	93.8115	94.5549	92.4743
G_6	44.0001	44	40.2022	40	40.0062	40	40	40	40	40	40.7171
H_5	58.0965	19.4242	33.6571	29.242	25.8365	27.1786	92.0061	28.6947	29.3704	29.2388	0
H_6	32.4116	77.0777	72.6890	75	74.9970	75	45.0590	74.9981	75	75	97.1926
H_7	59.4919	53.4981	43.6539	45.7579	49.1666	47.8214	12.9349	46.3072	29.3704	45.7612	52.8074
C	10667	10355	10121.9466	10111.0732	10094.8384	10094.1267	10226.8556	10094.2351	10111.24	10107.9071	10045.4
L_P	–	–	–	–	–	–	–	–	0.76499	–	0.7346

Table 15
Comparison analysis for test system VI.iv.

Output	CPSO (Mohammadi-Ivatloo et al., 2013)	TVAC-PSO (Mohammadi-Ivatloo et al., 2013)	GSO (Hagh et al., 2014)	IGSO (Hagh et al., 2014)	OTLBO (Roy et al., 2014)	GWO (Jayakumar et al., 2016)	RCGA-IMM (Haghrah et al., 2016)	MPHS (Nazari-Heris et al., 2019)	IGA-NCM (Zou et al., 2019)	KKO
G_1	680	538.5587	627.7455	628.152	538.5656	538.8440	448.8000	628.3185	628.3188	575.764
G_2	0	224.4608	76.2285	299.4778	299.2123	299.3423	299.9568	299.1993	299.1982	356.429
G_3	0	224.4608	299.5794	154.5535	299.1220	299.3423	299.2108	299.1993	299.1665	49.4861
G_4	180	109.8666	159.4386	60.846	109.9920	109.9653	109.8694	109.8665	109.8673	69.8744
G_5	180	109.8666	61.2378	103.8538	109.9545	109.9653	109.8679	60	109.8662	100.285
G_6	180	109.8666	60	110.0552	110.4042	109.9653	159.7353	109.8665	60	115.712
G_7	180	109.8666	157.1503	159.0773	109.8045	109.9653	109.8684	109.8665	109.8608	140.679
G_8	180	109.8666	107.2654	109.8258	109.6862	109.9653	60.6545	60	109.8237	128.701
G_9	180	109.8666	110.1816	159.992	109.8992	109.9653	159.7354	159.7331	109.8523	124.517
G_{10}	50.5304	77.5210	113.9894	41.103	77.3992	77.6223	75.8146	40	40.0001	59.0988
G_{11}	50.5304	77.5210	79.7755	77.7055	77.8364	77.6223	40.1672	40	77.0316	82.4809
G_{12}	55	120	91.1668	94.9768	55.2225	55.0000	92.6079	55	55.0098	82.6289
G_{13}	55	120	115.6511	55.7143	55.0861	55.0000	92.4056	91.9038	55	94.9095
G_{14}	117.4854	88.3514	84.3133	83.9536	81.7524	83.4650	83.0376	81.0006	81.0035	83.2419
G_{15}	45.9281	40.5611	40	40	41.7615	40.0000	40.0071	40.0079	40.0003	41.4101
G_{16}	117.4854	88.3514	81.1796	85.7133	82.2730	82.7732	81.4577	81.0002	81.0003	99.468
G_{17}	45.9281	40.5611	40	40	40.5599	40.0000	41.6937	40.0059	40.0001	66.5692
G_{18}	10.0013	10.0245	10	10	10.0002	10.0000	10.0042	10.0317	10.0002	10.3179
G_{19}	42.1109	40.4288	35.0970	35	31.4679	31.4568	35.1058	35	35.0003	68.4258
H_{14}	125.2754	108.9256	106.6588	106.4569	105.2219	106.0991	105.9431	104.8003	104.8013	171.612
H_{15}	80.1175	75.4844	74.9980	74.998	76.5205	75.0000	75.0059	75.0068	75.0001	114.465
H_{16}	125.2754	108.9256	104.9002	107.4073	105.5142	105.7890	105.0550	104.8	104.7995	87.5713
H_{17}	80.1174	75.484	74.9980	74.998	75.4833	75.0000	76.4619	75.0051	74.9988	131.198
H_{18}	40.0005	40.0104	40	40	39.9999	40.0000	40.0007	40.0136	39.9993	36.5691
H_{19}	23.2322	22.4676	19.7385	20	18.3944	18.3782	20.0477	20 20.0001	42.8038	
H_{20}	415.9815	458.7020	469.3368	466.2575	468.9043	469.7337	467.4871	470.3751	470.4096	312.577
H_{21}	60	60	60	60	59.9994	60.0000	59.9999	60	60	60
H_{22}	60	60	60	60	59.9999	60.0000	59.9997	60	60	57.2295
H_{23}	120	120	119.6511	120	119.9854	120.0000	119.9991	119.9994	120	115.974
H_{24}	120	120	119.7176	119.8823	119.9768	120.0000	119.9998	119.9995	119.9913	119.999
C	59736.26 35	58122.7460	58225.74 50	58049.01 97	57856.2676	57846.84	57927.6919	57845.639	57826.0902	57735.9

10. Conclusion

In this article, the authors have proposed a new population-based meta-heuristic optimization algorithm, i.e. Kho-Kho optimization algorithm. The proposed technique is inspired by a well-known game played in India known as Kho-Kho. This technique mimics the strategies and behavior of players of a chasing team while they chase and touch the runner of a opposition team. The proposed algorithm is tested with twenty-nine benchmark functions, five test system for combined

Table 16
Comparison analysis for test system VI.v.

Units	OGSO (Basu, 2015a)	MPSO (Basu, 2015b)	GSO (Basu, 2016)	IGANCM (Zou et al., 2019)	KKO	Units	OGSO (Basu, 2015a)	MPSO (Basu, 2015b)	GSO (Basu, 2016)	IGANCM (Zou et al., 2019)	KKO
P_1	628.291	179.7773	179.8745	538.5723	371.36	P_{31}	10.0213	10	10.1191	10.15	10.7027
P_2	224.8131	299.9621	360	298.7715	191.207	P_{32}	45.0197	35.7983	35.1879	35.0002	35.1253
P_3	359.9981	290.3748	150.7185	299.4666	256.636	P_{33}	81.033	82.5995	96.8952	81.054	81.8093
P_4	160.311	179.5655	60	109.8936	144.536	P_{34}	47.3353	44.2751	44.8817	40.0192	40.6326
P_5	159.8826	160.0408	60.0648	159.7952	164.287	P_{35}	87.6544	91.0334	86.3425	81.0563	81.1798
P_6	160.6078	119.7704	159.8784	110.2276	169.152	P_{36}	56.2196	45.0766	44.867	40.3107	40.6314
P_7	159.98	162.9561	160.3713	159.7388	114.192	P_{37}	10.265	10.0002	10.0624	10.0033	10
P_8	160.6525	159.7361	177.5771	109.8456	105.963	P_{38}	35.5826	35.1411	35.1607	35.0039	35.276
P_9	110.0147	119.8672	120.0893	110.0311	153.815	H_{27}	115.5514	105.6169	112.5046	104.8189	28.923
P_{10}	40.0104	114.8494	115.373	40.2712	103.196	H_{28}	83.6316	78.8068	78.4728	75.0113	58.9431
P_{11}	40.0044	114.9528	114.9535	40.1973	76.0931	H_{29}	105.9949	113.8706	112.6499	105.013	143.437
P_{12}	119.9535	90.8763	94.5954	55.0122	67.734	H_{30}	79.3856	79.0298	79.0427	75.386	129.366
P_{13}	55.1371	92.7706	55.188	55.0817	69.0932	H_{31}	40.0003	40.0004	40.02	40.0493	43.135
P_{14}	0.0464	179.556	628.9382	538.5544	329.117	H_{32}	24.5077	20.3628	20.0605	19.9957	34.3535
P_{15}	151.8432	226.5089	360	299.234	296.511	H_{33}	104.7314	105.6984	113.5695	104.6892	137.125
P_{16}	151.3518	299.2174	299.4804	299.485	299.199	H_{34}	81.3557	8.7186	79.1803	75.0147	135.491
P_{17}	160.1601	159.9777	122.0289	159.8926	137.854	H_{35}	108.4749	110.4315	107.6768	104.8247	149.79
P_{18}	159.763	109.8667	110.0491	109.9319	122.598	H_{36}	89.033	79.4109	79.2272	75.2599	97.1279
P_{19}	159.9246	159.7483	60	110.0782	137.979	H_{37}	40.0628	40.0004	40.0098	39.9925	15.8136
P_{20}	160.3244	170.6698	87.0872	109.7738	178.525	H_{38}	20.2503	20.0641	19.9447	19.9972	41.3296
P_{21}	159.7561	160.6908	159.9218	109.7706	81.5462	H_{39}	448.2892	522.1121	435.2939	469.5234	399.09
P_{22}	159.8332	109.9663	60.0079	109.7815	151.841	H_{40}	60	60	59.9635	60	59.8181
P_{23}	114.8032	114.837	120	40.0047	53.8394	H_{41}	60	59.9999	59.993	60	58.9775
P_{24}	115.8473	116.0113	40	40	101.547	H_{42}	119.9937	120	119.995	119.8133	119.88
P_{25}	55.0071	92.493	108.1572	56.0734	74.8033	H_{43}	119.9991	119.9996	119.8985	119.8159	119.29
P_{26}	119.967	92.4025	93.5594	55	102.842	H_{44}	438.8062	385.8784	462.5644	471.182	370.001
P_{27}	100.2424	82.4557	94.7653	81.043	124.523	H_{45}	59.998	59.9997	59.9983	59.9974	59.1116
P_{28}	50.009	44.3773	44.0478	40.0145	47.5859	H_{46}	60	60	59.9987	59.9938	59.8754
P_{29}	83.2757	97.1621	95.0884	81.4082	96.5791	H_{47}	119.9749	119.9995	120	119.7899	119.886
P_{30}	45.0583	44.6356	44.6682	40.4519	40.4875	H_{48}	119.9591	119.9996	119.9359	119.8319	119.236
–	–	–	–	–	–	C	116678.1987	116918.9761	117098.4186	115685.1807	115422

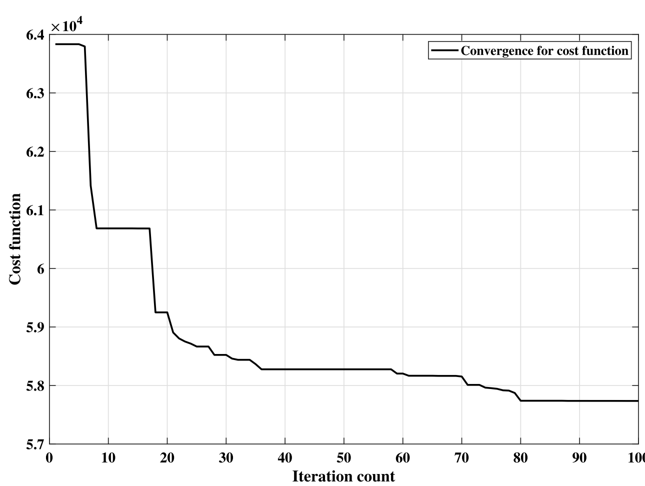

Fig. 19. Convergence profile for test system VI.iv.

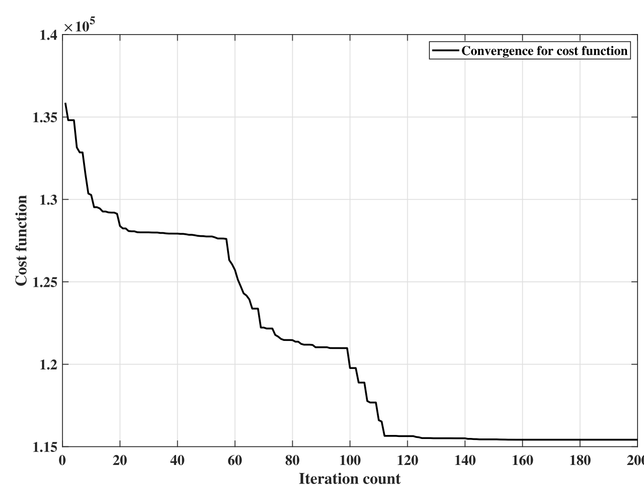

Fig. 20. Convergence profile for test system VI.v.

emission economic dispatch problem and five test systems for combined heat and power economic dispatch problem. The result obtained concludes that

- KKO exhibits good exploration capabilities.
- KKO exhibits good exploitation capabilities.
- KKO exhibits good balance between exploration and exploitation capabilities.
- KKO has a better convergence rate compared to some other existing well-known optimization techniques.
- KKO can effectively solve real-time optimization technique such as CEED and CHPED problem.

Further, the proposed KKO technique can also be used to solve other large and complex optimization problems related to other fields of engineering, mathematics, etc. In future, hybrid techniques using proposed KKO technique and other optimization techniques will be developed and tested to further improve the stability and convergence speed of the proposed the KKO technique. This could help to solve more complex problems with less computational time.

CRediT authorship contribution statement

Abhishek Srivastava: Conceptualization, Methodology, Writing - original draft . **Dushmanta Kumar Das:** Conceptualization, Methodology, Supervision, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Abdelaziz, A., Ali, E., Elazim, S.A., 2016a. Combined economic and emission dispatch solution using flower pollination algorithm. Int. J. Electr. Power Energy Syst. 80, 264–274.

Abdelaziz, A., Ali, E., Elazim, S.A., 2016b. Implementation of flower pollination algorithm for solving economic load dispatch and combined economic emission dispatch problems in power systems. Energy 101, 506–518.

Abdolmohammadi, H.R., Kazemi, A., 2013. A benders decomposition approach for a combined heat and power economic dispatch. Energy Convers. Manage. 71, 21–31.

Alipour, M., Mohammadi-Ivatloo, B., Zare, K., 2014. Stochastic risk-constrained short-term scheduling of industrial cogeneration systems in the presence of demand response programs. Appl. Energy 136, 393–404.

Alomoush, M.I., Oweis, Z.B., 2018. Environmental-economic dispatch using stochastic fractal search algorithm. Int. Trans. Electr. Energy Syst. 28 (5), e2530.

Amiri, M., Khanmohammadi, S., Badamchizadeh, M., 2018. Floating search space: A new idea for efficient solving the economic and emission dispatch problem. Energy 158, 564–579.

Banzhaf, W., Nordin, P., Keller, R.E., Francone, F.D., 1998. Genetic Programming. Springer.

Basu, M., 2011. Economic environmental dispatch using multi-objective differential evolution. Appl. Soft Comput. 11 (2), 2845–2853.

Basu, M., 2015a. Combined heat and power economic dispatch using opposition-based group search optimization. Int. J. Electr. Power Energy Syst. 73, 819–829.

Basu, M., 2015b. Modified particle swarm optimization for non-smooth non-convex combined heat and power economic dispatch. Electric Power Compon. Syst. 43 (19), 2146–2155.

Basu, M., 2016. Group search optimization for combined heat and power economic dispatch. Int. J. Electr. Power Energy Syst. 78, 138–147.

Beigvand, S.D., Abdi, H., La Scala, M., 2016. Combined heat and power economic dispatch problem using gravitational search algorithm. Electr. Power Syst. Res. 133, 160–172.

Beigvand, S.D., Abdi, H., La Scala, M., 2017. Hybrid gravitational search algorithm-particle swarm optimization with time varying acceleration coefficients for large scale CHPED problem. Energy 126, 841–853.

Bellman, R., 2013. Dynamic Programming. Courier Corporation.

Bertsekas, D.P., 1999. Nonlinear Programming. Athena scientific Belmont.

Bonabeau, E., Dorigo, M., Marco, D.d.R.D.F., Theraulaz, G., Théraulaz, G., et al., 1999. Swarm Intelligence: From Natural to Artificial Systems, no. 1. Oxford university press.

Bora, T.C., Mariani, V.C., dos Santos Coelho, L., 2019. Multi-objective optimization of the environmental-economic dispatch with reinforcement learning based on non-dominated sorting genetic algorithm. Appl. Therm. Eng. 146, 688–700.

Cuevas, E., Cienfuegos, M., ZaldíVar, D., Pérez-Cisneros, M., 2013. A swarm optimization algorithm inspired in the behavior of the social-spider. Expert Syst. Appl. 40 (16), 6374–6384.

Cuevas, E., Cortés, M.A.D., Navarro, D.A.O., 2016. Optimization based on the behavior of locust swarms. In: Advances of Evolutionary Computation: Methods and Operators. Springer, pp. 101–120.

Das, P., Das, D.K., Dey, S., 2018a. A modified bee colony optimization (MBCO) and its hybridization with k-means for an application to data clustering. Appl. Soft Comput. 70, 590–603.

Das, P., Das, D.K., Dey, S., 2018b. A new class topper optimization algorithm with an application to data clustering. IEEE Trans. Emerg. Top. Comput..

Davoodi, E., Zare, K., Babaei, E., 2017. A GSO-based algorithm for combined heat and power dispatch problem with modified scrounger and ranger operators. Appl. Therm. Eng. 120, 36–48.

Deb, S., Fong, S., Tian, Z., 2015. Elephant search algorithm for optimization problems. In: Digital Information Management. ICDIM, 2015 Tenth International Conference on. IEEE, pp. 249–255.

Dey, B., Roy, S.K., Bhattacharyya, B., 2019. Solving multi-objective economic emission dispatch of a renewable integrated microgrid using latest bio-inspired algorithms. Eng. Sci. Technol. 22 (1), 55–66.

Eberhart, R.C., Shi, Y., Kennedy, J., 2001. Swarm Intelligence (Morgan Kaufmann Series in Evolutionary Computation). Morgan Kaufmann Publishers.

Eiben, A.E., Smith, J.E., et al., 2003. Introduction to Evolutionary Computing, Vol. 53. Springer.

Elattar, E.E., 2019. Environmental economic dispatch with heat optimization in the presence of renewable energy based on modified shuffle frog leaping algorithm. Energy 171, 256–269.

Elsakaan, A.A., El-Sehiemy, R.A., Kaddah, S.S., Elsaid, M.I., 2018. An enhanced moth-flame optimizer for solving non-smooth economic dispatch problems with emissions. Energy 157, 1063–1078.

Frank, M., Wolfe, P., 1956. An algorithm for quadratic programming. Nav. Res. Logist. 3 (1–2), 95–110.

Gandomi, A.H., Alavi, A.H., 2012. Krill herd: a new bio-inspired optimization algorithm. Commun. Nonlinear Sci. Numer. Simul. 17 (12), 4831–4845.

García, S., Molina, D., Lozano, M., Herrera, F., 2009. A study on the use of non-parametric tests for analyzing the evolutionary algorithms behaviour: a case study on the CEC 2005 special session on real parameter optimization. J. Heuristics 15 (6), 617.

Gherbi, Y.A., Bouzeboudja, H., Gherbi, F.Z., 2016. The combined economic environmental dispatch using new hybrid metaheuristic. Energy 115, 468–477.

Ghorbani, N., 2016. Combined heat and power economic dispatch using exchange market algorithm. Int. J. Electr. Power Energy Syst. 82, 58–66.

Gonidakis, D., Vlachos, A., 2019. A new sine cosine algorithm for economic and emission dispatch problems with price penalty factors. J. Inf. Optim. Sci. 40 (3), 679–697.

Güvenç, U., Sönmez, Y., Duman, S., Yörükeren, N., 2012. Combined economic and emission dispatch solution using gravitational search algorithm. Sci. Iran. 19 (6), 1754–1762.

Hagh, M.T., Teimourzadeh, S., Alipour, M., Aliasghary, P., 2014. Improved group search optimization method for solving CHPED in large scale power systems. Energy Convers. Manage. 80, 446–456.

Haghrah, A., Nazari-Heris, M., Mohammadi-Ivatloo, B., 2016. Solving combined heat and power economic dispatch problem using real coded genetic algorithm with improved Mühlenbein mutation. Appl. Therm. Eng. 99, 465–475.

Jayabarathi, T., Yazdani, A., Ramesh, V., Raghunathan, T., 2014. Combined heat and power economic dispatch problem using the invasive weed optimization algorithm. Front. Energy 8 (1), 25–30.

Jayakumar, N., Subramanian, S., Ganesan, S., Elanchezhian, E., 2016. Grey wolf optimization for combined heat and power dispatch with cogeneration systems. Int. J. Electr. Power Energy Syst. 74, 252–264.

Jeddi, B., Vahidinasab, V., 2014. A modified harmony search method for environmental/economic load dispatch of real-world power systems. Energy Convers. Manage. 78, 661–675.

Jordehi, A., 2015. A chaotic artificial immune system optimisation algorithm for solving global continuous optimisation problems. Neural Comput. Appl. 26 (4), 827–833.

Joshi, P.M., Verma, H., 2019. An improved TLBO based economic dispatch of power generation through distributed energy resources considering environmental constraints. Sustain. Energy Grids Netw. 18, 100207.

Karaboga, D., Akay, B., 2009. A comparative study of artificial bee colony algorithm. Appl. Math. Comput. 214 (1), 108–132.

Karthik, N., Parvathy, A.K., Arul, R., 2019. Multi-objective economic emission dispatch using interior search algorithm. Int. Trans. Electr. Energy Syst. 29 (1), e2683.

Kennedy, J., Eberhart, R., 1995. Particle swarm optimization (PSO). In: Proc. International Conference on Neural Networks, Perth, Australia. IEEE, pp. 1942–1948.

Kheshti, M., Ding, L., Ma, S., Zhao, B., 2018. Double weighted particle swarm optimization to non-convex wind penetrated emission/economic dispatch and multiple fuel option systems. Renew. Energy 125, 1021–1037.

Kheshti, M., Kang, X., Li, J., Regulski, P., Terzija, V., 2017. Lightning flash algorithm for solving non-convex combined emission economic dispatch with generator constraints. IET Gener. Transm. Distrib. 12 (1), 104–116.

Liang, H., Liu, Y., Li, F., Shen, Y., 2018. A multiobjective hybrid bat algorithm for combined economic/emission dispatch. Int. J. Electr. Power Energy Syst. 101, 103–115.

Mahdavi, M., Fesanghary, M., Damangir, E., 2007. An improved harmony search algorithm for solving optimization problems. Appl. Math. Comput. 188 (2), 1567–1579.

Mahdi, F.P., Vasant, P., Abdullah-Al-Wadud, M., Kallimani, V., Watada, J., 2018. Quantum-behaved bat algorithm for many-objective combined economic emission dispatch problem using cubic criterion function. Neural Comput. Appl. 1–13.

Manteaw, E.D., Odero, N.A., 2012. Combined economic and emission dispatch solution using ABC_PSO hybrid algorithm with valve point loading effect. Int. J. Sci. Res. Publ. 2 (12), 1–9.

Mehdinejad, M., Mohammadi-Ivatloo, B., Dadashzadeh-Bonab, R., 2017. Energy production cost minimization in a combined heat and power generation systems using cuckoo optimization algorithm. Energy Effic. 10 (1), 81–96.

Mellal, M.A., Williams, E.J., 2015. Cuckoo optimization algorithm with penalty function for combined heat and power economic dispatch problem. Energy 93, 1711–1718.

Meng, X., Liu, Y., Gao, X., Zhang, H., 2014. A new bio-inspired algorithm: chicken swarm optimization. In: International Conference in Swarm Intelligence. Springer, pp. 86–94.

Meng, A., Mei, P., Yin, H., Peng, X., Guo, Z., 2015. Crisscross optimization algorithm for solving combined heat and power economic dispatch problem. Energy Convers. Manage. 105, 1303–1317.

Mirjalili, S., Lewis, A., 2016. The whale optimization algorithm. Adv. Eng. Softw. 95, 51–67.

Mohammadi-Ivatloo, B., Moradi-Dalvand, M., Rabiee, A., 2013. Combined heat and power economic dispatch problem solution using particle swarm optimization with time varying acceleration coefficients. Electr. Power Syst. Res. 95, 9–18.

Murugan, R., Mohan, M., Rajan, C.C.A., Sundari, P.D., Arunachalam, S., 2018. Hybridizing bat algorithm with artificial bee colony for combined heat and power economic dispatch. Appl. Soft Comput. 72, 189–217.

Narang, N., Sharma, E., Dhillon, J., 2017. Combined heat and power economic dispatch using integrated civilized swarm optimization and powell pattern search method. Appl. Soft Comput. 52, 190–202.

Nazari-Heris, M., Mohammadi-Ivatloo, B., Asadi, S., Geem, Z.W., 2019. Large-scale combined heat and power economic dispatch using a novel multi-player harmony search method. Appl. Therm. Eng. 154, 493–504.

Nguyen, T.T., Nguyen, T.T., Vo, D.N., 2018. An effective cuckoo search algorithm for large-scale combined heat and power economic dispatch problem. Neural Comput. Appl. 30 (11), 3545–3564.

Nguyen, T.T., Vo, D.N., Dinh, B.H., 2016. Cuckoo search algorithm for combined heat and power economic dispatch. Int. J. Electr. Power Energy Syst. 81, 204–214.

Rajagopalan, A., Kasinathan, P., Nagarajan, K., Ramachandaramurthy, V.K., Sengoden, V., Alavandar, S., 2019. Chaotic self-adaptive interior search algorithm to solve combined economic emission dispatch problems with security constraints. Int. Trans. Electr. Energy Syst. e12026.

Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S., 2009. GSA: a gravitational search algorithm. Inf. Sci. 179 (13), 2232–2248.

Rezaie, H., Kazemi-Rahbar, M., Vahidi, B., Rastegar, H., 2019. Solution of combined economic and emission dispatch problem using a novel chaotic improved harmony search algorithm. J. Comput. Design Eng. 6 (3), 447–467.

Rizk-Allah, R.M., El-Sehiemy, R.A., Wang, G.-G., 2018. A novel parallel hurricane optimization algorithm for secure emission/economic load dispatch solution. Appl. Soft Comput. 63, 206–222.

Roy, P.K., Paul, C., Sultana, S., 2014. Oppositional teaching learning based optimization approach for combined heat and power dispatch. Int. J. Electr. Power Energy Syst. 57, 392–403.

Sadollah, A., Bahreininejad, A., Eskandar, H., Hamdi, M., 2013. Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems. Appl. Soft Comput. 13 (5), 2592–2612.

Secui, D.C., 2015. A new modified artificial bee colony algorithm for the economic dispatch problem. Energy Convers. Manage. 89, 43–62.

Storn, R., Price, K., 1997. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. 11 (4), 341–359.

Sun, Y., Dong, W., Chen, Y., 2017. An improved routing algorithm based on ant colony optimization in wireless sensor networks. IEEE Commun. Lett..

Sun, J., Li, Y., 2019. Social cognitive optimization with tent map for combined heat and power economic dispatch. Int. Trans. Electr. Energy Syst. 29 (1), e2660.

Vasebi, A., Fesanghary, M., Bathaee, S., 2007. Combined heat and power economic dispatch by harmony search algorithm. Int. J. Electr. Power Energy Syst. 29 (10), 713–719.

Whitley, D., 1994. A genetic algorithm tutorial. Stat. Comput. 4 (2), 65–85.

Yang, X.-S., 2009. Harmony search as a metaheuristic algorithm. In: Music-Inspired Harmony Search Algorithm. Springer, pp. 1–14.

Yang, X.-S., 2010. Engineering Optimization: An Introduction with Metaheuristic Applications. John Wiley & Sons.

Yang, X.-S., 2012. Flower pollination algorithm for global optimization. In: UCNC. Springer, pp. 240–249.

Yao, X., Liu, Y., Lin, G., 1999. Evolutionary programming made faster. IEEE Trans. Evol. Comput. 3 (2), 82–102.

Yazdani, M., Jolai, F., 2016. Lion optimization algorithm (LOA): a nature-inspired metaheuristic algorithm. J. Comput. Design Eng. 3 (1), 24–36.

Zhang, R., Zhou, J., Mo, L., Ouyang, S., Liao, X., 2013. Economic environmental dispatch using an enhanced multi-objective cultural algorithm. Electr. Power Syst. Res. 99, 18–29.

Zhang, Q., Zou, D., Duan, N., Shen, X., 2019. An adaptive differential evolutionary algorithm incorporating multiple mutation strategies for the economic load dispatch problem. Appl. Soft Comput. 78, 641–669.

Zheng, Y.-J., 2015. Water wave optimization: a new nature-inspired metaheuristic. Comput. Oper. Res. 55, 1–11.

Zou, D., Li, S., Kong, X., Ouyang, H., Li, Z., 2019. Solving the combined heat and power economic dispatch problems by an improved genetic algorithm and a new constraint handling strategy. Appl. Energy 237, 646–670.