An Improved Population Migration Algorithm Introducing the Local Search Mechanism of the Leap-Frog Algorithm and Crossover Operator

Yanqing Zhang, Xueying Liu*

Department of Mathematics, Inner Mongolia University of Technology, Hohhot, China

Abstract

The population migration algorithm (PMA) is a simulation of a population of the intelligent algorithm. Given the prematurity and low precision of PMA, this paper introduces a local search mechanism of the leap-frog algorithm and crossover operator to improve the PMA search speed and global convergence properties. The typical test function verifies the improved algorithm through its performance. Compared with the improved population migration and other intelligent algorithms, the result shows that the convergence rate of the improved PMA is very high and its convergence is proved.

Citation: Zhang Y, Liu X (2013) An Improved Population Migration Algorithm Introducing the Local Search Mechanism of the Leap-Frog Algorithm and Crossover Operator. PLoS ONE 8(2): e56652. doi:10.1371/journal.pone.0056652

Editor: Vladimir Brusic, Dana-Farber Cancer Institute, United States of America

Received: November 2, 2012; **Accepted:** January 11, 2013; **Published:** February 27, 2013

Copyright: © 2013 Zhang, Liu. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Funding: This work was supported by the Scientific Research Project of the Higher Education Institutions of Inner Mongolia Autonomous Region No. NJZY12070. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: liuxy@imut.edu.cn

Introduction

The population migration algorithm (PMA) was proposed by Zhou et al in 2003 [1–2]. The PMA is a simulated population migration theory global optimization algorithm. The PMA is also a simulated mechanism that involves population along with economic center transfer and population pressure diffusion in the field. In other words, people relocate to a preferential region that has a high level of economic development and employment opportunities. When the preferential region has relative overpopulation, the population pressure constantly increases. When population pressure exceeds a certain limit, people will move to a more suitable preferential region. Population pressure makes the PMA engage in a better regional search. To a certain extent, population proliferation can prevent the PMA from falling into a local optimal solution. The whole algorithm shows alternate features that localized search and scatter search have in the search process.

In 2004, Xu gave an improved PMA (IPMA), which can be summarized into three forms: population flow, population migration, and population proliferation [3]. Population flow states that the population residing in the region is spontaneous and the overall flow is uncertain. Population migration is based on the mechanism across a wide range of the selection movement, in which people flow to the rich place. When the population pressure is very high, population proliferation is a selective movement from a preferential area to a no preferential area. It also reflects the people's pioneering spirit.

Recent, various kinds of intelligent algorithms have been improved by many researchers. Wang et al improved the PMA with the steepest descent operator and the Gauss mutation [4]. Ouyang et al introduced the simplex in the PMA to improve the search performance of the algorithm [5]. Guo et al increased

performance of the population migration by improving the migrations' efficiency [6]. Karaboga improved the bee swarm [7]. Li et al introduced extremal optimization to improve the performance of the shuffled frog leaping algorithm (SFLA) [8]. However, the results of all these studies show low accuracy or fall into the local optimal solution. To further improve the local search ability of the PMA and speed up convergence, we introduce a local search mechanism of frog leaping and crossover. This paper is improved and uses a number of classic functions for the simulation to show that the algorithm is effective and feasible.

Population Migration Algorithm

Consider the following unconstrained single-objective optimization problem:

$$\min f(X) \quad (1)$$

$$s. t. X \in S$$

In which $f: S \rightarrow R^1$ is a real valued mapping, $X \in R^n$, $S = \prod_{i=1}^n [a_i, b_i]$ is a search space, $a_i < b_i$. We assume that problem (1) has a constant solution, namely, global optimal solution exists and the set H of the global optimal solution is not empty. Seeking the most value problem can be transformed into the form of problem (1). In the algorithm, X is an individual and stands for the population and various kinds of information. $X^i = (X_1^i, X_2^i, \dots, X_n^i)$ is the i th individual in the solution space, $X^i \in R^n$; X_j^i is the i th individual of the j th weight; $\delta^i \in R^n$, δ_j^i is the j th weight of the i th region's radius δ^i , in which, $\delta_j^i > 0$, $i = 1, 2, \dots, N$, $j = 1, 2, \dots, n$; and N

stands for population size. Target function $f(x)$ corresponds to the population attraction. The optimal solution (a local optimal solution) of problem (1) corresponds to the most attractive region (the preferential). The ability of the PMA to escape from the local optimal solution corresponds to the ability. Due to population pressure increase, the population moves out of the preferential area. The rise of algorithm or mountain climbing made the populations move into the preferential area. Local random search of algorithm corresponds to the population flow. The method by which the algorithm selects the approximate solution corresponds to the population migration. Escaping from the local optimal strategies corresponds to the population proliferation.

Frog Leaping Algorithm of Local Search

SFLA is proposed, as proved by Eusuff in 2003 [9]. SFLA is based on the heuristic and collaborative search of species. The implementation of the algorithm simulates the element evolution behavior of nature.

The implementation process of SFLA simulates the feeding behavior of a group of frogs in a wetland [10]. In the feeding process, every frog can be seen as the carrier of ideas and information. Frogs can communicate through the exchange of information, which can improve the knowledge of others. Every frog represents a solution to the optimization problem (1). Wetland represents the solution space. In the algorithm implemental stage, there are F frogs divided into N groups. Each group has l frogs. Different groups have different ideas and information. In a group, frogs follow the element evolution strategy in the solution space for local depth search and internal communication.

$$S = \text{rand}() * (P_{bi} - P_{wi}) \quad (2)$$

$$P_{1wi} = P_{wi} + S \quad (3)$$

In the above formulas, $\text{rand}()$ is a random function that randomly generates the value in the range $[0,1]$, S is a step of the frog leaping, P_{wi} stands for the worst frog of the i th group, P_{bi} stands for the best frog of the i th group, and P_{1wi} stands for an update of the worst frog of the i th group. If the adaptive value of the updated worst frog is more than that of the original worst frog in the same group, the updated worst frog will replace the original worst frog. Otherwise, a frog will be randomly generated in the group instead of the original worst frog. The updated process is repeated until the predetermined number of local search L_s . When all groups have completed the depth of local search, all the frogs of the groups are again mixed and divided into N groups. We continue the local search in the groups until the termination rule holds true, and then we stop.

Crossover Operators

By two leaner combinations, we can obtain two new individuals. Arithmetic crossover operation is often used to express individuals by floating individual coding. Hypothesis: The two parents are $X_a = (X_1^1, X_2^1, \dots, X_n^1)$ and $X_b = (X_1^2, X_2^2, \dots, X_n^2)$.

Then, n random numbers are randomly generated, in which $a_i \in (0,1)$, $i = 1, 2, \dots, n$. After hybrid implementation, two new progenies are generated:

$$Y_a = (Y_1^1, Y_2^1, \dots, Y_n^1), Y_b = (Y_1^2, Y_2^2, \dots, Y_n^2)$$

$$Y_i^1 = a_i X_i^1 + (1 - a_i) X_i^2, Y_i^2 = a_i X_i^2 + (1 - a_i) X_i^1, i = 1, 2, \dots, n.$$

Methods

In the PMA, the population flow includes extensive information exchange. In practice, population information exchange decides the direction of the migration under certain conditions, so we should give full attention to population flow. When under pressure, population migration is the random population flow in the range of the population pressure, which does not have a significant role in further improving the algorithm.

The local search mechanism of frog-leaping algorithm and the crossover operator are introduced into the PMA in the process of population flow, which can well improve the original search results. IMPA organically made the frog-leaping combine with the crossover operator in the PMA. The process of the algorithm is as follows:

Initialization

Input the population size N , the initial radius $\delta(0)$, the vigilance parameter of population pressure $\alpha(0)$, the number of the population flow l , the number of local search L_s , and the maximum number of iterations T . We randomly generate N individuals in the search space S , $X^1, X^2, \dots, X^N$. The i th regional center $center^i = X^i$ determines the upper and lower $center^i \pm \delta^i$ bounds of the i th region, in which $\delta_j^i > 0$, $i = 1, 2, \dots, N, j = 1, \dots, n$. The extraction method δ_j^i makes δ^i equal, so we cancel the superscript of δ^i in the following steps. We obtain an initial search space $\Omega(0) = \bigcup_{i=1}^N B(X^i(0), \delta(0))$, which stands for the initial residential area. The set $B(X, \delta)$ is taken as the sphere, in which X is the center and r is the radius. The following variables appearing that way are the same. $f(X^0) = \max_{1 \leq i \leq N} (X^i)$ and $X_{best}(0) = X^0$. We set iteration counter $t = 0$.

Evolutionary

1. Preparation: $\eta(t) = \delta(t)$

2. Population flow: In every $B(X^i(t), \eta(t))$, we randomly generate l_i individuals, so we can obtain the group $Y(t)$ in $\Omega(t)$, which has N individuals.

3. Population migration

We next introduce the local search mechanism of frog-leaping algorithm and crossover operator. At this time, the role of population has been transformed to the frog. The search area is seen as the frog's wetland. The formed N local areas are seen as the N groups. The individual of every local area is seen as the frog.

(a) We choose N best frogs in $\mathcal{Y}(t)$, which form the intermediate frog group $Y_{best}(t) = \{Y_{best}^{(1)}(t), Y_{best}^{(2)}(t), \dots, Y_{best}^{(N)}(t)\}$.

(b) Forming N regions: $\{B(Y_{best}^{(i)}(t), \eta(t)), i = 1, 2, \dots, N\}$.

(c) In every group, we produce a frog body flow. In other words, l_i individuals are randomly generated in $B(Y_{best}^{(i)}(t), \eta(t))$. l_i and the adaptive value of $Y_{best}^{(i)}$ are proportional.

(d) We set the group counter $iN = 1$, which is compared with the total of groups N ; the local evolution counter $im = 1$, which is compared with the number of local search L_s .

(e) In the iN th group, we calculate the adaptive value of every frog in the group and arrange them in descending order. We can obtain the best solution P_{bi} and the worst solution P_{wi} .

(f) $Ph_{wi} = P_{wi}$; according to the following four formulas, we can update and improve the position of the worst frog in the group.

$$S = \text{rand}() * (P_{bi} - P_{wi}) \quad (4)$$

$$P1_{wi} = P_{wi} + S, \quad -S_{\max} \leq S \leq S_{\max} \quad (5)$$

$$e = \text{rand}() \quad (6)$$

$$\text{new}P_{wi} = e * P1_{wi} + (1 - e) * Ph_{wi} \quad (7)$$

(g) If step (f) improved the position of the worst frog, the adaptive value of $\text{new}P_{wi}$ is more than the P_{wi} 's, the position of P_{wi} will be replaced by the position of $\text{new}P_{wi}$. That is, $P_{wi} = \text{new}P_{wi}$. Otherwise, we randomly generate a position of a frog instead of the worst frog in the group.

(h) If $im < L_s$, then $im = im + 1$ and turn to step (f). Otherwise, we report the best frog $Z_{best}^{(i)}(t)$ and turn to step (i).

(i) If $iN < N$, then $iN = iN + 1$ and we turn to step (e). Otherwise, we turn to step (j).

(j) Contraction of preferential region: $\eta(t) = (1 - \Delta)\eta(t)$, ($0 < \Delta < 1$).

(k) If $\eta(t) > \alpha(t)$ (not more than the population pressure alert), we continue the frog flow and turn to step (c). Otherwise, we turn to step (l).

(l) We report N the best frogs: $Z_{best}(t) = \{Z_{best}^1(t), Z_{best}^2(t), \dots, Z_{best}^N(t)\}$.

4. Population proliferation:

(a) We keep the best individual in the Z_{best} , which is credited as Z_{best}^0 .

(b) In the $\Omega(t)$, we randomly sample $N-1$ new individuals instead of the $N-1$ individuals of $\{Z_{best}(t) \setminus Z_{best}^0\}$.

(c) We define a new generation of population

$$X(t+1) = \{X^1(t+1), X^2(t+1), \dots, X^{N-1}(t+1), Z_{best}^0(t)\}.$$

Termination of Inspection

If the new generation of population $X(t+1)$ contains the eligible solution, we stop and output the best solution in $X(t+1)$. Otherwise, we narrow $\delta(t)$ and $\alpha(t)$ properly and set $t=t+1$. If the maximum numbers of iterations for the algorithm are not met, we turn to the second step.

Convergence Analysis

This paper uses the axiom model to analyze and prove the convergence of IMPA.

Definition 1 Optimization variables X have limited-length characters and have the form $A = a_1 a_2 \dots a_n \Rightarrow X = (x_1, x_2, \dots, x_n)$. n is called the code string length of X , A is called the code of X , and X is called the decode of the character A . a_i is considered a genetic

gene. All possible values of a_i are called allelic gene. A is a chromosome made up of n genes [11].

Definition 2 (Individual Space) Γ is an allelic gene. n is given the code string length. The set $H_I = \{a_1 a_2 \dots a_n, a_i \in \Gamma, i = 1, 2, \dots, n\}$ is called individual, in which $\forall N$ is a natural number and $H^N = \underbrace{H * H * \dots * H}_N$ is the N order of population space [11].

Definition 3 (Satisfaction Set) If $\forall X \in B, Y \notin B$, then $f(X) > f(Y)$. $B(X)$ is called the satisfaction set. The number of individuals in the set $B(X)$ is used for $|B(X)|$. $\bar{X} \in H_n^N$ is any of the population and X_b is the best individual in the set $\bar{X}$. The satisfaction set induced by X_b is called satisfaction set of the population $\bar{X}$, which is marked $B_b(\bar{X}) = \{X \in H_I, f(X) \geq f(\bar{X})\}$. All of the satisfaction sets' intersections are formed with the global optimal solution B^* [11].

Definition 4 (Selection Operator) The selection operator S is a random mapping, $S: H^N \rightarrow H^N$. It meets the following two conditions. (1) For $\forall \bar{X} \in H_n^N$, we can obtain $[S\bar{X}]_i \in \bar{X}, i = 1, 2, \dots, N$, in which $[S\bar{X}]_i$ stands for the individual in the $S\bar{X}$; (2) Any population $\bar{X}$ that meets $|B(\bar{X})| < N$ has $P\{|B(S\bar{X})| + (N - M) > |B(\bar{X})|\} > 0$. The first condition states that selected individuals should be in the selected population. The second condition states that the selection operator may increase the number of optimal individuals in the population, in addition to variations of the individuals' numbers between selected population and original population [11].

Definition 5 (Reproduction Operator) Reproduction operator R is a random mapping, $R: H^N \rightarrow H^N$. If any population $\bar{X} \in H_n^N$, which meets $P\{R(\bar{X}) \cap B_b(\bar{X}) \neq \Phi\} > 0$, the new population $R(\bar{X})$ which $\bar{X}$ becomes under the action of the operator R , may contain more satisfied individuals than the original population [11].

Definition 6 (Mutation Operator) Mutation operator M is a random mapping: $M: H_I^N \rightarrow H_I^N$. $\forall \bar{X} \in H_n^N$ meets $P\{M(\bar{X}) \cap B^* \neq \Phi\} > 0$. Mutation operator is the reproduction operator that has a much better nature. If the population contains satisfied individuals, then after mutation, the population should contain satisfied individuals. If the population does not have satisfied individuals, then after mutation, the population may contain satisfied individuals [11].

In improved population migration, extended operator $L_0: H_I^N \rightarrow H_I^N$ meets the following expressions:

$$L_0(X) = (\underbrace{X^{(1)}, \dots, X^{(1)}}_I, \dots, \underbrace{X^{(N)}, \dots, X^{(N)}}_I),$$

$$H_I^N = \underbrace{H_I * H_I * \dots * H_I}_N$$

and

$$\forall X = (X^{(1)}, X^{(2)}, \dots, X^{(N)}) \in H_I^N.$$

$$\text{Uniform mutation operator } M_t: H_I^N \rightarrow H_I^N,$$

$$\forall Y = (Y^{(1)}, Y^{(2)}, \dots, Y^{(N)}) \in H_n^N, M_t(Y) = (M_t(Y^{(1)}), M_t(Y^{(2)}), \dots, M_t(Y^{(N)})).$$

Therefore, we can use the operator $M_t L_0: H_I^N \rightarrow H_I^N$ to stand for the process of population flow. The selection operator S_1 stands

for the process of step (a) in population migration, where we chose N , the best individuals from population $\mathcal{I}(t)$. The reproduction operator R_1 stands for the progress from step (b) to the process of final output $\mathcal{Z}_{best}(t)$ in step (l). The reproduction operator R_2 stands for the process from population proliferation to finally generate $X(t+1)$. Therefore, the IPMA can be expressed as:

$$X(t+1) = R_2 * R_1 * S_1 * M_t * L_0(X(t)), t=0,1,2,\dots \quad (8)$$

$$Y(t) = M_t * L(X(t)), S = S_1, R = M_t * L_0 * R_2 * R_1, t=1,2,\dots \quad (9)$$

After both sides have the same function of operator $M_t * L_0$ for (8), we can find that $Y(t+1) = R * S(Y(t))$. The general simulated evolution algorithm can be composed of a series of selection operators $\{S_t\}$ and reproduction operators $\{R_t\}$ to be represented abstractly as follows: $X(t+1) = R_t * S_t(X(t))$.

Lemma 1 In the IPMA, S is a selection operator and R is a reproduction operator. The proof can be obtained from reference [11].

Theorem 1 The IPMA is a simulated evolutionary algorithm.

Lemma 2 In the IPMA, the characteristic number of the preferred selection operator, the selection pressure of the operator $S: P_S = 1$, the selection intensity $\alpha_S = 1$. The IPMA just takes the preferred selection operator. According to reference [3], the selection pressure and the selection intensity are defined to calculate the characteristic number of the operator S . The proof can be obtained from reference [11].

Lemma 3 In the IPMA, the characteristic number of the reproduction operator, the operator $R: H_I^N \rightarrow H_I^{N+1}$, then the absorption and scattering rate of the reproduction meet the following estimates: $A_E \geq Nl\rho\beta_t(1-\rho(C_{t+1})^{n+1})^{l-1}(\varepsilon)[1-\beta_t(\frac{\varepsilon}{C_{t+1}})]^1$, $S_R^k = 0$. Where, $\varepsilon = C(\varepsilon)\delta$ (δ is the radius of the search space) is the radius of the set of optimal solution B and β_t is a conversion coefficient. If the set B includes a division region in the $\Omega(+1)$, then $\beta_t = 1$; otherwise, β_t is the rate of volume of the intersection of the set B and a division region. Thus, we can obtain $0 \leq \beta_t \leq 1$. The proof can be obtained from reference [11].

Lemma 4 If the abstract simulated evolutionary algorithm (SEA) satisfies the following conditions: (1) the pressure of selection operator $\{S_t\}$ is uniformly bounded; (2) the absorption of reproduction operator $\{R_t\}$: $\{A_{e(t)}\}$ meets $\sum_{t=1}^{\infty} E(t) = \infty$; (3) the strength of selection $\{\alpha_{s(t)}\}$, absorption $\{A_{e(t)}\}$, and scattering $\{S_{e(t)}^{m_0}\}$ content are as follows: $\lim_{t \rightarrow \infty} [1 - (1 - S_{R(t)}^{m_0})\alpha_{s(t)}] / A_{R(t)} = 0$. Then, an abstract and a general SEA will probably weakly approach the optimal solution set. The proof can be obtained from reference [11].

Theorem 2 (Convergence of IPMA) If the IPMA adopts the selection operator family $\{S_t\}$, which comprise all the preferred selection operators, reproduction operators family $\{R_t\}$, in which $Y(t+1) = R * S(Y(t))$. The following conditions are thus met: (1); (2) $\beta_t = O(1/t)$ ($\beta_t = O((C_{t+1})^n)$).

Then, to prove that the IPMA will probably weakly approach the solution set, the following steps are performed.

Proving. Known by lemma 2: $P_{s(t)} = 1, \alpha_{s(t)} = 1$; Known by lemma 3: $A_E \geq Nl\rho\beta_t(1-\rho(C_{t+1})^{n+1})^{N-1}(C(\varepsilon)^n)[1-\beta_t(\frac{C(\varepsilon)}{C_{t+1}})]^{l-1} = O(1/t)$, $S_R^k = 0$; when ε is given, $C(\varepsilon)$ is also given. By condition (1), we obtain $\lim_{t \rightarrow \infty} \rho((C_{t+1})^{n+1})^{N-1} = 1$. By condition (2), we obtain $\beta_t = O((C_{t+1})^n)$, then $\lim_{t \rightarrow \infty} [1 - \beta_t(C(\varepsilon)/C_{t+1})^n]^{l-1} \neq 0$. We have

$\beta_t = O(1/t)$ and lemma 3, then $A_E \geq Nl\rho\beta_t(1-\rho(C_{t+1})^{n+1})^{N-1}(C(\varepsilon)^n)[1-\beta_t(\frac{C(\varepsilon)}{C_{t+1}})]^{l-1} = O(1/t)$. We know the divergence of the harmonic series, so $\sum_{t=1}^{\infty} A_{R(t)} = \infty$. According to lemma 3: $S_R^1 = 0$.

According to lemma 2: $\alpha_{s(t)} = 1$, so $\lim_{t \rightarrow \infty} (1 - (1 - S_{R(t)}^1)\alpha_{s(t)}) / A_{R(t)} = 0$.

The three conditions of lemma 3 are met. The IPMA probably weakly approaches the global optimal solution.

Results and Discussion

Simulation Examples and Experimental Parameter Setting

Setting

To verify the performance of the IPMA, this paper performs an experiment. The IPMA is compared with the basic population algorithm and other improved algorithms. In the simulation experiment, we choose 10 classic test functions to test the algorithm. According to the performance of the functions, they are divided into a single minimum (single) and a number of local minima (multimodal) into two categories. The following lists the 10 functions' definition, variable range, and global optimal values of the theory and optimal solution of the theory. The function f_1 is a unimodal function used to examine the accuracy of the algorithm. The function f_2 which contains infinite local optimal solution and has infinite local maximum near suboptimal solution, is a strong shock multimodal function. General optimization algorithms easily fall into local optimal solution. Thus, the ability of the algorithm is used to avoid falling into a local optimization algorithm. The function f_5 is a multimodal function. It has about 10^4 (D is dimension) local optimal solution. The function f_6 is a deep and local minimum multimodal function that has many local optimal solutions. The functions f_5 and f_7 are complex nonlinear functions used to test global search performance of algorithms. The function f_8 is a complex and classical function used to test the optimization efficiency of algorithms. In a certain number of iterations, this paper estimates the performance of IMPM by the test function of the best value, the worst value, average value, variance, average running time, and convergence rate.

1. Sphere Model function: $\min f_1 = \sum_{i=1}^n X_i^2, |X_i| \leq 100$. The optimal solution is $(X_1, X_2, \dots, X_n) = (0, 0, \dots, 0)$ and the optimal minimum value is 0, when $n = 30$.

Table 1. Parameter setting of the functions for the IPMA.

Function	N	δ	l	L_S	$\mathcal{A}$	α	K
1	3	10	10	5	0.1	1E-6	30
2	10	10	10	5	0.1	1E-7	30
3	3	2	10	5	0.1	1E-7	30
4	3	0.5	10	5	0.1	1E-9	30
5	3	1.6	10	5	0.15	1E-8	30
	10	0.512	10	5	0.1	1E-9	30
	10	0.206	10	5	0.1	1E-9	30
6	5	0.15	10	5	0.15	1E-9	30
7	5	5	10	5	0.1	1E-10	30
8	10	0.2	10	5	0.01	1E-9	30
9	10	60	10	5	0.1	1E-9	30
10	10	50	10	5	0.1	1E-6	30

doi:10.1371/journal.pone.0056652.t001

Table 2. Comparison performance of the IPMA in the ten functions.

Function	Algorithm	Best value	Worst value	Average value	Variance	Average running time	Convergence rate
1	[12]	10^{-13}	10^{-6}	10^{-7}	10^{-7}	26.8	–
	[6]	1.52E–02	5.47E–02	10^{-2}	–	–	–
	IPMA	10^{-18}	10^{-12}	10^{-13}	10^{-25}	0.749	100%
2	[1]	1.000000	0.990284	–	–	–	75%
	[4]	1.000000	1.000000	–	–	4.768	–
	[5]	1E–10	1E–10	–	–	7.626	–
	IPMA	1	1	1	0	2.104	100%
3	[1]	–30	–25	–	–	–	75%
	[4]	–30	–29	–	–	2.881	95%
	IPMA	–30	–29	29.86	0.2040	0.542	96%
4	[4]	–6.000000	–6.000000	–	–	7.015	100%
	IPMA	–6	–6	–6	0	0.704	100%
5	[4]	0	0	0	0	4.047	100%
	[7]	–	–	4.68E–17	2.64E–17	–	–
	IPMA	0	0	0	0	0.614	100%
	[8]	–	–	0	–	0.27	–
	IPMA	0	0	0	0	1.529	100%
	[6]	0	–	–	–	–	40%
	IPMA	0	0	0	0	2.731	100%
6	[4]	10^{-17}	10^{-17}	–	–	23.375	100%
	IPMA	10^{-21}	10^{-18}	10^{-19}	10^{-35}	1.043	100%
7	[12]	3.693E–07	2.738	6.375E–01	7.788E–01	29	–
	[4]	10^{-10}	10^{-10}	–	–	29.893	100%
	[6]	3.59E–02	9.94E–02	10^{-2}	–	–	–
	[8]	–	–	10^{-2}	–	0.42	–
	IPMA	10^{-13}	10^{-11}	10^{-11}	10^{-21}	1.675	100%
8	[1]	10^{-6}	10^{-6}	–	–	–	100%
	[12]	4.667	2.920E+02	6.690E+01	5.512E+01	50.0	–
	[7]	–	–	0.002234	0.002645	–	–
	IPMA	10^{-19}	10^{-17}	10^{-18}	10^{-36}	10.013	100%
9	[8]	–	–	10^{-2}	–	0.21	–
	IPMA	0	0	0	0	3.52	100%
10	[8]	–	–	–2094.57	–	0.06	–
	IPMA	–2094.90	–2094.90	–2094.90	0	11.12	100%
	[8]	–	–	–20946.58	–	1.97	–
	IPMA	–20949.00	–20949.00	–20949.00	0	5.99	100%

This table shows performance of IPMA compared with other algorithms through the above indicators in 50 cycles.
doi:10.1371/journal.pone.0056652.t002

2. Schaffer function: $\max f_2 = 0.5 - \frac{\sin^2(\sqrt{X_1^2 + X_2^2}) - 0.5}{[1.0 + 0.001(X_1^2 + X_2^2)]^2}$, $-100 \leq X_i \leq 100$, $i = 1, 2$. The optimal solution is $(X_1, X_2) = (0, 0)$ and the optimal maximum value is 1.

3. $\min f_3 = \sum \text{interger}(X_i)$, $-5.12 \leq X_i \leq 5.12$, $i = 1, 2, \dots, 5$. The function in the section $(-5.12, -5)$ has a global minimum value -30 .

4. $\min f_4 = -5 \sin X_1 \sin X_2 \sin X_3 \sin X_4 \sin X_5 - \sin(5X_1) \sin(5X_2) \sin(5X_3) \sin(5X_4) \sin(5X_5)$, $0 \leq X_i \leq \pi$, $i = 1, 2, \dots, 5$. The optimal solution is $(X_1, X_2, \dots, X_5) = (\pi/2, \pi/2, \dots, \pi/2)$ and the optimal minimum value is -6 .

5. Generalized Rastrigin function: . The function has many local minimum solutions, but it has only one global minimum

solution $(X_1, X_2) = (0, 0)$ and the global minimum value is 0, when $n = 2$, $n = 5$ and $n = 10$.

6. Quartic function: $\min f_6 = \sum_{i=1}^n iX_i^2$, $-100 \leq X_i \leq 100$, $i = 1, 2, \dots, n$. The optimal solution is $(X_1, X_2, \dots, X_5) = (0, 0, \dots, 0)$ and the optimal minimum value is 0, when $n = 20$.

7. Ackley's function: $\min f_7 = -20 \exp[-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n X_i^2}] - \exp(\sum_{i=1}^n \cos((2\pi X_i)/n)) + 20 - e$, $|X_i| \leq 32$. The optimal solution is $(X_1, X_2, \dots, X_n) = (0, 0, \dots, 0)$ and the optimal minimum value is 0, when $n = 20$.

8. Generalized Rosen Brock's function:

$\min f_8 = \sum_{i=1}^{n-1} [100(X_{i+1} - X_i^2)^2 + (1 - X_i)^2], |X_i| \leq 2.048$. The

optimal solution is $(X_1, X_2, \dots, X_n) = (1, 1, \dots, 1)$ and the optimal minimum value is 0, when $n = 3$.

9. Griewank function: $\min f_9 = 1/4000 \sum_{i=1}^n X_i^2 - \prod_{i=1}^n \cos(X_i/\sqrt{i}) + 1, X_i \in (-600, 600)$. The optimal minimum value is 0, when $n = 30$.

10. Schwefel function: $\min f_{10} = -\sum_{i=1}^n X_i \sin \sqrt{|X_i|}, X_i \in (-500, 500)$. The optimal minimum value is -2094.9 , when $n = 5$. The optimal minimum value is -20949 , when $n = 50$.

To verify the performance of the IPMA, we choose a computer whose CPU is made by AMD and has an operating system of Windows 7. The mathematical software Matlab 7.0 was used for the testing. Each function is applied 50 times. We present the parameters in the process of solving the functions for this paper in the table 1. To show the better performance of the IPMA, we compare it with other algorithms through the following indicators: best and worst solutions, average solution, variance, average running time, and average convergence rate in 50 cycles.

This table shows parameter setting of the ten functions. Population size N , number of population flow l , radius δ , coefficient of contraction A , pressure of population parameters α , number of local search L_s , and largest number of iterations K . $l_i = l, \delta_j(0) = (b_i - a_i)/2N$.

Experimental Result and Analysis

In Table 2, the data of function f_1 show that the improved population algorithm is better than frog-leaping algorithm in terms of precision. Other algorithms find it very difficult to achieve the optimal solution of function f_2 , but the IPMA can easily do so,

which indicates that the algorithm's ability of escaping from local optimal solution is very ideal. Function f_8 can test whether the algorithm's ability of global optimization is good. It is used to measure the efficiency of the algorithm, so we know the efficiency of the IPMA is very high compared with other algorithms from the data of function f_8 . The data functions f_3, f_4, f_5, f_7, f_9 and f_{10} reflect that the IPMA is a better algorithm from a certain angle. Through cooperation of the best value, the worst value, the average value, the variance, and the convergence rate, the IPMA is found to be better than the other algorithms in the calculation of stability and precision.

Conclusions

In this paper, based on the local search mechanism and crossover operator, we put forward an IPMA. The parameters of the IPMA are simple and easy to realize. By testing the 10 functions, we find that the IPMA is better than the basic PMA and is superior to many intelligent algorithms. The PMA is good at global searches, but it does not perform well in local search and its precision is low. We introduce local search in population and crossover operator to improve these deficits. Therefore, the IPMA is increased to avoid falling into local optimum capacity; its calculation speed and precision are also improved.

Author Contributions

Conceived and designed the experiments: XYL. Performed the experiments: YQZ. Analyzed the data: YQZ XYL. Contributed reagents/materials/analysis tools: YQZ. Wrote the paper: YQZ.

References

1. Zhou YH, Mao ZY (2003) A new search algorithm for global optimization: population migration algorithm (I). Journal of South China University of Technology (Nature Science) 31:1–5. (in Chinese).
2. Zhou YH, Mao ZY (2003) A new search algorithm for global optimization: population migration algorithm (II). Journal of South China University of Technology (Nature Science) 31: 41–43. (in Chinese).
3. Xu ZB (2004) Computational intelligence (Volume 1) - Simulated evolutionary computer. Beijing: High Education Press. 152p. (in Chinese).
4. Wang XH, Liu XY, Bai XH (2009) Population migration algorithm with Gauss mutation and steepest descent operator. Computer Engineering and Application 45: 57–60. (in Chinese).
5. Ouyang YJ, Zhang WW, Zhou YQ (2010) A based on the simplex and population migration hybrid global optimization algorithm. Computing Engineering and Application 46: 29–35. (in Chinese).
6. Guo YN, Cheng J, Cao YY (2011) A novel multi-population cultural algorithm adopting knowledge migration. Soft Computing 15: 897–905.
7. Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department. 10p.
8. Li X, Luo JP, Chen MR, Wang N (2012) An improved shuffled frog-leaping algorithm with extremal optimization for continuous optimization. Information Science 192: 143–151.
9. Eusuff MM, Lansey KE (2003) Optimization of water distribution network design using the shuffled frog leaping algorithm. Journal of Water Resource Planning and Management 129: 210–225.
10. Han Y, Cai JH (2010) Advances in Shuffled Frog Leaping algorithm. Computer Science 37: 16–19. (in Chinese).
11. Wu Y (2005) Convergence analysis on population migration algorithm. Xi'an: Xi'an University of science and Technology. 76p. (in Chinese).
12. He B, Che LX, Liu CS (2011) Novel hybrid shuffled frog leaping and differential evolution algorithm. Computer Engineering and Application 47:4–8. (in Chinese).