Computing and Informatics, Vol. 35, 2016, 222–240

# IPO: AN INCLINED PLANES SYSTEM OPTIMIZATION ALGORITHM

Mohammad Hamed Mozaffari

*School of Electrical Engineering and Computer Science*
*University of Ottawa*
*800 King Edward Avenue, Ottawa*
*Ontario, Canada K1N-6N5*
*e-mail:* mmoza102@uottawa.ca

Hamed Abdy

*Thin Film Laboratories*
*School of Electrical and Computer Engineering*
*University College of Engineering*
*University of Tehran*
*North Kargar st.*
*Tehran, I. R. Iran*
*P. O. Box: 14395-515*
*e-mail:* hamed.abdy@ut.ac.ir

Seyed-Hamid Zahiri

*Department of Electrical Engineering*
*Faculty of Engineering*
*University of Birjand*
*Birjand, I. R. Iran*
*P. O. Box: 97175-376*
*e-mail:* hzahiri@birjand.ac.ir

**Abstract.** In the last decades, heuristic algorithms are widely used in solving problems in different fields of science and engineering. Most of these methods are inspired by natural phenomena, such as biological behaviours or physical principles.

In this paper, a new optimization method based on the dynamic of sliding motion along a frictionless inclined plane is introduced. In the proposed algorithm, a collection of agents cooperate with each other and move toward better positions in the search space by employing Newton's second law and equations of motion. Our method is compared with other popular optimization algorithms and the results on 23 standard benchmark functions show its effectiveness in most cases.

**Keywords:** Heuristic algorithm, inclined planes system optimization, Newton's second law, optimization, swarm intelligence

**Mathematics Subject Classification 2010:** 90C59

## 1 INTRODUCTION

With advance of science, solving complex optimization problems attracts more research efforts. By increasing the dimensions of optimization problems, the search space increases exponentially. This makes classical approaches incapable in solving most of these problems. Over the last decades, new algorithms called heuristics have been developed [1, 2, 3, 4] to find suitable solutions at a reasonable computational cost. These algorithms have shown their effectiveness in solving many problems in different fields of science such as logistics [5], bioinformatics [6], data mining [7], chemical physics [8], electronics [9], etc.

It is hard (if not possible) to find a mathematical model for heuristic algorithms search process [7]. Many optimization methods have been proposed in the literature [1, 2, 3, 4, 11, 12, 13, 14, 15, 16, 17, 18], but there is not a supreme method for all optimization problems and some algorithms may achieve better results in some specific cases. This encourages researchers to develop new optimization algorithms.

In this paper, a new optimization algorithm called Inclined Planes system Optimization (IPO), based on the dynamic of moving tiny balls on inclined planes is proposed. The rest of this paper is organised as follows: Section 2 presents a brief review on previous optimization methods in the literature. In Section 3 the details of proposed method is described. The experimental and comparison results on 23 well-known benchmark functions are presented in Section 4. Finally, Section 5 draws our conclusion and future works.

## 2 A BRIEF REVIEW ON HEURISTIC ALGORITHMS

A heuristic algorithm is the way for finding an adequate and quick solution to a problem without detailed knowledge about the problem [19].

Heuristic algorithms are usually inspired by nature and its physical or biological processes. A large number of heuristic algorithms were presented in the last decades. The most popular and famous of these algorithms are Genetic Algorithm (GA), Simulated Annealing (SA), Harmony Search (HS), Artificial Immune System (AIS), Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO).

GA is formed from the laws of natural selection and genetics based on the Darwin's theory of evolution [22]. SA is designed by using of the process of annealing in metallurgy [13]. HS is an algorithm mimicking musician's behaviours in improvisation process [14]. AIS is inspired by biological immune systems [12]. ACO is simulated from the foraging behaviour of real ants when they search for food [3]. PSO simulates the social behaviour in flock of birds in migration [2].

Population-based methods are inspired by the dynamics of social interactions between individuals. In this procedure, each particle tries to move toward the best position by using its own previous experience and guidance of its neighbour particles [23]. Sharing information in population-based algorithms is a common politics, so each individual shares its information with others in order to guide the swarm to its goal (optimum position). This cooperation between particles is known as swarm intelligence, which has significant improvement effect on results of algorithms [3].

This paper investigates a design for a new heuristic algorithm to address various optimization problems. In the next section, our method will be explained in details.

## 3 INCLINED PLANES SYSTEM OPTIMIZATION (IPO)

In this section the proposed optimization algorithm is explained in details. This section is divided into two subsections: at first, the basic concepts and important explanations are presented for minimization problems. Then, its l-best and g-best versions are introduced for the best compromise between exploration and exploitation.

### 3.1 Principles of IPO

The proposed IPO algorithm is inspired by the dynamic of sliding motion along frictionless inclined surfaces. In IPO, some tiny balls search the problem space to find near optimal (here, minimum) solutions. These tiny balls in IPO are agents of the algorithm (like particles in PSO or ants in ACO). The main idea of IPO is to assign heights to these tiny balls, regarding to their fitnesses. These heights are employed to estimate the potential energy of each ball that should be transformed to kinetic energy by achieving appropriate acceleration. In fact, balls would like to lose their potential energy and to get the reference point(s), which is (are) equal to the minimum solution(s). Hence, balls accelerate and search the problem space for better solutions, iteratively (see Figure 1).

Each ball has three specifications: position, height, and angles made with other balls. The position of each ball is a feasible solution for the problem, and heights of balls are determined using a fitness function. To estimate an inclined plane on which the ball is located, angles are calculated by crossing straight lines from center of ball to centers of other balls. For a minimization problem, angles that emerge between

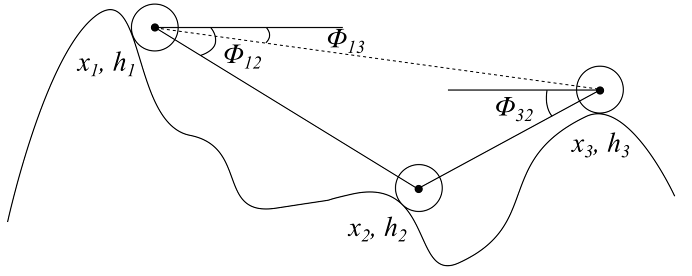



Figure 1. An example of problem search space with three balls and estimated inclined planes

these straight lines and the horizontal line are employed to find the direction and amount of acceleration of each ball. Consider a system with $N$ balls. We define the position of the $i^{\text{th}}$ ball as the following equation:

$$\vec{x}_i = (x_i^1, \ldots, x_i^d, \ldots, x_i^n), \quad \text{for } i = 1, 2, \ldots, N \tag{1}$$

and

$$x_j^{min} \leq x_j \leq x_j^{max}, \quad 1 \leq j \leq n \tag{2}$$

where $x_i^d$ represents the position of $i^{\text{th}}$ ball in the $d^{\text{th}}$ dimension in an $n$ dimensional space. The goal is to determine the location of global minima of objective function $f(\vec{x})$ defined on the problem space. At the specific time $t$, angle between the $i^{\text{th}}$ ball and $j^{\text{th}}$ ball in dimension $d$, i.e. $\phi_{ij}^d$, is calculated as follows:

$$\phi_{ij}^d(t) = \left(\tan^{-1}\left(\frac{f_j(t) - f_i(t)}{x_i^d(t) - x_j^d(t)}\right)\right) \tag{3}$$
$$\text{for } d = 1, \ldots, n \text{ and } i, j = 1, 2, \ldots, N, i \neq j$$

where $f_i(t)$ is the value of fitness function (height) for the $i^{\text{th}}$ ball in time $t$. For reducing computational complexity in IPO, the acceleration of each ball in each dimension, is calculated separately from other dimensions. According to our experiments, calculating the acceleration in each dimension separately also leads to better results. Whilst each ball tends to move toward lower heights on each plane, acceleration is evaluated only for balls with lower heights (fitnesses). These accelerations on various planes are added up to obtain the total acceleration of the ball. It is known from the Newton's second law:

$$\Sigma\vec{F} = m \cdot \vec{a}. \tag{4}$$

Hence, the acceleration of a moving object on a frictionless inclined plane is calculated as follows:

$$a = g \cdot \sin(\phi) \tag{5}$$

where $g$ is constant value of gravitational acceleration, and $\phi$ is the angle with respect to horizontal surface. In the proposed IPO algorithm, the amount and direction of acceleration for the $i^{\text{th}}$ ball at time (iteration) $t$ and in dimension $d$ is calculated as follows:

$$a_i^d(t) = \sum_{j=1}^{N} \mathrm{U}\left(f_j(t) - f_i(t)\right) \cdot \sin\left(\phi_{ij}^d(t)\right) \tag{6}$$

where $\mathrm{U}(\cdot)$ is the Unit Step function:

$$\mathrm{U}(w) = \begin{cases} 1 & w > 0, \\ 0 & w \leq 0. \end{cases} \tag{7}$$

In Equation (6), constant $g$ is omitted for reducing the computational costs. IPO uses the law of motion with constant acceleration for updating positions of balls:

$$x_i^d(t+1) = k_1 \cdot rand_1 \cdot a_i^d(t) \cdot \Delta t^2 + k_2 \cdot rand_2 \cdot v_i^d(t) \cdot \Delta t + x_i^d(t) \tag{8}$$

where $rand_1$ and $rand_2$ are two random weights with uniform distribution in the interval $[0, 1]$ to give stochastic characteristic to our algorithm, and $v_i^d(t)$ is the velocity of ball $i$ in dimension $d$, at time $t$. For controlling the search process of algorithm, two essential constants named $k_1$ and $k_2$ are used. These constants can be functions of time (iteration): $k_1$ should be decreased and $k_2$ should be increased from their initial values. These two parameters balance the trade-off between the exploration and exploitation of the algorithm. Additional descriptions about these parameters are presented in the next subsection. $v_i^d(t)$ is calculated as follows:

$$v_i^d(t) = \frac{x_{best}^d(t) - x_i^d(t)}{\Delta t} \tag{9}$$

where $x_{best}$ is the ball with lowest height, i.e. fitness, among other balls in all iterations till current iteration. In fact, Equations (8) and (9) are adopted from the dynamic of motion with constant acceleration in classical mechanics:

$$\vec{x} = \frac{1}{2}\vec{a} \cdot t^2 + \vec{v}_0 \cdot t + \vec{x}_0 \tag{10}$$

and

$$\vec{v} = \frac{\Delta \vec{x}}{\Delta t}. \tag{11}$$

In calculating initial velocity in Equation (9), $x_{best}$ is used in numerator. This change is applied to the initial velocity because in any iteration, balls have a tendency to reach to the best individual location in society.

The pseudo code of IPO algorithm is illustrated in Algorithm 1 and its flowchart is shown in Figure 2.

Algorithm 1 and above mentioned equations show that how the proposed algorithm is related to the mechanical frictionless movements. But for mathematical

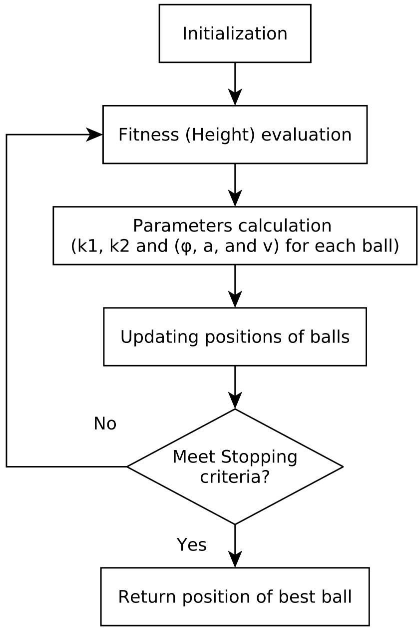



Figure 2. Flowchart of Inclined Planes system Optimization (IPO) algorithm

**Algorithm 1** Pseudo code for Inclined Planes system Optimization (IPO) algorithm

Generate randomized population, $k_1$, $k_2$, and other parameters
**repeat**
    Evaluate the fitness (height) of each ball
    Calculate angle, acceleration, and the velocity of each ball (Equations (3), (6), and (9)).
    Update the position of each ball (Equation (8)).
    Evaluate $k_1$ and $k_2$
    Return particles that went out of problem boundaries into problem boundaries
**until** Meeting the stopping criteria

calculations and coding the algorithm, this is not necessary to impose these computational costs to IPO. Referring to Equation (5), only $\sin(\phi)$ have to be measured and it can be calculated from the Figure 1 without necessity to value of $\tan(\phi)$, by this equation:

$$\sin(\phi_{ij}^d) = \frac{fit_j(t) - fit_i(t)}{R(b_i, b_j)} \tag{12}$$

where $R(b_i, b_j)$ is the Euclidean distance of $i^{\text{th}}$ and $j^{\text{th}}$ ball in dimension $d$. Indeed, only a simple division is needed to evaluate the $\sin(\phi)$ after computing $R(b_i, b_j)$.

It should be mentioned that in both IPO and PSO the optimization is performed by agents' movement in the search space; however, the movement strategies are different. Some important differences are as follows:

- PSO stores best visited positions by each particle for updating the velocity, but in IPO only the current positions of the agents play an important role in updating the positions.
- The search mechanisms of these algorithms are different. PSO simulates the social behaviour of birds and IPO utilizes the dynamic rules of sliding motion along a frictionless inclined plane.
- In PSO, updating is performed without considering the distance between solutions while in IPO the acceleration is reversely proportional to the distance between the agents.
- In PSO, updating is performed without considering quality of the solutions and the fitness values are not important in updating procedure. While in IPO, the acceleration is proportional to the fitness value and so the agents see the search space around themselves in the influence of their fitnesses.
- In PSO the direction of an agent is calculated using only two of the best positions, pbest and gbest. But in IPO, the direction of agent is calculated based on the overall interactions among all other agents placed below it.

### 3.2 Exploration and Exploitation in IPO

Exploration and exploitation are two essential concepts that cause optimization algorithms to efficiently search the problem space without being captured in the local optimums. Exploration reinforces algorithm to search the whole range of problem space by searching new places (like mutation operator in GA), while exploitation causes algorithm to search good places locally to find better positions (like selection operator in GA). For reaching the optimum solution, a good trade-off between exploration and exploitation is needed.

Usually, an appropriate schedule for good compromise is using exploration in earlier steps and with the lapse of time, exploitation operator will be subsequently used more, while the power of exploration will be reduced [11, 24].

In above mentioned IPO, the angle and acceleration of each ball is calculated regarding all other agents. This version of IPO is called global-best IPO (like gbest version of PSO). In global-best IPO, the balance of exploration and exploitation is made by controlling the values of $k_1$ and $k_2$. Equation (8) clearly shows that higher values of $k_1$ and lower values of $k_2$ cause greater accelerations. This leads to greater mobility of agents. It means that global searching or exploration occurs with larger values of $k_1$ and lower measures of $k_2$. On the contrary, if the values of $k_1$ and $k_2$ become smaller and larger, respectively, exploitation is emphasized. Because, in this case, the mobility of balls is limited and a local search scheme or exploration has happened.

In fact, if the acceleration is calculated only by considering the best ball, all other agents have to trace a specific direction towards the best ball. Since the acceleration of each ball is estimated regarding all the balls below it, the agent direction is formed based on the overall interactions by all other agents, placed below of it.

In next Section (Experimental and Comparative Results) an appropriate setting up of $k_1$ and $k_2$ for having a good trade-off between exploration and exploitation is proposed for global-best IPO. Another effort for more confident balance of exploration and exploitation is to introduce another version of IPO, called local-best IPO. In the local-best IPO, a neighbourhood is defined for each ball and all related calculations to angle and acceleration for this ball is executed regarding the balls in its neighbourhood. This leads to a local search for each agent and global search for the whole population of balls.

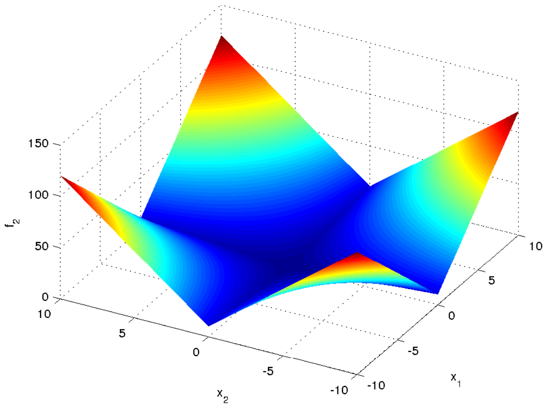
## 4 EXPERIMENTAL AND COMPARATIVE RESULTS

To show the effectiveness and the power of IPO algorithm, it is evaluated on minimizing 23 well-known benchmark functions [25]. Three examples of these benchmark functions ($f_2$, $f_9$, and $f_{14}$) are presented in Equations (13), (14), and (15), respectively. Graphs of these functions in two dimensions are also illustrated in Figure 3.

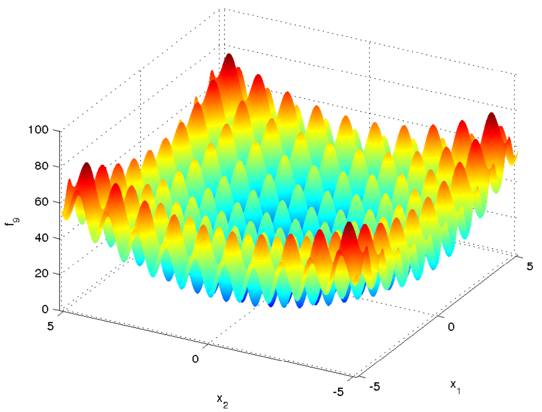
$$\begin{aligned} f_2(x) &= \sum_{i=1}^{30} |x_i| + \prod_{i=1}^{30} |x_i| \\ &\qquad -10 \leq x_i \leq 10 \\ \min(f_2) &= f_2(0,\dots,0) = 0 \end{aligned} \tag{13}$$

$$\begin{aligned} f_9(x) &= \sum_{i=1}^{30} \left[{x_i}^2 - 10\cos(2\pi x_i) + 10\right] \\ &\qquad -5.12 \leq x_i \leq 5.12 \\ \min(f_9) &= f_9(0,\dots,0) = 0 \end{aligned} \tag{14}$$

$$\begin{aligned} f_{14}(x) &= \left[\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6}\right]^{-1} \\ &\qquad -65.536 \leq x_i \leq 65.536 \\ \min(f_{14}) &= f_{14}(-32,-32) \approx 1 \\ (a_{ij}) &= \begin{pmatrix} -32 & -16 & 0 & 16 & 32 & -32 & \dots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & \dots & 32 & 32 & 32 \end{pmatrix} \end{aligned} \tag{15}$$




a)



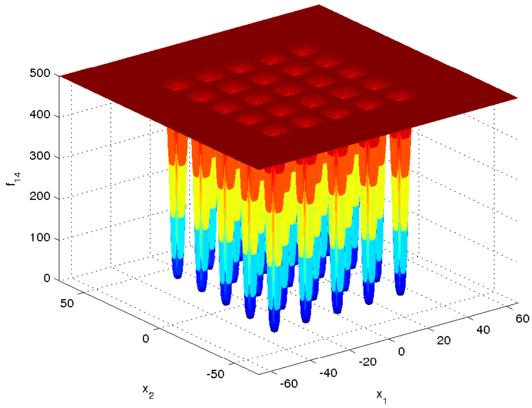

b)

For comparison, the results of two other algorithms are provided: Particle Swarm Optimization (PSO) as a swarm intelligence technique, and Genetic Algorithm (GA) as an evolutionary one. This section illustrates and discusses the results obtained by these three algorithms. Also, comparisons of computational costs of three algorithms are reported.

The problem setup is provided in the next subsection.

## 4.1 Problem Setup

In this subsection, the problem and software setup used to generate results are explained.




c)

Figure 3. Graphs of benchmark functions in two dimensions: a) $f_2$, b) $f_9$, and c) $f_{14}$

For applying different algorithms to minimization benchmark functions, Mathworks® MATLAB R2011a [26] on a PC with 2.0 GHz (two cores) CPU and 1 GB memory is used. For all test functions, in all algorithms, number of population is set to 50, number of dimensions is set to 30 (except for functions with fixed number of dimensions), and default number of iterations is set to 1 000 for $f_1$ to $f_{13}$ and 500 for $f_{14}$ to $f_{23}$. Each algorithm runs until reaching the mentioned number of iterations and there is no time limitation. More information about specific setup of functions (i.e. boundaries, etc.) is available in [25]. All algorithms are initialized by random population. The results are obtained by independent running each algorithm for 30 times, for each function, and reporting the average of results.

Genetic Algorithm toolbox of Mathworks® MATLAB is used here with its default values except number of generations, population size, and boundaries of function. The initial population used for Genetic Algorithm is generated by an external function, which is also used for generating the initial population of IPO and PSO.

Different schedules may be adopted for control parameters ($k_1(t)$ and $k_2(t)$) for IPO. In our experiments these are defined as functions of time (t), as follows:

$$k_1(t) = \frac{c_1}{1 + \exp\left((t - \mathit{shift}_1) \cdot \mathit{scale}_1\right)}, \tag{16}$$

$$k_2(t) = \frac{c_2}{1 + \exp\left(-(t - \mathit{shift}_2) \cdot \mathit{scale}_2\right)}, \tag{17}$$

where $c_1$, $c_2$, $\mathit{shift}_1$, $\mathit{shift}_2$, $\mathit{scale}_1$, and $\mathit{scale}_2$ are constants, experimentally determined for each function as follows: different combinations of constants with regard to the structure of function were tested and later the best items were fine tuned. Figure 4 illustrates an example diagram for $k_1(t)$ and $k_2(t)$ for 1 000 iterations.

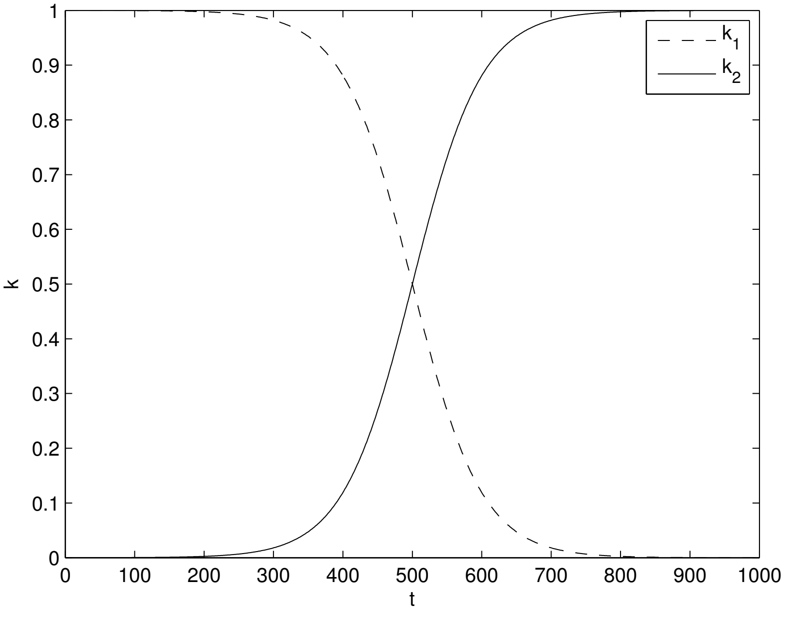



Figure 4. Control functions ($k_1$ and $k_2$) of IPO search process with $c_1 = c_2 = 1$, $shift_1 = shift_2 = 500$, and $scale_1 = scale_2 = 0.02$

For PSO, the velocity boundaries are equal to 0.2 of distance between function boundaries, i.e.:

$$\vec{v}_{max} = 0.2 \cdot (\vec{x}_{max} - \vec{x}_{min}) = -\vec{v}_{min} \tag{18}$$

where $\vec{v}_{max}$ and $\vec{v}_{min}$ are vectors of maximum and minimum velocity, respectively, and $\vec{x}_{max}$ and $\vec{x}_{min}$ are vectors of maximum and minimum boundaries of fitness function, respectively.

To improve the search process of PSO, the inertia weight ($w$) is decreased from 0.9 to 0.2 linearly with iteration number. Also, $c_1$ and $c_2$, in PSO are adapted with time from 0 to 2, just like $k_1$ and $k_2$. It means that in the first iterations the exploration is considered and in the final iterations exploitation is emphasized.

IPO, PSO, and GA are evaluated on three sets of benchmark functions which are discussed in the next subsections.

### 4.2 Unimodal Functions

Functions $f_1$ to $f_7$ are unimodal functions. While there are specific methods designed for optimizing these functions, in these cases, we are more interested in the convergence rate of algorithm. The average of final best fitnesses obtained by GA, PSO, and IPO over 30 runs are presented in Table 1.

| Function | GA | PSO | IPO |
|---|---|---|---|
| F1 | 468.94 | $3.72 \times 10^{-10}$ | $2.64 \times 10^{-20}$ |
| F2 | 0.31 | 2.00 | $2.32 \times 10^{-16}$ |
| F3 | $1.50 \times 10^{4}$ | 348.77 | $1.25 \times 10^{-4}$ |
| F4 | 63.33 | 2.73 | $2.99 \times 10^{-3}$ |
| F5 | 952.36 | 167.42 | 29.93 |
| F6 | 5 175.20 | 0 | 0 |
| F7 | 0.90 | $2.05 \times 10^{-2}$ | $1.27 \times 10^{-2}$ |

Table 1. The average of final best fitnesses for 30 runs of minimizing benchmark functions 1 to 7, number of iterations = 1 000

Table 1 shows that IPO generates better results than GA and PSO in all cases, which means that the IPO is more effective and powerful than GA and PSO in unimodal functions. To compare their relative performance against the number of iterations, Figure 5 is presented. This figure presents the optimization process of function $f_4$ for three algorithms, from which we can conclude the higher convergence rate of IPO than GA and PSO.

### 4.3 Multimodal Functions with Many Local Minima

In multimodal functions with many local minima ($f_8$ to $f_{13}$), the number of local minima exponentially increases by increasing the function dimensions. Hence, these functions are the most difficult to optimize. In multimodal functions with many local minima, final results that show the ability of algorithm in escaping from local minima are more important. Table 2 shows the obtained results by GA, PSO, and IPO over 30 runs in minimizing $f_8$ to $f_{13}$. These values are the same as Table 1.

| Function | GA | PSO | IPO |
|---|---|---|---|
| F8 | −9 406.14 | −9 102.19 | −10 403.51 |
| F9 | 10.42 | 44.45 | 3.59 |
| F10 | 5.86 | $6.16 \times 10^{-6}$ | 0.74 |
| F11 | 63.85 | $9.93 \times 10^{-5}$ | $7.47 \times 10^{-3}$ |
| F12 | 31.96 | $3.46 \times 10^{-3}$ | $6.22 \times 10^{-2}$ |
| F13 | 8.31 | $3.66 \times 10^{-3}$ | $2.93 \times 10^{-3}$ |

Table 2. The average of final best fitnesses for 30 runs of minimizing benchmark functions 8 to 13, number of iterations = 1 000

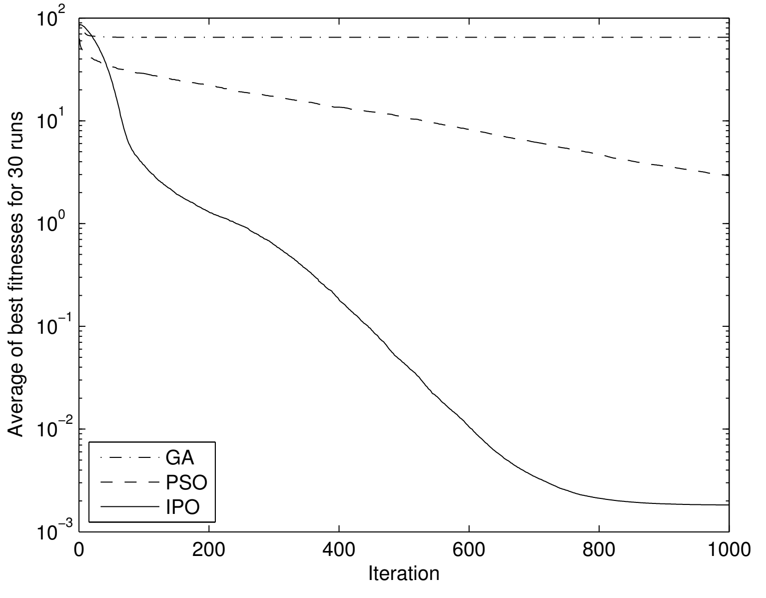



Figure 5. Performance of GA, PSO, and IPO on function $f_4$. Values of 0.72, 2.76, 72.47, 188.51, 0.04, and 0.82 are used for $c_1$, $c_2$, $shift_1$, $shift_2$, $scale_1$, and $scale_2$ constants in IPO algorithm, respectively

From Table 2 we can see that IPO outperforms GA for all functions, also it could be seen that IPO obtains better results than PSO on $f_8$, $f_9$, and $f_{13}$. This shows that performance of the proposed IPO could be favorable when compared to other evolutionary and swarm intelligence based methods.

Figure 6 visualizes the process of optimization by three algorithms for $f_{13}$. In this figure the ability of IPO in escaping from trapping in local minima and its high convergence rate can be seen.

### 4.4 Multimodal Functions with Only a Few Local Minima

Functions $f_{14}$ to $f_{23}$ have small dimensions and only a few local minima. Table 3 displays the results of GA, PSO, and IPO in optimizing these functions. These results are as same as Tables 1 and 2.

For this type of functions, results generated by IPO are better or similar to other algorithms. Figure 7 shows the performance of three algorithms in optimization of $f_{15}$.

This figure shows that IPO also has a high convergence rate in optimizing multimodal low dimensional functions.

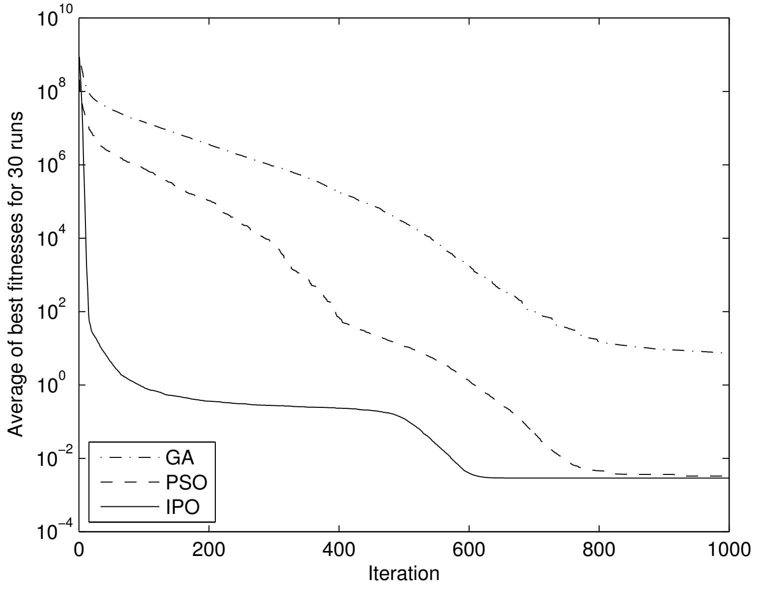



Figure 6. Performance of GA, PSO, and IPO on function $f_{13}$. Values of 0.45, 2.61, 475.55, 602.85, 0.04, and 0.05 are used for $c_1$, $c_2$, $shift_1$, $shift_2$, $scale_1$, and $scale_2$ constants in IPO algorithm, respectively

### 4.5 Comparative Results for Local-IPO and Lbest PSO

Regarding Subsection 3.2, IPO has the capability of confident balance between local and global search by introducing the local-IPO (like lbest PSO). In local-IPO, the calculation of angle and acceleration is executed only for a pre-specified neighbourhood and with respect to the best ball in this area of the search space. This idea could lead to a good local search and refining the final solutions. To show the performance of local-best IPO, comparative results with lbest PSO are presented in Table 4.

In this Table, the obtained overall best fitness for 30 runs of local-IPO and lbest PSO on the first two functions of each function-group functions are reported. The results of Table 4 show that in all cases the performances of the proposed IPO are better than or comparable to lbest PSO.

## 5 CONCLUSION AND FUTURE WORKS

Increase in the number of dimensions of scientific problems attracts more efforts for finding new solutions. In this paper, a new heuristic algorithm, called Inclined Planes system Optimization (IPO) is introduced. IPO is based on the dynamics of

| Function | GA | PSO | IPO |
|---|---|---|---|
| F14 | 4.12 | 0.9980 | 0.9980 |
| F15 | $2.44 \times 10^{-3}$ | $1.23 \times 10^{-3}$ | $4.33 \times 10^{-4}$ |
| F16 | −1.0316 | −1.0316 | −1.0316 |
| F17 | 0.3979 | 0.3979 | 0.3979 |
| F18 | 3.90 | 3.00 | 3.00 |
| F19 | −3.8628 | −3.8628 | −3.8628 |
| F20 | −3.2744 | −3.2683 | −3.2881 |
| F21 | −5.2121 | −6.0637 | −9.8164 |
| F22 | −6.8356 | −8.1109 | −10.0513 |
| F23 | −4.6898 | −9.7793 | −10.0254 |

Table 3. The average of final best fitnesses for 30 runs of minimizing benchmark functions 14 to 23, number of iterations = 500

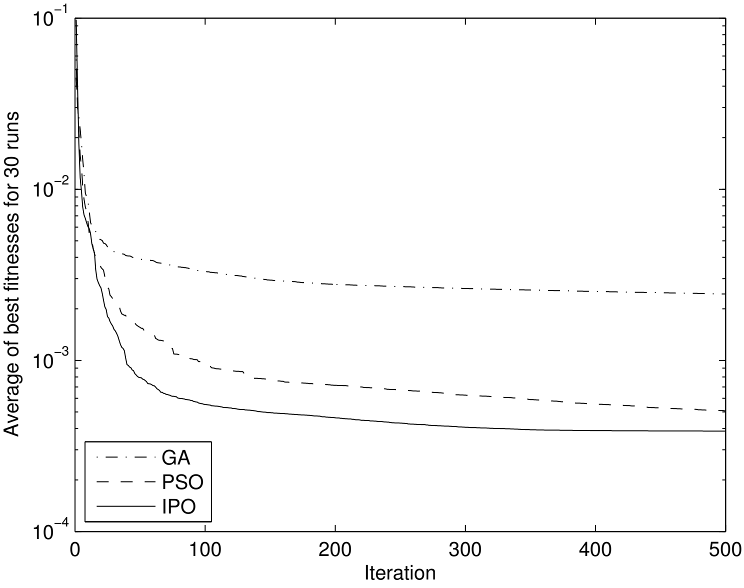



Figure 7. Performance of GA, PSO, and IPO on function $f_{15}$. Values of 0.2, 0.38, 1.27, 332.67, 0.01, and 0.01 are used for $c_1$, $c_2$, $shift_1$, $shift_2$, $scale_1$, and $scale_2$ constants in IPO algorithm, respectively

| Algorithm | $f_2$ | $f_4$ | $f_8$ | $f_9$ | $f_{14}$ | $f_{15}$ |
|---|---|---|---|---|---|---|
| local-IPO | $1.08 \times 10^{-20}$ | $1.29 \times 10^{-6}$ | −11 498.59 | 6.11 | 0.9980 | $4.07 \times 10^{-4}$ |
| lbest-PSO | 9.01 | 17.93 | −5 812.85 | 158.20 | 0.9980 | $6.48 \times 10^{-4}$ |

Table 4. Results for 30 runs of minimizing some of the benchmark functions for local-IPO and lbest PSO

sliding motion along a frictionless inclined plane. In the proposed method, Newton's second law and estimated inclined planes are used to evaluate the acceleration of particles towards better positions in problem search space.

In order to show the effectiveness of our algorithm, we compared its performance over 23 benchmark functions with GA and PSO. The results have shown that IPO is quite effective in optimizing these benchmark functions.

Although IPO has shown that it can be so powerful in solving optimization problems, with a considerable scalability and convergence rate, it is still in its infancy and can be improved by further research and development efforts. In this article, convergence of IPO has not proven theoretically, and this can be done as future works. Also, the performances of the proposed method against practical engineering problems (e.g. pattern recognition, and image processing) are important topics for future research.

## Acknowledgement

The authors would like to thank Dr. Marjan Fooladi for her valuable editorial assistances.

## REFERENCES

[1] Holland, J. H.: Adaption in Natural and Artificial Systems. The University of Michigan Press, 1975.

[2] Kennedy, J.—Eberhart, R.: Particle Swarm Optimization. Proceedings of the IEEE International Conference on Neural Networks, 1995, Vol. 4, pp. 1942–1948. doi: 10.1109/ICNN.1995.488968.

[3] Dorigo, M.: Optimization, Learning and Natural Algorithms. Ph.D. Thesis, Politecnico di Milano, Italy, 1992.

[4] Formato, R. A.: Central Force Optimization: A New Metaheuristic with Applications in Applied Electromagnetics. Progress in Electromagnetics Research, Vol. 77, 2007, pp. 425–491.

[5] Zarandi, M. H. F.—Hemmati, A.—Davari, S.: The Multi-Depot Capacitated Location-Routing Problem with Fuzzy Travel Times. Expert Systems with Applications, Vol. 38, 2011, No. 8, pp. 10075–10084. doi: 10.1016/j.eswa.2011.02.006. Available on: `http://www.sciencedirect.com/science/article/pii/S0957417411002120`.

[6] Mitra, S.—Banka, H.: Multi-Objective Evolutionary Biclustering of Gene Expression Data. Pattern Recognition, Vol. 39, 2006, No. 12, pp. 2464–2477. doi: 10.1016/j.patcog.2006.03.003. Available on: `http://www.sciencedirect.com/science/article/pii/S0031320306000872`.

[7] Zahiri, S.-H.: Swarm Intelligence and Fuzzy Systems. Nova Science Publishers, 2010.

[8] Darby, S.—Mortimer-Jones, T. V.—Johnston, R. L.—Roberts, C.: Theoretical Study of Cu–Au Nanoalloy Clusters Using a Genetic Algorithm. The Journal of Chemical Physics, Vol. 116, 2002, No. 4, pp. 1536–1550. doi: 10.1063/ 1.1429658. Available on: `http://scitation.aip.org/content/aip/journal/jcp/116/4/10.1063/1.1429658`.

[9] Coello Coello, C. A.—Luna, E. H.—Aguirre, A. H.: Use of Particle Swarm Optimization to Design Combinational Logic Circuits. In: Tyrrell, A., Haddow, P., Torresen, J. (Eds.): Evolvable Systems: From Biology to Hardware. Springer Berlin Heidelberg, Lecture Notes in Computer Science, Vol. 2606, 2003, pp. 398–409. doi: 10.1007/3-540-36553-2_36.

[10] Wolpert, D. H.—Macready, W. G.: No Free Lunch Theorems for Optimization. IEEE Transactions on Evolutionary Computation, Vol. 1, 1997, No. 1, pp. 67–82.

[11] Rashedi, E.—Nezamabadi-pour, H.—Saryazdi, S.: GSA: A Gravitational Search Algorithm. Information Sciences, Vol. 179, 2009, No. 13, pp. 2232–2248. doi: 10.1016/j.ins.2009.03.004. Available on: `http://www.sciencedirect.com/science/article/pii/S0020025509001200`.

[12] Farmer, J.—Packard, N. H.—Perelson, A. S.: The Immune System, Adaptation, and Machine Learning. Physica D: Nonlinear Phenomena, Vol. 22, 1986, No. 1–3, pp. 187–204. doi: 10.1016/0167-2789(86)90240-X. Available on: `http://www.sciencedirect.com/science/article/pii/016727898690240X`.

[13] Kirkpatrick, S.—Gelatt, C. D.—Vecchi, M. P.: Optimization by Simulated Annealing. Science, Vol. 220, 1983, No. 4598, pp. 671–680. doi: 10.1126/science.220.4598.671. Available on: `http://www.sciencemag.org/content/220/4598/671.full.pdf`, `http://www.sciencemag.org/content/220/4598/671.abstract`.

[14] Geem, Z. W.—Kim, J. H.—Loganathan, G.: A New Heuristic Optimization Algorithm: Harmony Search. SIMULATION, Vol. 76, 2001, No. 2, pp. 60–68. doi: 10.1177/003754970107600201. Available on: `http://sim.sagepub.com/content/76/2/60.full.pdf+html`, `http://sim.sagepub.com/content/76/2/60.abstract.html`.

[15] Karaboga, D.—Basturk, B.: A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm. Journal of Global Optimization, Vol. 39, 2007, No. 3, pp. 459–471. doi: 10.1007/s10898-007-9149-x.

[16] Yang, X.-S.: A New Metaheuristic Bat-Inspired Algorithm. In: González, J., Pelta, D., Cruz, C., Terrazas, G., Krasnogor, N. (Eds.): Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), Springer Berlin Heidelberg, Studies in Computational Intelligence, Vol. 284, 2010, pp. 65–74. doi: 10.1007/978-3-642-12538-6_6.

[17] Duman, E.—Uysal, M.—Alkaya, A. F.: Migrating Birds Optimization: A New Metaheuristic Approach and Its Performance on Quadratic Assignment Problem. Information Sciences, Vol. 217, 2012, pp. 65–77. doi: 10.1016/j.ins.2012.06.032. Available on: `http://www.sciencedirect.com/science/article/pii/S0020025512004483`.

[18] Tamura, K.—Yasuda, K.: Primary Study of Spiral Dynamics Inspired Optimization. IEEJ Transactions on Electrical and Electronic Engineering, Vol. 6, 2011, No. S1, pp. S98–S100. doi: 10.1002/tee.20628.

[19] Gigerenzer, G.—Todd, P. M.—Group, A. R. et al.: Simple Heuristics That Make Us Smart. Oxford University Press New York, 1999.

[20] Voges, K. E.—Pope, N.—Brown, M. R.: Cluster Analysis of Marketing Data Examining On-Line Shopping Orientation: A Comparison of K-Means and Rough Clustering Approaches. Heuristics and Optimization for Knowledge Discovery, 2002, pp. 207–224.

[21] Gigerenzer, G.—Gaissmaier, W.: Heuristic Decision Making. Annual Review of Psychology, Vol. 62, 2011, No. 1, pp. 451–482. doi: 10.1146/annurev-psych-120709-145346. Available on: `http://www.annualreviews.org/doi/pdf/10.1146/annurev-psych-120709-145346`, `http://www.annualreviews.org/doi/abs/10.1146/annurev-psych-120709-145346`.

[22] Tang, K.-S.—Man, K.—Kwong, S.—He, Q.: Genetic Algorithms and Their Applications. Signal Processing Magazine, IEEE, Vol. 13, 1996, No. 6, pp. 22–37.

[23] Silva, A.—Neves, A.—Costa, E.: An Empirical Comparison of Particle Swarm and Predator Prey Optimisation. In: O'Neill, M., Sutcliffe, R., Ryan, C., Eaton, M., Griffith, N. (Eds.): Artificial Intelligence and Cognitive Science, Springer Berlin Heidelberg, Lecture Notes in Computer Science, Vol. 2464, 2002, pp. 103–110. doi: 10.1007/3-540-45750-X_13.

[24] Eiben, A. E.—Schippers, C.: On Evolutionary Exploration and Exploitation. Fundamenta Informaticae, Vol. 35, 1998, No. 1, pp. 35–50.

[25] Yao, X.—Liu, Y.—Lin, G.: Evolutionary Programming Made Faster. IEEE Transactions on Evolutionary Computation, Vol. 3, 1999, No. 2, pp. 82–102.

[26] Mathworks(TM), MATLAB R2011a. Available on: `http://www.mathworks.com`, 2011.

**Mohammad Hamed Mozaffari** graduated from the Sadjad University of Technology, Mashhad, Iran in 2010 with his B.Sc. degree in electronics with a specialty in control systems and received his M.Sc. degree in electronics engineering from the University of Birjand, Iran in 2012. He has been researching in the Department of Electrical Engineering and Computer Science, University of Ottawa, Canada as a Ph.D. candidate since 2015. His current research interests include medical image processing, 3D visualization, computer vision, pattern recognition and soft computing.

**Hamed Abdy** received his B.Sc. and M.Sc. degrees in electronics engineering from Zanjan University, Iran, in 2010 and from University of Birjand, Iran, in 2012, respectively. Currently he is a Ph.D. candidate in nano-electronics at University of Tehran, Iran. His main research interests include perovskite solar cells, solar cells, micro and nano-electronics devices, evolutionary and swarm intelligence algorithms, and pattern recognition.

**Seyed-Hamid Zahiri** received his B.Sc., M.Sc. and Ph.D. degrees in electronics engineering from Sharif University of Technology, Tehran, Iran, Tarbiat Modarres University, Tehran, Iran and Mashhad Ferdowsi University, Mashhad, Iran, in 1993, 1995, and 2005, respectively. Currently, he is Professor with the Department of Electronics Engineering, University of Birjand, Iran. His research interests include pattern recognition, evolutionary algorithms, swarm intelligence algorithms, and soft computing.