

Research Article

Multiobjective Optimization of Irreversible Thermal Engine Using Mutable Smart Bee Algorithm

M. Gorji-Bandpy and A. Mozaffari

Department of Mechanical Engineering, Babol University of Technology, P.O. Box 484, Babol, Iran

Correspondence should be addressed to A. Mozaffari, amozaffari10@yahoo.com

Received 13 July 2011; Revised 6 October 2011; Accepted 14 November 2011

Academic Editor: Chuan-Kang Ting

Copyright © 2012 M. Gorji-Bandpy and A. Mozaffari. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A new method called mutable smart bee (MSB) algorithm proposed for cooperative optimizing of the maximum power output (MPO) and minimum entropy generation (MEG) of an Atkinson cycle as a multiobjective, multi-modal mechanical problem. This method utilizes mutable smart bee instead of classical bees. The results have been checked with some of the most common optimizing algorithms like Karaboga's original artificial bee colony, bees algorithm (BA), improved particle swarm optimization (IPSO), Lukasik firefly algorithm (LFFA), and self-adaptive penalty function genetic algorithm (SAPF-GA). According to obtained results, it can be concluded that Mutable Smart Bee (MSB) is capable to maintain its historical memory for the location and quality of food sources and also a little chance of mutation is considered for this bee. These features were found as strong elements for mining data in constraint areas and the results will prove this claim.

1. Introduction

The Atkinson cycle was designed by James Atkinson in 1882 [1]. This engine has two important advantages comparing to other engines; it is one of the most heat efficient as well as high expansion ratio cycles. Generally, four procedures called Intake, Compression, Power, and Exhaust take place in cycle per turn of crankshaft. In fact a classic Atkinson engine is a four-stroke engine and, in a same condition, it can reach a higher efficiency comparing to Otto cycle.

Recently, researchers focused on analyzing and optimizing Atkinson cycle using different optimization techniques and intelligent controlling systems. Leff [2] determined the thermal efficiency of a reversible Atkinson cycle at maximum work output, Al-Sarkhi et al. [3] compared the performance characteristic curves of the Atkinson cycle to Miller and Brayton cycles using numerical examples and simulation techniques. Wang and Hou [4] studied the performance of Atkinson cycle in variable temperature heat reservoirs. Hou [5] investigated the effects of heat leak due to percentage of fuels energy, friction, and variable specific heats of working fluid. Here we proposed a new metaheuristic algorithm to analyze the performance of an air standard Atkinson cycle

with heat transfer losses, friction, and variable specific heats of the working fluid.

Metaheuristic algorithms are population-based methods working with a set of feasible solutions and trying to improve them gradually. These algorithms can be divided into two main parts: evolutionary algorithms (EAs) which attempt to simulate the phenomenon of natural evolution and swarm intelligence base algorithms [6–8]. There are many different variants of evolutionary algorithms. The common ideas behind all of these techniques are the same: defining a population of individuals, selection phase (survival of the fittest according to the theory of evolution) which causes a rise in the fitness of the population. In these methods we randomly create a set of candidate solutions (elements of the function domain) and evaluate the quality of the function through fitness measuring (the higher is better). Based on this fitness, some of the better candidates are chosen to seed the next generation by applying recombination and/or mutation to them. Recombination is an operator applied to two or more selected candidates and result in one or more new candidate. Mutation is applied to one candidate and results in one new candidate. Executing recombination and mutation leads the algorithm to a set of new candidates and this procedure

will continue until criteria have been met. Genetic algorithm (GA) which introduced by Holland [9] is one of the most popular algorithms among the EAs. Genetic algorithm (GA) is a powerful numerical optimization algorithm that reaches an approximate global maximum of a complex multivariable function over a wide search space [10]. It always produces high-quality solution because of its independency for selecting the initial configuration of population. But sometimes it may perform inefficient in constraint optimizing problems. In order to make a successful decision in constraint spaces, Tessema and Yen [11] used self adapting penalty function genetic algorithm (SAPF-GA) for optimizing constraint problems which is able to tune some of its characteristics during the optimization and made a powerful algorithm for finding feasible solution in constraint spaces [12].

Other branches of population-based algorithms which are called swarm intelligence focused on collective behavior of some self-organized systems in order to develop some metaheuristics procedures which can mimic such system's problem solution abilities. The interactive behavior between individuals locally with one another and with their environment contributes to the collective intelligence of the social colonies [13, 14] and often leads to convergence of global behavior. There is a wide variety of swarm base algorithms which mimics the natural behavior of insects and animals such as ants, fishes, birds, bees, fireflies, penguins, frogs, and many other organisms. Particle swarm optimization algorithm (PSO) which first developed by Kennedy and Eberhart [15] is one of the most applicable method for optimizing engineering problems which inspired by social behavior of birds flocking or fish schooling [16]. Till now many researchers proposed modified PSO algorithms which have advantages in handling with particular type of problems. Here one of this improved particle swarm algorithms (IPSO) which is strong in optimizing constraint engineering problems [17] is used and also its results compared with proposed modified MSB algorithm.

There are also some algorithms that improved the performance of swarm base algorithms by utilizing some natural concepts. In 2009, Yang and Deb [18] proposed a modern metaheuristic algorithm based on the obligate brood parasitic behavior of some cuckoo species in combination with the Lévy flight behavior of some birds and fruit flies which is called Cuckoo Search (CS).

One of the other improved algorithms which is used in this paper was produced in 2009 by Łukasik and Zak [19] that focused on the characteristics of fireflies and introduced an improved concept of the firefly algorithm (FA) which was strong for constraint continuous optimization tasks. Their improved method was inspired by imitating social behavior of fireflies and the phenomenon of bioluminescent communication [20].

In this paper entropy generation and power output of air standard Atkinson cycle will be analyzed in different situations as a multiobjective problem using MSB algorithm. It will be proved that different types of constraints should be considered to derive to an acceptable engineering solution. Besides, the performance of proposed algorithm will be compared to some other well-known optimization techniques

such as Karaboga's original ABC [21, 22], bees algorithm (BA) [23, 24], improved particle swarm optimization (IPSO) [17], Łukasik fire fly algorithm (LFFA), and self-adaptive penalty function genetic algorithm (SAPF-GA) [11, 12].

2. Bee Colony Optimization Strategies

Recently, many researchers focused on the interactive behavior of bees that occur through a waggle dance during the foraging process. Successful foragers share the information about the direction and the distance to patches of flower and the amount of nectar with their hive mates. Foragers can recruit other bees in their society to search in productive locations for collecting nectars with higher quality. These procedures suggest a successful data mining mechanism.

For the first time Seeley proposed a behavioral model for a colony of honey bees [25]. According to his theory, foraging bees visiting patch of flowers and then return to the hive with their collected nectars. Responding to the quality of the nectar that had been collected, waggle dance take place on the floor where each individual forager can observe the dancing process. The foragers are capable to randomly select a dance to observe and follow the dancer to the flower patch and continuing these processes will lead the colony to optimal food (solution).

Thereafter, many researchers focused on the honey bee organism and several metaheuristics were proposed based on the peculiar intelligent behavior of honey bee swarms. Yonezawa and kikuchi proposed ecological algorithm (EA) which was focused on the description of the collective intelligence based on bees' behavior [26]. Sato and Hagiwara proposed bee system (BS) which was a modified version of genetic algorithm (GA) and reach some acceptable results in optimizing engineering problems [27]. Teodorovic proposed bee colony optimization (BCO) based on forward and backward pass to generate feasible solutions during the searching procedure [28]. In 2001 Abbas [29] inspired mating bee optimization (MBO) for propositional satisfiability problems. Karaboga [30] released the first version of artificial bee colony (ABC) which is one of the most applicable algorithms in numerical optimizing field. Yang [8] concentrated on the virtual bee algorithm (VBA) due to function optimizations with the application in engineering problems. Chong et al. inspired honey bee colony (HBC) for training artificial neural network and job shop scheduling problem [31]. In 2011, Stanarevic et al. [32] introduced a modified artificial bee colony algorithm utilizing smart bees in optimizing constraint problems and demonstrated that this algorithm has better performance for optimizing constraint problems than the Karaboga's artificial bee colony (ABC). There are many other methods in optimizing application that utilized bee's behavior in nature and each one have some advantages for peculiar type of problems.

3. The Mutable Smart Bee Algorithm

Many real-world optimization problems involve inequality and equality constraints. It is hard and also takes a long

time to find a feasible solution in searching space which optimizes a constraint problem with traditional strategies. Since one of the crucial problems is to gain a feasible answer in the searching spans, different concepts proposed by researchers and a variety of methods implemented for different optimization situations [33]. Hillier [34] proposed a procedure to predict the chaotic constraints which called linear constraints. Seppälä [35] proposed a set of uniform constraints that replace a single chance constraint and he has also conclude that his method is more accurate, but less efficient than Hiller's procedure. After that, Seppälä and Orpana [36] examined the efficiency of the method which proposed by Seppälä. There are also many other methods and concepts proposed by different researchers for constraint optimization.

Recently, Stanarevic et al. [32] proposed a modified artificial bee colony algorithm (SB-ABC) based on Deb's rule [37] which is really efficient for optimizing the engineering problems that possessed different types of constraints. They improved the performance of artificial bee colony (ABC) algorithm by applying Deb's rule and also defining a penalty function in the structure of ABC algorithm. They also used smart bees in the searching space which were able to maintain their memory. Smart bees are able to compare the new candidate solution to the old one and choose the better one due to their greedy instinct. Results demonstrated that this concept is really useful for optimizing engineering problems with are often multimodal.

Here, we will analyze some features that make this algorithm really strong for optimizing multi-modal problems.

In classical ABC proposed by Karaboga and Basturk [38], the following equation was utilized to produce candidate solution in searching spans (by an employed bee or onlooker bee):

$$v_{ij} = \begin{cases} x_{ij} + \varnothing_{ij} * (x_{ij} - x_{kj}), & R_j < MR \\ x_{ij}, & \text{otherwise,} \end{cases} \quad (1)$$

where $k \in \{1, 2, \dots, SN\}$ is a randomly chosen index, x_{ij} is the variable j of the food source, x_k is a neighbor solution around i , R_j is a random number in the range (0,1), and MR is a parameter, which control the modification of parameter x_{ij} . In Karaboga's algorithm, the variable in the candidate solution which exceeds from its spans, takes value of the upper bound or lower bound regarding to its exceeding position. It is obvious that this policy may cause a local convergence.

In SB-ABC algorithm a different style was used to modify the solution:

$$\begin{aligned} & 2 * lb_j - v_{ij}, & \text{if } v_{ij} < lb_j, \\ & 2 * ub_j - v_{ij}, & \text{if } v_{ij} > ub_j, \\ & v_{ij}, & \text{otherwise,} \end{aligned} \quad (2)$$

where v_{ij} is the variable j of the candidate solution i and ub_j is the upper bound of variable j .

One of the other advantages in this method is hiring smart bees. These artificial insects can memorize the position

of the best food source and its quality which was found before and replace it to new candidate solution if the new solution is unfeasible or the new solution has a lower fitness than the best-saved solution in the SB memory.

Another important advantage of this method is the time duration for smart bee's data processing procedure. This feature will make the algorithm more durable when high amount of these artificial organisms being hired for searching the solution space. To overcome this problem, we utilized a low amount of smart bees in constraint searching space. Besides, we add a new mutation operator to SB-ABC for overcoming subsequence fast convergence. In each of the iterations, bees that exceed from a finite number of trials will be sent to a container and participate in mutation process based on their mutation probability. The results show that the global solution can be obtained faster and by adapting a dynamic mutation probability (P_m), due to the type of problem, the algorithm escape from local convergence conveniently. In the next parts, the efficient performance of proposed algorithm for optimizing a real life multimodal engineering problem will be shown more closely.

The pseudocode of MSB-ABC is given in the following:

- (1) initialize the population of solutions x_{ij} ;
- (2) evaluate the population;
- (3) cycle = 1;
- (4) repeat;
- (5) produce new solutions (food source positions) ϑ_{ij} using (1) and evaluate them;
- (6) if cycle \neq 1 use smart bee;
- (7) apply selection process based on Deb's method;
- (8) calculate the probability values P_{ij} for the solutions x_{ij} using fitness of the solutions and the constraint violations (CV) by:

$$P_i : \begin{cases} 0.5 + \left(\frac{\text{fitness}_i}{\sum_{i=1}^{SN} \text{fitness}_i} \right) * 0.5, & \text{if solution is feasible} \\ \left(1 - \frac{CV}{\sum_{i=1}^{SN} CV} \right) * 0.5, & \text{if solution is infeasible,} \end{cases} \quad (3)$$

where CV is defined by:

$$CV = \sum_{j=1}^q g_j(x) + \sum_{j=q+1}^m h_j(x), \quad (1)'$$

- (9) for each onlooker bee, produce a new solution ϑ_i by (1) in the neighborhood of the solution selected depending on P_i and evaluate it;

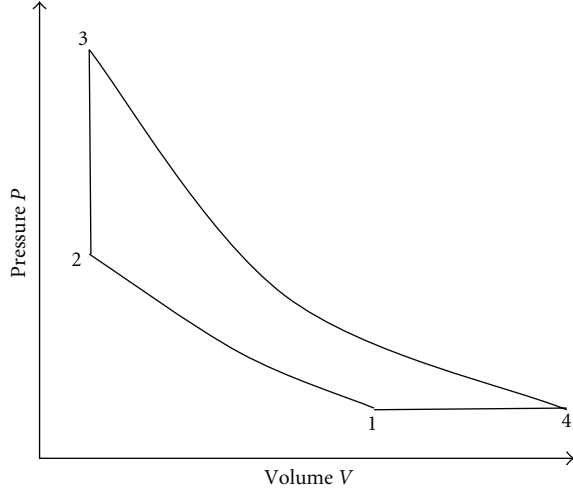


FIGURE 1: P-V diagram of the theoretical air standard Atkinson cycle.

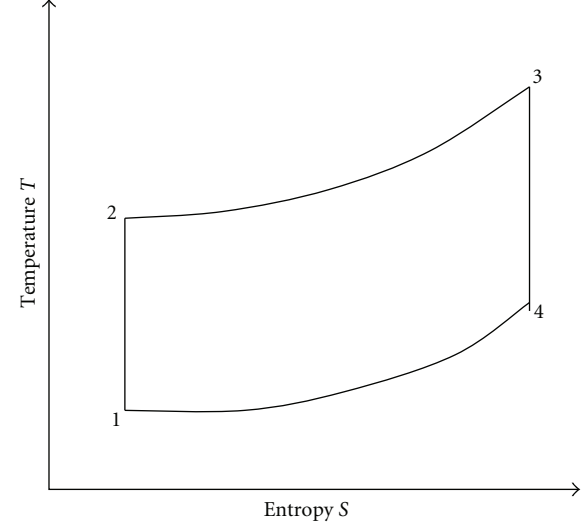


FIGURE 2: T-S diagram of theoretical air standard Atkinson cycle.

- (10) apply selection process between ϑ_i and x_i based on Deb's method;
- (11) determine the abandoned solutions (source), if exists, and perform mutation on each abandoned solution by following formula:

$$v'_j : \begin{cases} v_j + \Delta(t, ub_j - v_j), \\ \text{or} \\ v_j - \Delta(t, v_j - lb_j), \\ v_j, \end{cases} \quad \begin{matrix} \text{rand} < P_m \\ \\ \text{rand} > P_m, \end{matrix} \quad (4)$$

where t is current generation number, P_m is mutation probability and $\Delta(t, y)$ is defined by: $\Delta(t, y) = yr(1 - (t/T))^b$; $b > 1$, T is maximum cycle;

- (12) memorize the best food source position (solution) achieved so far;
- (13) cycle = cycle + 1;
- (14) until cycle = maximum cycle number.

4. Atkinson Engine

Here the performance of an air standard Atkinson cycle with heat-transfer loss, friction, and variable specific-heats of the working fluid will be analyzed precisely. According to (P-V) diagram in Figures 1 and 2, process (1-2) is an adiabatic (isentropic) compression then heat is added in process (2-3) at a constant volume. Process (3-4) is an adiabatic (isentropic) expansion, and the last process (4-1) is heat injection which takes place at constant pressure.

According to [39], assume that the specific heat ratio of the working fluid is a function of temperature, so the following linear equation can be considered:

$$\gamma = \gamma_0 - k_1 T, \quad (5)$$

where γ is the specific heat ratio and T is the absolute temperature.

It is assumed that air is an ideal gas that consists of 78.1% nitrogen, 20.95% oxygen, 0.92% argon, and 0.03% carbon dioxide.

Heat added to the working fluid in isochoric process 2 → 3 can be derived by:

$$\begin{aligned} Q_{in} &= M \int_{T_2}^{T_3} C_v dT \\ &= M \int_{T_2}^{T_3} \frac{R}{\gamma_0 - k_1 T - 1} dT \\ &= \frac{MR}{k_1} \ln \left(\frac{\gamma_0 - k_1 T_2 - 1}{\gamma_0 - k_1 T_3 - 1} \right), \end{aligned} \quad (6)$$

where M is the molar number of the working fluid, R is molar gas constant, and C_v is molar specific heat at constant volume.

Heat rejected by the working fluid in isobaric process 4 → 1 is obtained by:

$$\begin{aligned} Q_{out} &= M \int_{T_1}^{T_4} C_p dT \\ &= M \int_{T_1}^{T_4} \left(\frac{(\gamma_0 - k_1 T)R}{\gamma_0 - k_1 T - 1} \right) dT \\ &= MR \left[T_4 - T_1 + \frac{1}{k_1} \ln \left(\frac{\gamma_0 - k_1 T_1 - 1}{\gamma_0 - k_1 T_4 - 1} \right) \right], \end{aligned} \quad (7)$$

where C_p is molar specific heat at constant pressure.

According to [40, 41] the relation between parameters of a reversible adiabatic process with variable specific heat ratio can be considered by following equation:

$$TV^{\gamma-1} = (T + dT)(V + dV)^{\gamma-1}. \quad (8)$$

Respecting to (5) and (8), the following equation can be written:

$$T_i(\gamma_0 - k_1 T_j - 1) = T_j(\gamma_0 - k_1 T_i - 1) \left(\frac{V_j}{V_i} \right)^{\gamma_0 - 1}. \quad (9)$$

r_c and r_c^* were defined as specific compression ratio and compression ratio, respectively,

$$\begin{aligned} r_c &= \frac{V_1}{V_2}, \\ r_c^* &= \frac{V_4}{V_2} = \frac{T_4}{T_1} r_c, \end{aligned} \quad (10)$$

and two other processes (1-2) and (3-4) can be indicated, respectively, by the following equations:

$$T_1(\gamma_0 - k_1 T_2 - 1)(r_c)^{\gamma_0 - 1} = T_2(\gamma_0 - k_1 T_1 - 1), \quad (11)$$

$$T_3(\gamma_0 - k_1 T_4 - 1) = T_4(\gamma_0 - k_1 T_3 - 1) \left(\frac{T_4}{T_1} r_c \right)^{\gamma_0 - 1}. \quad (12)$$

By combusting an amount of energy received by working fluid that is calculated by following linear equation:

$$Q_{in} = M[A - B(T_2 + T_3)], \quad (13)$$

where A and B are two constant parameters which they relate to heat transfer and combustion that are function of engine speed. One of the other important aspects of analyzing real cycles is facing with heat leakage loss through the cylinder walls which is proportional to average temperature of the both working fluid and the cylinder wall which can be calculated by following equation [40]:

$$Q_{leak} = MB(T_2 + T_3 - 2T_0). \quad (14)$$

The power output of the Atkinson cycle engine can be derived by the following equation:

$$\begin{aligned} \dot{W}_{out} &= \frac{d}{dt}(Q_{in} - Q_{out}) \\ &= \frac{MR}{k_1} \ln \left(\frac{(\gamma_0 - k_1 T_2 - 1)(\gamma_0 - k_1 T_3 - 1)}{(\gamma_0 - k_1 T_4 - 1)(\gamma_0 - k_1 T_1 - 1)} \right) \\ &\quad + MR(T_4 - T_1), \end{aligned} \quad (15)$$

where \dot{W}_{out} represents the power output of cycle during the process.

Now the thermal efficiency of the Atkinson cycle engine can be expressed as following:

$$\eta_{th} = \frac{W_{out}}{Q_{in} - Q_{leak}}. \quad (16)$$

The amounts of r_c and T are depending on engine initial condition and can be supposed as given data. T_2 determined by (11), after that substituting (6) into (13) concludes T_3 and T_4 calculated by (12). Now these parameters can be placed into (15) and (16) for determining the output power and the thermal efficiency of the Atkinson cycle engine.

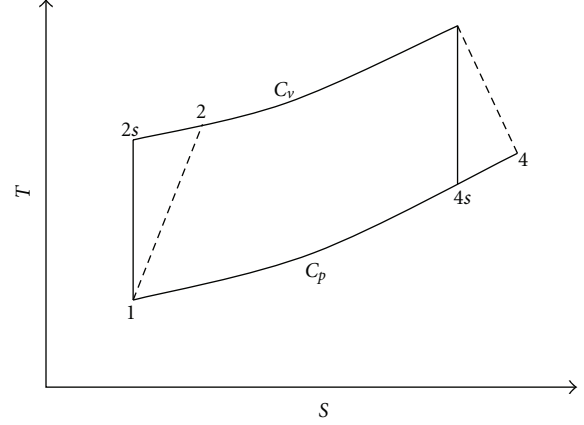


FIGURE 3: T-S diagram of real air standard Atkinson cycle.

After obtaining appropriate equations and data for calculating the power output of the Atkinson cycle, the relations between obtain parameters and entropy generation will be checked. Figure 2 does not represent the real indicated diagram of an internal combustion engine. For example, the actual cooling process between 4 and 1 cannot be compared with that of the theoretical cycle, because real engines are modeled as open systems where mass flows in and out of the system, which leads to a T-S diagram quite different from the theoretical one. Figure 3 indicates the different behavior of an ideal reversible and real irreversible Atkinson cycle.

Process 1 → 2S is an ideal reversible adiabatic compression, while process 1 → 2 is an irreversible adiabatic process with high approximation to real compression process in cycle. Heat addition in 2 → 3 is an isochoric process. Process 3 → 4S is an ideal reversible adiabatic expansion while process 3 → 4 is an irreversible adiabatic process with high approximation to real expansion process in cycle. Heat reject in 4 → 1 is an isobaric process.

As it is shown in Figure 3 in real Atkinson cycles, some amount of unexpected entropy generation must be considered. Here we consider two heat transfer units of the hot-and-cold side heat exchangers, ($N_H = N_L = 2$) due to the product of heat-transfer coefficient (α) and heat transfer surface area (F) [42].

And the effectiveness of the hot-and-cold side heat exchangers can be written as following:

$$\begin{aligned} E_H &= 1 - \exp(-N_H), \\ E_L &= 1 - \exp(-N_L). \end{aligned} \quad (17)$$

According to [42] the entropy generation rate for the Atkinson cycle is equal to:

$$\begin{aligned} \sigma &= \frac{Q_L}{T_L} - \frac{Q_H}{T_H} \\ &= m \left[\frac{C_p E_L (G - 1)}{1 - (1 - E_L)G} - \frac{C_v E_H (T_H - T_2)}{T_H + C_i (T_L/T_H) (T_H - T_2)^2} \right], \end{aligned} \quad (18)$$

where G can be written:

$$G = \left(\frac{(1 - E_H)T_2 + E_H T_H}{T_2} \right)^{C_v/C_p}. \quad (19)$$

5. Optimization Process

As it was mentioned before, we have to minimize the unexpected entropy generation and maximize the power output to obtain an efficient performance of the Atkinson cycle. In order to achieve a suitable engineering solution for optimizing the cycle under different situations, we have to face different types of constraints, and under these constraints in searching space it will be harder to find the feasible solution. In this section the efficiency of the Atkinson cycle will be checked using proposed mutable smart bee (MSB-ABC) algorithm and compared to different methods of optimizing and the results will be shown in tables at the end.

The objective functions are defined as following:

$$\begin{aligned} \dot{W}_{\text{out}} &= f(T_1, T_2, T_3, T_4, \gamma_0), \\ \sigma &= h(T_1, T_2, T_3, E_L, E_H). \end{aligned} \quad (20)$$

And signalized objective function can be considered using following approach:

$$\text{Objective function} = \sum_{i=1}^n \alpha_i f_i, \quad (21)$$

where α is the weighted coefficient and show the value of a function comparing to another objective functions and $\sum_{i=1}^n \alpha_i = 1$.

In this work, $\alpha_1 = 0.6$ for power output and $\alpha_2 = 0.4$ for entropy generation are considered for finding a suitable engineering solution.

Due to (21) the single-objective function will be derived:

$$\text{Objective} = \text{Min} \left\{ \alpha_1 \sigma + \alpha_2 \frac{1}{\dot{W}_{\text{out}}} + \sum_{j=1}^q \lambda_j g_j(x) \right\}, \quad (22)$$

where σ is the total entropy generation and \dot{W}_{out} is the power output of the Atkinson cycle, and according to (1)', $\sum_{j=1}^q g_j(x)$ shows the sum of governing constraints which represents the constraint violence (CV) and λ_j indicates the impact of each constraint. These finite numbers of

constraints have been set as following in order to lead the algorithm to a make feasible decision in the searching space:

$$\begin{aligned} C_1: T_2 > T_1 &\rightarrow g_1 = T_1 - T_{2s}, \\ C_{2,3}: T_2, T_{2s} < T_3 &\rightarrow g_2 = T_2 - T_3; g_3 = T_{2s} - T_3, \\ C_{4,5}: T_4, T_{4s} > T_1 &\rightarrow g_4 = T_1 - T_4; g_5 = T_1 - T_{4s}, \\ C_{6,7}: T_4, T_{4s} < T_2, T_{2s} &\rightarrow g_6 = T_4 - T_2; g_7 = T_{4s} - T_{2s}, \\ C_8: T_2 - T_4 > 200 &\rightarrow g_8 = 200 - T_2 + T_4, \\ C_9: T_4 - T_1 < 30 &\rightarrow g_9 = 30 + T_1 - T_4, \\ C_{10}: ||T_2 - T_1| - |T_3 - T_4|| < 200 &\rightarrow g_{10} = ||T_2 - T_1| - |T_3 - T_4|| - 200, \\ C_{11}: \left| T_1(\gamma_0 - k_1 T_2 - 1)(r_c)^{\gamma_0 - 1} - T_2(\gamma_0 - k_1 T_1 - 1) \right| &\leq \varepsilon; \varepsilon = 0.001 \\ &\rightarrow g_{11} = \left| T_1(\gamma_0 - k_1 T_2 - 1)(r_c)^{\gamma_0 - 1} \right. \\ &\quad \left. - T_2(\gamma_0 - k_1 T_1 - 1) \right| - \varepsilon, \\ C_{12}: \left| \left(\frac{\text{MR}}{k_1} \ln \left(\frac{\gamma_0 - k_1 T_2 - 1}{\gamma_0 - k_1 T_3 - 1} \right) \right) - M[A - B(T_2 + T_3)] \right| &\leq \varepsilon; \varepsilon = 0.001 \\ &\rightarrow g_{12} = \left| \left(\frac{\text{MR}}{k_1} \ln \left(\frac{\gamma_0 - k_1 T_2 - 1}{\gamma_0 - k_1 T_3 - 1} \right) \right) \right. \\ &\quad \left. - M[A - B(T_2 + T_3)] \right| - \varepsilon, \\ C_{13}: \left| T_3(\gamma_0 - k_1 T_4 - 1) - T_4(\gamma_0 - k_1 T_3 - 1) \left(\frac{T_4}{T_1} r_c \right)^{\gamma_0 - 1} \right| &\leq \varepsilon; \varepsilon = 0.001 \\ &\rightarrow g_{13} = \left| T_3(\gamma_0 - k_1 T_4 - 1) - T_4(\gamma_0 - k_1 T_3 - 1) \right. \\ &\quad \left. \times \left(\frac{T_4}{T_1} r_c \right)^{\gamma_0 - 1} \right| - \varepsilon, \\ C_{14}: \dot{W}_{\text{out}} \geq 0 &\leftrightarrow E_{14}: \\ &\text{if } \{ \exists i \mid (\dot{W}_{\text{out}})_i \in (-\infty, 0) \rightarrow \text{execute the solution}(i) \}, \\ C_{15}: \sigma \geq 0 &\leftrightarrow E_{15}: \\ &\text{if } \{ \exists i \mid (\sigma)_i \in (-\infty, 0) \rightarrow \text{execute the solution}(i) \}, \end{aligned} \quad (23)$$

where $300 < T_1 < 380$, $1000 < T_2 < 1450$, $1500 < T_3 < 2200$, and $400 < T_4 < 900$. According to [29] the following constants and ranges are set for the analyzing process:

$$T_1 = 360 \text{ k}, \quad A = 60000 \text{ J} \cdot \text{mol}^{-1}, \quad 1.31 < \gamma_0 < 1.41. \quad (24)$$

Once the constraints and the equations are obtained, the essentials for the optimizing with the mutable smart bee

TABLE 1: Performance of tested algorithm in $\gamma_0 = 1.31$ and $k_1 = 0.00006$.

	Parameters				Entropy generation	Power output	CPU time
	T_1	T_2	T_3	T_4	σ	\dot{W}	t (sec)
MSB-ABC	360	1023	2015	755.8	0.0008	0.3912	5.2
ABC	360	1030	2019	701.4	0.0015	0.2331	16.7
BA	360	1029	2081	737.0	0.0011	0.2054	18.4
IPSO	360	1036	2043	799.8	0.0012	0.2759	12.7
LFFA	360	1039	2016	789.3	0.0013	0.2690	12.3
SAPF-GA	360	1041	2090	699.2	0.0009	0.3759	22.2
Optimum performance reported in [40, 42]					0.0012	0.3112	—

algorithm are prepared. This method will find a suitable answer that is enabling to satisfy all of the constraints. Like any other evolutionary computation methods, the answer which is found by mutable smart bee algorithm is not the definite best answer; actually there are slight differences between them. These differences are usually acceptable and in engineering applications these small differences can be disregarded, Moreover in practical works these answers provide a better performance for the systems comparing to answers which are concluding from experimental works.

The difference between the algorithm answer and the real answer can be extended by finding the local optimization instead of global optimization. For avoiding this matter a suitable probability of mutation is necessary. Indeed mutation it can developed the search space for finding the answer and avoid local optimization. Although mutation is necessary to find a global optimization and seek a wide variety of answers but in latest generations can be reduce the convergence rate. Thus, as the algorithm go ahead, the mutation probability should be decreased for a better convergence in answers. A suitable mutation probability is effective on the speed of the algorithm. All the topics that were mentioned in precede will be shown later. Note that all of algorithms and programs are implemented in Matlab software with a computer with 2.21 GHZ and with 1.00 GB RAM memory.

As an initial setting for running mutable smart bee algorithm, the following values for the basic algorithm parameters were selected: maximum cycle number = 2000, number of colony size (NP) = 8, limit = 10, solution number (SN) = NP/2, the modification rate (MR) = 0.8, and $P_m = 0.02$. As expressed before one of the important advantage of this algorithm comparing to other heuristic algorithm is hiring low amount of population (10 bees in our case) for performing search in the area and also this feature leads the algorithm to perform faster and consuming lower cost.

For bee algorithm (BA) following parameters being set: number of scout bees in hive (n) = 30, number of elite patches (e) = 3, number of best sites (m) = 10, number of bees around elite sites (nep) = 11, number of bees around best sites (nsp) = 7, and neighborhood of sites which scout bees can search (ngh) which experiments show that BA have better performance in searching the local spaces when $ngh = (ub - lb)/11$.

For Lukasik firefly algorithm the parameters set due to [20] and also for improved particle swarm optimization

algorithms the parameters being set respecting to Bae et al. [43] researches which proved that perform are acceptable in mining data in constraint spaces.

Initial parameters for self-adaptive penalty function genetic algorithm set as $P_c = .8$, and tunable $P_m = 0.04$ decrease to 0.02, and the algorithm being implemented with respect to Tessema's method [11] in Matlab.

Arithmetic experiments were repeated 30 times, starting from a random population with different seeds [38]. Also behavior of the cycle has been analyzed in three different states of constant k_1 and γ_0 to find out the effect of these terms on the power output and entropy generation by bee algorithm (BA), improved particle swarm optimization (IPSO), Lukasik firefly algorithm (LFFA), classical artificial bee colony (ABC), and self-adaptive penalty function genetic algorithm (SAPF-GA) for making a compromise.

At the first step the performance of the Atkinson cycle analyzed in $\gamma_0 = 1.31$ and $k_1 = 0.00006$ and the results are shown in Table 1.

It is obvious that the proposed algorithm performs better than others and in some cases we find self-adaptive penalty function genetic algorithm (SAPF-GA) as well as proposed MSB-ABC algorithm after 30 times running but this algorithm use more time (22.2 seconds) for reaching to optimum solution comparing to other algorithms because this algorithm hire more than 60 chromosomes for performing efficient search in constraint spaces. As the table shows the MSB-ABC algorithm reached to fitter maximum power output and lower entropy generated during the performance of the Atkinson cycle and also because of hiring just 8 bees for searching in the constraint area of our problem, it takes acceptable CPU time (just 5.2 seconds) for finding the optimal condition. IPSO and LFFA show similar results and also the results show that they consume equal CPU time. Karaboga's classical artificial bee colony find acceptable solution in this case but as it is shown it takes noticeable time for reaching to fit solution and this matter refers to hiring 30 bees in the searching space. Bee algorithm finds an acceptable amount of entropy generation but it was weak in finding optimum power output and it takes 18.4 seconds for optimizing process. At the end of the first step the power output of the Atkinson cycle and the performance of each algorithms are shown in the following plots and after that the convergence rate of each algorithm and the capability of each of them will be discussed briefly due to obtained plots.

TABLE 2: performance of tested algorithm in $\gamma_0 = 1.41$ and $k_1 = 0.00006$.

	Parameters				Entropy generation	Power output	CPU time
	T_1	T_2	T_3	T_4	σ	\dot{W}	$t(\text{sec})$
MSB-ABC	360	954.9	2126.7	714.6	0.0013	0.3148	10.05
ABC	360	908.9	2067.0	721.6	0.0012	0.3009	15.30
BA	360	1007.6	2200.0	466.7	0.0019	0.2376	13.22
IPSO	360	974.3	2200.0	782.6	0.0017	0.2444	14.12
LFFA	360	991.9	2112.9	689.2	0.0012	0.3070	17.21
SAPF-GA	360	992.4	2008.1	699.1	0.0014	0.2912	44.09
Optimum performance reported in [40, 42]					0.0012	0.3327	—

TABLE 3: performance of tested algorithm in $\gamma_0 = 1.36$ and $k_1 = 0.00009$.

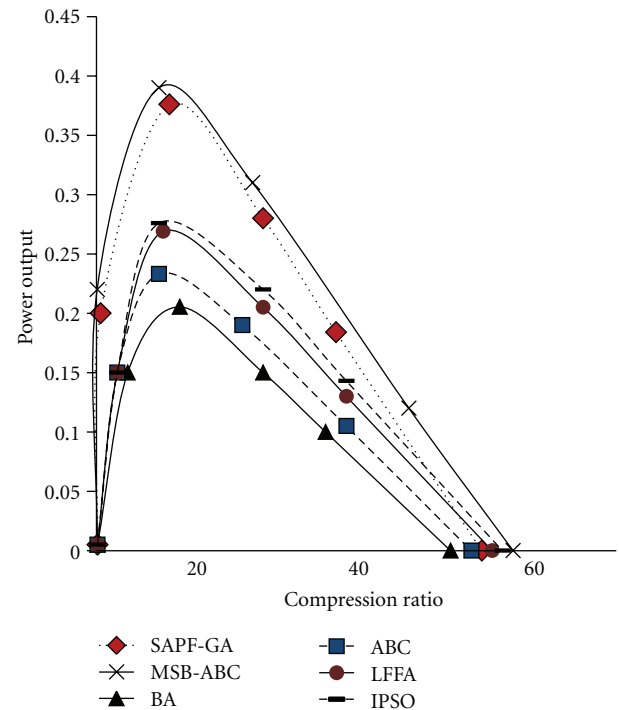
	Parameters				Entropy generation	Power output	CPU time
	T_1	T_2	T_3	T_4	σ	\dot{W}	$t(\text{sec})$
MSB-ABC	360	1100.1	2200.0	779.4	0.0009	0.3253	11.05
ABC	360	1050.1	2138.2	726.5	0.0008	0.3848	17.34
BA	—	—	—	—	—	—	—
IPSO	360	1323.2	2090.1	703.1	0.0015	0.3155	15.01
LFFA	360	1125.7	1823.9	630.0	0.0012	0.2889	14.32
SAPF-GA	360	1253.7	1902.8	570.3	0.0010	0.2773	44.72
Optimum performance reported in [40, 42]					0.0012	0.3202	—

In the first step the performance of each algorithm for finding the maximum power output will be analyzed and the maximum power out will be shown in Figure 4. As it is shown in Table 1 and Figure 4 MSB-ABC and SAPF-GA find more optimum results and IPSO and LFFA act similar; also artificial bee colony (ABC) and bee algorithm (BA) show acceptable results.

In Figure 5, the performance of each algorithm is shown during the iterations. According to results MSB-ABC show better performance in this case and also it escapes from restricted area faster than other algorithms. The capability of each algorithm for escaping from unfeasible regions is shown in Figure 6.

The results indicate that MSB-ABC and SAPF-GA are more capable to escape from constraints and also BA and ABC have lower performance to escape from restricted area and spend more time for this process. LFFA and IPSO are very similar in beating the tricks both in quality and duration. According to initial setting of the parameters it seems that MSB-ABC must try harder than other algorithms to escape from local convergence because of its low amount of initial searcher, but when we set a fit mutation probability and limit, MSB-ABC performs really efficient for escaping from unfeasible region.

Also the performance of the Atkinson cycle will be shown in different compression ratios in Figures 7 and 8 and it will be indicated that in three states of γ_0 and k_1 the algorithms found fitter power output comparing to experimental data in [39]. As it is shown in $\gamma_0 = 1.31$ and $k_1 = 0.00009$, the Atkinson cycle produces maximum power output. According to Tables 2 and 3, it is obvious that the power output

FIGURE 4: Performance of tested algorithm in $\gamma_0 = 1.31$ and $k_1 = 0.00006$.

will rise when the constant γ_0 reduced and the constant k_1 increased.

In the next step, the performance of the Atkinson cycle will be analyzed under $\gamma_0 = 1.36$ and $k_1 = 0.00006$ and the results are shown in Table 4.

TABLE 4: Performance of tested algorithm in $\gamma_0 = 1.36$ and $k_1 = 0.00006$.

	Parameters				Entropy generation σ	Power output \dot{W}	CPU time $t(\text{sec})$
	T_1	T_2	T_3	T_4			
MSB-ABC	360	953.4	2137.6	751.8	0.0012	0.3740	12.88
ABC	360	881.9	2052.1	766.4	0.0015	0.3490	19.1
BA	360	1615.7	2200.0	481.7	0.0007	0.2052	15.2
IPSO	360	920.9	2052.5	741.1	0.0014	0.3464	22.3
LFFA	360	903.8	2095.8	757.7	0.0012	0.3368	21.9
SAPF-GA	360	892.5	2142.8	792.5	0.0016	0.3584	34.2
Optimum performance reported in [40, 42]					0.0012	0.3165	—

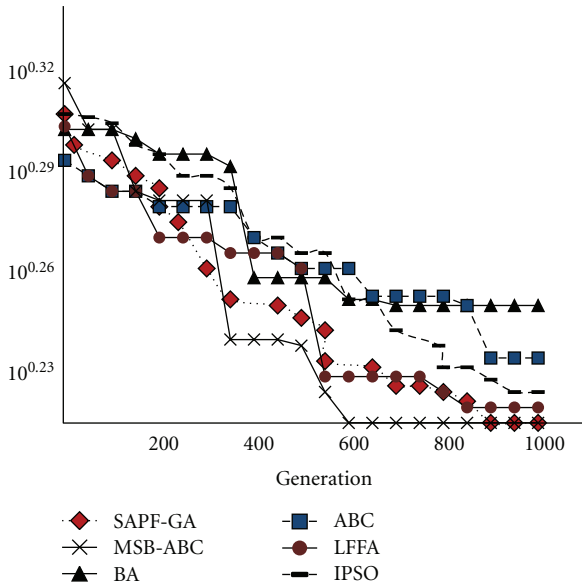


FIGURE 5: Comparison of performance of SPFA-GA, L-FFA, I-PSO, BA, ABC, and MSB-ABC in efficiency analyzing of the Atkinson cycle.

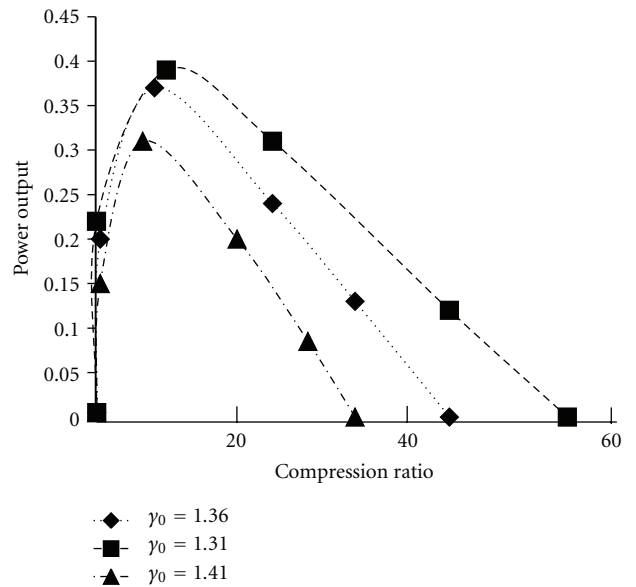


FIGURE 7: Effect of γ_0 on the power output with compression ratio.

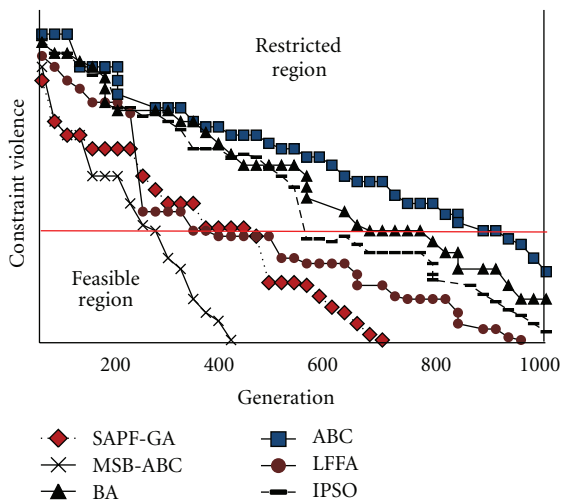


FIGURE 6: Compression of capability of algorithms to escape from restricted area.

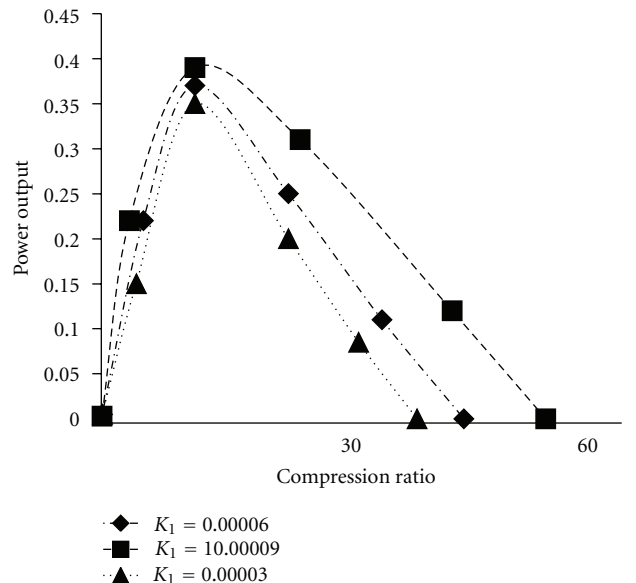


FIGURE 8: Effect of K_1 on the power output with compression ratio.

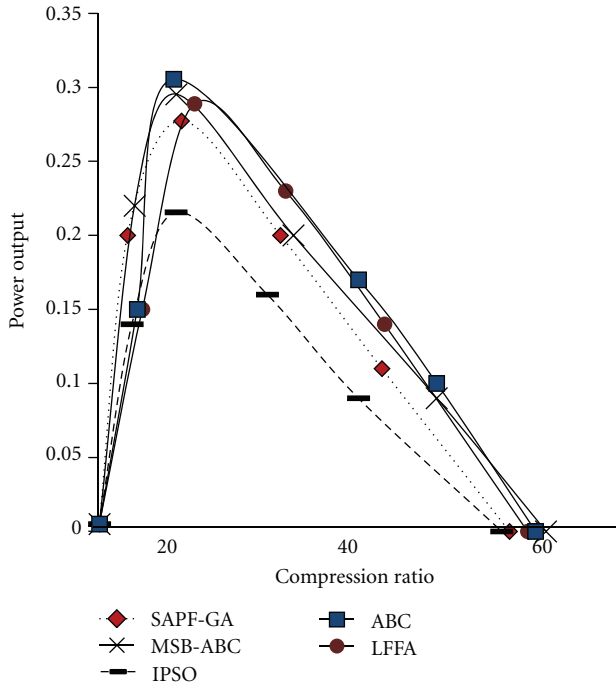


FIGURE 9: Performance of tested algorithm in $\gamma_0 = 1.36$ and $k_1 = 0.00009$.

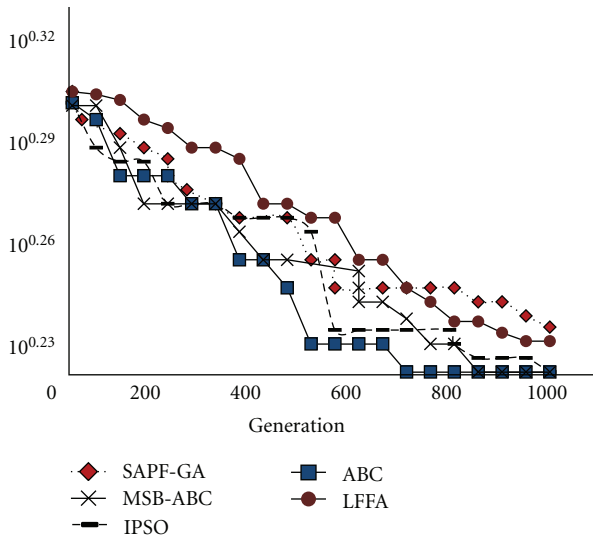


FIGURE 10: Comparison of performance of SPFA-GA, L-FFA, I-PSO, ABC, and MSB-ABC in efficiency analyzing of the Atkinson cycle.

Again the MSB-ABC shows promising results. The time duration for finding optimal solution is acceptable and also it finds better power output. This time bees algorithm (BA) finds the minimum entropy generation rate, but it was not successful in finding maximum power output. SAPF-GA finds near optimal solution but it performs weaker than other algorithms. In fact it reaches to a local optimum solution. Figure 11 shows the performance of the Atkinson cycle and Figure 12 shows the performance of these algorithms in a semilogical plot.

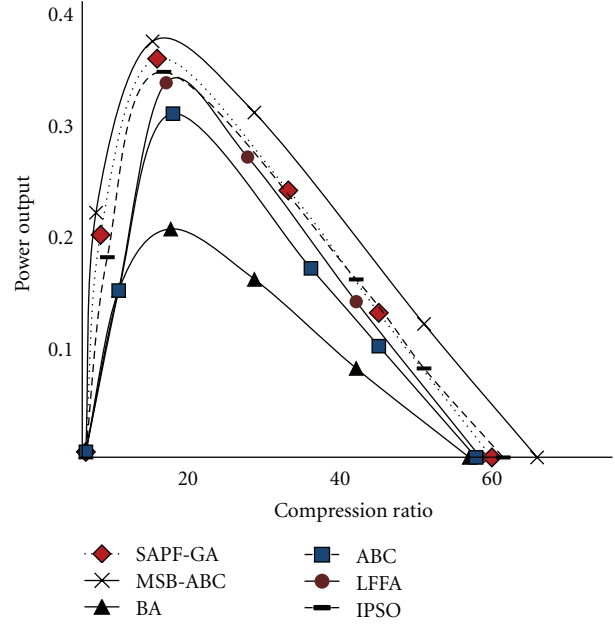


FIGURE 11: Performance of tested algorithm in $\gamma_0 = 1.36$ and $k_1 = 0.00006$.

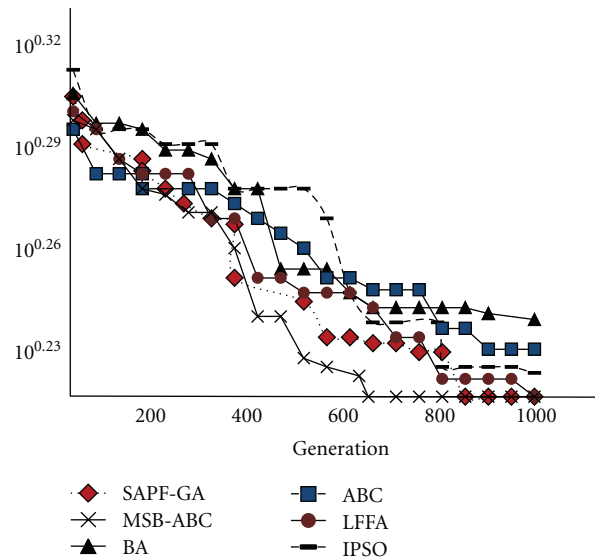


FIGURE 12: Comparison of performance of SPFA-GA, L-FFA, I-PSO, BA, ABC, and MSB-ABC in efficiency analyzing of the Atkinson cycle.

According to Figures 9 and 10, proposed modified algorithm performs more efficiently than other algorithms in most cases. Besides, smart bees are capable to escape from various constraints in a short time, where other algorithms use more time for escaping from all constraints. One of the other important advantages of proposed algorithm is its ability to work with a low population size. This feature makes this algorithm really faster than other algorithms. In addition, mutation phase helps smart bees to escape from local optimums.

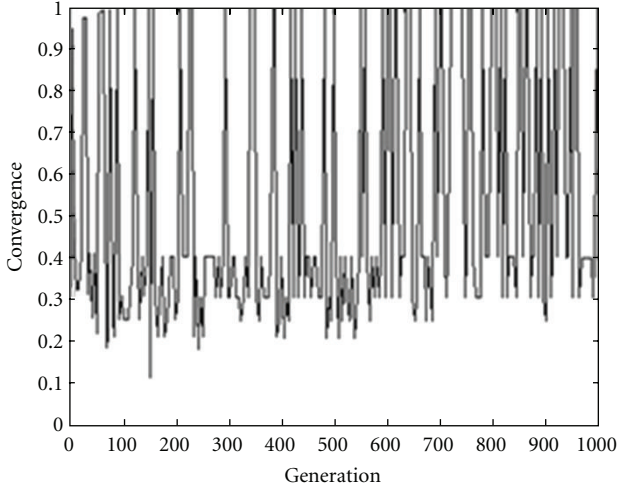


FIGURE 13: Convergence rate of MSB-ABC in the first step.

For the last case, the performance of the Atkinson cycle will be checked in $K = 0.00003$ and $\lambda = 1.36$ and the results will be shown in Table 5.

As it is shown, the Bee Algorithm find lower entropy generation, however, it does not find an acceptable power output. LFFA and MSB-ABC perform promising both in maximizing power output and minimizing the unexpected amount of entropy generation. MSB-ABC consumes lower CPU time to find the optimal solution and this feature leads the MSB-ABC algorithm to perform as a superior algorithm in this case.

6. Analyzing Convergence Rate

One of the other important aspects that prove the advantage of MSB-ABC algorithm is the capability of this algorithm to escape from local optimal values. This claim will be demonstrated in the following plots which indicate the rate of convergence for algorithms during the optimizing process.

For analyzing the convergence ratio of these algorithms this parameter should be defined as:

$$\begin{aligned} \text{Mean cost} &= \frac{\sum_{i=1}^{\text{popsize}} \text{cost}(i)}{\text{popsize}}, \\ \text{Best cost} &= \text{Min}\{\text{cost}\}, \\ \text{Convergence rate} &= \frac{\text{best cost}}{\text{Mean cost}}. \end{aligned} \quad (25)$$

In the first step the convergence rate of the proposed algorithm will be analyzed under $P_m = 0.08$ and limit = 7 during the optimizing process.

Figure 13 reveals that the algorithm's convergence rate changes very fast and it does not have enough time to perform an acceptable neighbor search during the process.

For that we tune the mutation probability as following:

$$P_m = P'_m - K_{P_m} \frac{t}{T}, \quad (26)$$

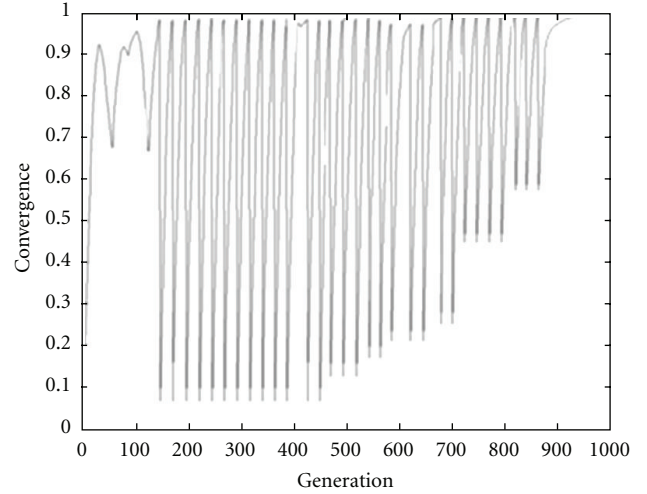


FIGURE 14: Convergence rate of MSB-ABC in the second step.

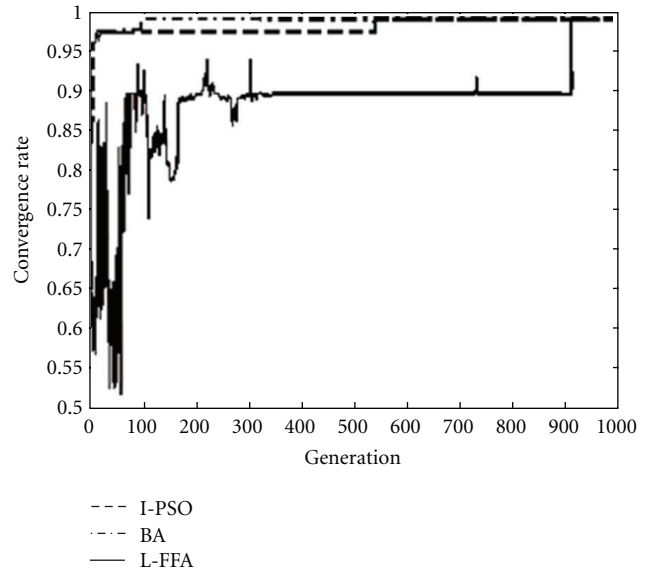


FIGURE 15: Analytic comparison of convergence ratio of BA, LFFA, and IPSO.

where $P'_m = 0.05$, K_{P_m} is a constant number that control P_m , t is current iteration, and T is maximum iteration.

According to Figure 14, the results are acceptable in this case. As it is shown, in the initial iterations, algorithm searches a wide space for finding better regions (food patches) and then it concentrates on neighbor search to reach to the bottom of the valley (global minimum).

At the end the convergence rate of BA, L-FFA, and I-PSO are shown (Figure 15) for make a contrast.

According to the results it is obvious that when we use adaptive parameters, MSB-ABC shows better reaction for escaping from local optimum regions comparing to other algorithms. It seems that bee algorithm (BA) suffers from fast local optimum convergence during optimizing process and again L-FFA and I-PSO have similar behavior. The obtained

TABLE 5: Performance of tested algorithm in $\gamma_0 = 1.36$ and $k_1 = 0.00003$.

	Parameters				Entropy generation	Power output	CPU time
	T_1	T_2	T_3	T_4	σ	\dot{W}	$t(\text{sec})$
MSB-ABC	360	992.3	1889.3	831.1	0.0012	0.3572	10.05
ABC	360	1987.6	1984.8	612.3	0.0009	0.2368	16.23
BA	360	1002.2	1760.5	566.0	0.0008	0.2067	8.05
IPSO	360	959.9	1730.9	698.9	0.0013	0.3069	12.41
LFFA	360	864.8	2200.0	764.8	0.0014	0.3489	12.79
SAPF-GA	360	1199.5	2112.1	618.3	0.0012	0.3262	35.23
Optimum performance reported in [40, 42]					0.0012	0.3148	—

results demonstrate that MSB-ABC algorithm is one of the most applicable algorithms for optimizing multimodal problems since it is capable to balance the intensive local search strategy and an efficient exploration of the whole search space simultaneously.

7. Conclusions

In this paper, a new method called MSB algorithm proposed for optimizing a well-known multimodal engineering problem, based on the reaction of mutable smart bees during the procedure. Thereafter, proposed algorithm has been compared with some famous optimization methods such as self-adapting penalty function genetic algorithms and improved particle swarm optimization. The results illustrate that MSB algorithm is superior or equal to these existing algorithms for optimizing multimodal problems in most cases. This issue refers to the fine tuning of the parameters that may results efficient searching in feasible space. Furthermore, our simulations indicate that because of adaptive mutation that occurs in smart bee, the algorithm has a suitable convergence rate that leads the algorithm to escape from local optimum solution. Subsequently, it seems that MSB algorithm is more generic and robust for many constraint optimization problems, comparing to other metaheuristic algorithms.

Nomenclature

C_p :	Isobaric molar specific heat (kJ/kg K)
C_v :	Isochoric specific heat(kJ/kg K)
E_H :	Effectiveness of hot heat exchanger
E_L :	Effectiveness of cold heat exchanger
F:	Heat transfer surface area (m^2)
M:	Molar mass of working fluid (kg/mol)
Q_{in} :	Heat added to working fluid (kW)
Q_{leak} :	Heat leakage (kW)
Q_{out} :	Heat rejected from working fluid (kW)
R:	Molar gas constant
r_C :	Specific compression ratio
r_C^* :	Compression ratio
V_1 :	Volume in state one (m^3)
V_2 :	Volume in state two (m^3)
V_3 :	Volume in state three (m^3)
V_4 :	Volume in state four (m^3)
\dot{W} :	Output power (kW).

Greek Symbols

γ :	Specific heat ratio
α :	Heat transfer coefficient (kW/Km^2)
η_{th} :	Thermal efficiency
σ :	Entropy generation of the cycle.

Acknowledgments

The authors would like to thank S. Noudeh and P. Samadian for their precious collaboration.

References

- [1] C. Lyle Cummins, *Internal Fire: The Internal-Combustion Engine 1673–1900*, Carnot Press, Wilsonville, Ore, USA, 2000.
- [2] H. S. Leff, "Thermal efficiency at maximum power output: new results for old heat engine," *American Journal of Physics*, vol. 55, pp. 602–610, 1987.
- [3] A. Al-Sarkhi, B. A. Akash, J. O. Jaber, M. S. Mohsen, and E. Abu-Nada, "Efficiency of miller engine at maximum power density," *International Communications in Heat and Mass Transfer*, vol. 29, no. 8, pp. 1159–1167, 2002.
- [4] P. Y. Wang and S. S. Hou, "Performance analysis and comparison of an Atkinson cycle coupled to variable temperature heat reservoirs under maximum power and maximum power density conditions," *Energy Conversion and Management*, vol. 46, no. 15-16, pp. 2637–2655, 2005.
- [5] S. S. Hou, "Comparison of performances of air standard Atkinson and Otto cycles with heat transfer considerations," *Energy Conversion and Management*, vol. 48, no. 5, pp. 1683–1690, 2007.
- [6] J. Dréo, P. Siarry, A. Pétrowski, and E. Taillard, *Metaheuristics for Hard Optimization*, Springer, Heidelberg, Germany, 2006.
- [7] Z. Michalewicz, "Heuristic methods for evolutionary computation techniques," *Journal of Heuristics*, vol. 1, no. 2, pp. 177–206, 1996.
- [8] X. S. Yang, *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, Beckington, UK, 2008.
- [9] J. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, Mich, USA, 1975.
- [10] L. Davis, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, NY, USA, 1991.
- [11] B. Tessema and G. G. Yen, "A self adaptive penalty function based algorithm for performing constrained optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation*, G. G. Yen and M. Simon, Eds., pp. 246–253, IEEE Publications, Vancouver, Canada, 2006.

- [12] A. C. C. Lemonge and H. J. C. Barbosa, "An adaptive penalty scheme in genetic algorithms for constrained optimization problems," in *Proceedings of the Genetic and Evolutionary Computation Conference*, G. Paun, Ed., pp. 287–294, ACM Press, New York, NY, USA, 2002.
- [13] S. M. Saab, N. Kamel, T. El-Omari, and H. Owaied, "Developing optimization algorithm using artificial Bee Colony system," *Ubiquitous Computing and Communication Journal*, vol. 4, pp. 391–396, 2009.
- [14] M. Dorigo, V. Maniezzo, and A. Colorni, "The ant system: optimization by a colony of cooperating agents," in *Transactions on Systems Man and Cybernetics*, W. Pedrycz, Ed., pp. 29–41, IEEE Publications, Alberta, Canada, 1996.
- [15] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, L. M. LeCam and J. Neyman, Eds., pp. 1942–1948, IEEE Publications, Piscataway, NJ, USA, 1995.
- [16] J. Kennedy, "Minds and cultures: particle swarm implications, Socially Intelligent Agents," *Ubiquitous Computing and Communication Journal*, vol. 23, pp. 67–72, 1997.
- [17] Y.-P. Bu, Z. Wei, and J.-S. You, "An improved PSO algorithm and its application to grid scheduling problem," in *Proceedings of the International Symposium on Computer Science and Computational Technology (ISCSC '08)*, pp. 352–355, IEEE, 2008.
- [18] X. S. Yang and S. Deb, "Cuckoo search via Levy flights," in *Proceedings of the World Congress on Nature & Biologically Inspired Computing (NaBIC '09)*, F. Rothlauf, Ed., pp. 210–214, IEEE Publications, India, December 2009.
- [19] S. Lukasik and S. Zak, "Firefly algorithm for continuous constrained optimization tasks," in *Proceedings of the International Conference on Computer and Computational Intelligence*, N. T. Nguyen, R. Kowalczyk, and S. M. Chen, Eds., vol. 5796 of *Lecture Notes in Computer Science*, pp. 97–106, Springer, Wrocław, Poland, 2009.
- [20] X.-S. Yang, "Firefly algorithms for multimodal optimization," in *Proceedings of the Stochastic Algorithms: Foundations and Applications*, vol. 5792 of *Lecture Notes in Computer Science*, pp. 169–178, Springer, Sapporo, Japan, 2009.
- [21] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial Bee Colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.
- [22] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," *Applied Soft Computing Journal*, vol. 8, no. 1, pp. 687–697, 2008.
- [23] S. Camazine and J. Sneyd, "A model of collective nectar source selection by honey bees: self-organization through simple rules," *Journal of Theoretical Biology*, vol. 149, no. 4, pp. 547–571, 1991.
- [24] D. T. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, and M. Zaidi, "The bees algorithm," Technical Note, Manufacturing Engineering Centre, Cardiff University, UK, 2005.
- [25] T. D. Seeley, *The Wisdom of the Hive: The Social Physiology of Honey Bee Colonies*, Harvard University Press, 1996.
- [26] Y. Yonezawa and T. Kikuchi, "Ecological algorithm for optimal ordering used by collective honey bee behavior," in *Proceedings of the 7th International Symposium on Micro Machine and Human Science*, pp. 249–256, Nagoya, Japan, 1996.
- [27] T. Sato and M. Hagiwara, "Bee system: finding solution by a concentrated search," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 3954–3959, Orlando, Fla, USA, 1997.
- [28] D. Teodorovic, *Bee Colony Optimization (BCO)*, Ministry of Science of Serbia, Belgrade, Serbia, 2001.
- [29] H. A. Abbass, "Marriage in honey bee optimization (MBO): a haplometrosis polygynous swarming approach," in *Proceedings of the IEEE Conference on Evolutionary Computation (CEEC '01)*, vol. 1, pp. 207–214, IEEE Publications, Seoul, Korea, 2001.
- [30] D. Karaboga, *An Idea Based on Honey Bee Swarm for Numerical Optimization*, Erciyes University, Kayseri, Turkey, 2005.
- [31] C. S. Chong, M. Y. H. Low, A. I. Sivakumar, and K. L. Gay, "A Bee Colony Optimization Algorithm to job shop scheduling simulation," in *Proceedings of the Winter Conference*, L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nichol, and R. M. Fujimoto, Eds., pp. 1954–1961, Washington, DC, USA, 2006.
- [32] N. Stanarevic, M. Tuba, and N. Bacanin, "Modified Artificial Bee Colony algorithm for constrained problems optimization," *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 5, no. 3, pp. 644–651, 2011.
- [33] S. E. Elmaghraby, H. Soewandi, and M. J. Yao, "Chance-constrained programming in activity networks: a critical evaluation," *European Journal of Operational Research*, vol. 131, no. 2, pp. 440–458, 2001.
- [34] F. Hillier, "Chance-constrained programming with 0-1 or bounded continuous decision variables," *Journal of Management Science*, vol. 14, pp. 34–57, 1967.
- [35] Y. Seppala, "Constructing sets of uniformly tighter linear approximations for a chance-constraint," *Journal of Management Science*, vol. 17, pp. 736–749, 1971.
- [36] Y. Seppälä and T. Orpana, "Experimental study on the efficiency and accuracy of a chance-constrained programming algorithm," *European Journal of Operational Research*, vol. 16, no. 3, pp. 345–357, 1984.
- [37] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2–4, pp. 311–338, 2000.
- [38] D. Karaboga and B. Basturk, "Artificial Bee Colony (ABC) optimization algorithm for solving constrained optimization problems," in *Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing*, C. Ozturk, Ed., pp. 789–798, Springer, Berlin, Germany, 2007.
- [39] R. Ebrahimi, "Performance of an Endoreversible Atkinson cycle with variable specific heat ratio of working fluid," *American Journal of Science*, vol. 6, pp. 12–17, 2010.
- [40] Y. Ge, L. Chen, and F. Sun, "Finite time thermodynamic modeling and analysis for an irreversible atkinson cycle," *Thermal Science*, vol. 14, no. 4, pp. 887–896, 2010.
- [41] A. Al-Sarkhi, B. Akash, E. Abu-Nada, and I. Al-Hinti, "Efficiency of atkinson engine at maximum power density using temperature dependent specific heats," *Jordan Journal of Mechanical and Industrial Engineering*, vol. 2, pp. 71–75, 2005.
- [42] C. Lingen, Z. Wanli, and S. Fenguri, "Power, efficiency, entropy-generation rate and ecological optimization for a class of generalized irreversible universal heat-engine cycles," *Applied Energy*, vol. 84, no. 5, pp. 512–525, 2007.
- [43] J. Bae J, P. Y. Won, J. J. Rin, and K. Y. S. Lee, "An improved particle swarm optimization for nonconvex economic dispatch problems," *IEEE*, vol. 25, pp. 156–166, 2010.




Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

