2015 International Conference on Computational Intelligence and Communication Networks

Multi-objective Artificial Bee Colony algorithm

Yanjiao Wang
School of Information Engineering
Northeast Dianli University
Jilin, China
e-mail: wangyanjiao1028@126.com

Yaojie Li
School of Information Engineering
Northeast Dianli University
Jilin, China
e-mail: 563274435@qq.com

***Abstract*—In order to approach the true Pareto front as fast as possible and make the distribution of solutions uniform on multi-objective optimization problems, a multi-objective optimization algorithm based on artificial bee colony algorithm has been presented in this paper, named MABC. Firstly, a novel selection scheme, which is used to guide the population evolution towards the true Pareto front and keep population diversified, substitutes the roulette wheel selection scheme. Secondly, the adaptive searching models are designed for the employed bees and onlookers, in which the convergence rate and diversity are considered simultaneously. Finally, an improved method of determining elite population is proposed to maintain diversity. Compared with other state-of-the-art algorithms, the simulation results of 5 standard test functions show that MABC achieves comparable results in terms of diversity and convergence metrics.**

Keywords multi-objective optimization, artificial bee colony algorithm, adaptive searching scheme, diversity maintenance

I. Introduction

Multi-objective optimization problems (MOPs) are frequently encountered in the field of science and engineering, which should guarantee that several targets in conflict achieve the best simultaneously. Many methods have been proposed to solve this problem. Non-dominated Sorting Genetic Algorithm (NSGA-II)[1] of second-generation multi-objective optimization algorithms is well-known and very successful, which adapts the fitness evaluation together with some sort of elitism combining fast non-dominated sorting rank and crowding distance to guide population evolution. More importantly, it has become a basic frame of most multi-objective optimization algorithms [3-6]. Because of complexity of MOPs, most of the existing multi-objective optimization algorithms are still poor in convergence and diversity. Researchers try to obtain a more efficient and robust multi-objective optimization algorithm. Through an intensive study, this paper has proposed a new multi-objective evolutionary algorithm — multi-objective artificial bee colony algorithm (MABC).

Artificial Bee Colony Algorithm (ABC), as a new biological-inspired optimization algorithm, was first proposed in 2005. It doesn't need set external parameters and has been confirmed to obtain better function optimization results than Differential Evolution algorithm and Particle Swarm Optimization algorithm. At present, it has been applied in many engineering areas, such as function optimization, artificial Neural Network Training, filter design, network optimization, robot path planning and production scheduling [8] etc. Some improvement mechanism has been proposed to obtain optimal multi-objective performance in this paper, such as: alternate onlookers' selection scheme, design novel self-adaptive searching mechanisms for employed bees and onlookers, and improve elite population maintenance strategy. The simulation results indicate that the solutions obtained by the proposed algorithm can efficiently converge to the true Pareto front and distribute uniformly on all test problems.

The rest of this paper is organized as follows. Section 2 and 3 provide respectively the description of ABC algorithm and MOPs. The proposed multi-objective artificial bee colony algorithm has been explained in Section 4. The experiment results and analysis have been presented in Section 5. Section 6 contains some conclusions.

II. ARTIFICAL BEE COLONY ALGORITHM

ABC algorithm is used to handle function optimization problems based on the actual foraging behavior of bees at present. The bees are classified into three types: employed bees, onlookers and scouts. Employed bees and onlookers are used to explore new food sources, while the effect of scouts is to prevent species of food sources from being too limited. The process of finding the optimal nectar is as follows: produce *2N* locations randomly, evaluate them and mark *N* food sources with comparable more nectar amounts. Employed bees are moved onto marked food sources and search new nectars around themselves according to the formula (1).

$$V_{ij} = x_{ij} + R_{ij}(x_{ij} - x_{kj}) \tag{1}$$

Where V_{ij} is a new location, R_{ij} is generated randomly in the range $[-1,1]$, x_{kj} is another nectar except the *i*-th one which is randomly selected .

Choose those better food sources as new locations of employed bees (called the initial marked nectars) according to the nectar amounts of parent and child food sources. Employed bees release the information proportional to the nectar amounts of marked food sources and recruit onlookers. Each onlooker chooses the appropriate food source by the roulette wheel selection and searches around it according to equation (1). Select the better half of solutions from candidate individuals generated by employed bees and onlookers. If a solution couldn't be improved within "*limit*" trials, it will be abandoned, and the corresponding scout produces randomly a new solution to replace the old one. The above method repeats until the termination condition is satisfied.

In order to further understand the algorithm principle, the main steps are given as below.

Initialize

REPEAT

Memorize the better half of food sources.

Move the employed bees onto these food sources, search around them, and evaluate them.

Move the onlookers onto the appropriate food sources by the roulette wheel selection, search around the chosen food sources, and determine their nectar amounts.

Combine the population generated by employed bees with that of onlookers.

Judge whether the scouts will appear, and carry out corresponding operation.

UNTIL (requirements are met)

III. MULTI-OBJECTIVE ARTIFICAL BEE COLONY ALGORITHM

To obtain the optimal performance of multi-objective optimization, a multi- objective artificial bee colony algorithm has been proposed in this paper. It adopts the basic framework of NSGA-II algorithm, improves ABC algorithm from two aspects and proposes an improved maintenance approach of elite population.

A. An improved selection scheme

In ABC algorithm, greedy roulette wheel selection method is adopted for each onlooker to choose an appropriate food source and search around it. Although it accelerates convergence rate, it makes this algorithm trap easily in local optimum to a great extent.

"Pheromone and Sensitivity" model is applied to choose suitable region in Free Search algorithm (FS) [8]. The details of the model are as follows.

a) Calculate pheromone of every individual ($nf(i)$) as follows:

$$nf(i) = \begin{cases} \dfrac{f(i) - f_{\min}}{f_{\max} - f_{\min}} & if \quad f_{\max} \neq f_{\min} \\ 0 & else \end{cases} \tag{2}$$

Where, $f(i)$ is the fitness of individual *i*, f_{max} and f_{min} represent separately the maximum and minimum fitness of all individuals.

b) Randomly select a region whose pheromone should be suited to its sense (The sense of individual *i*($S(i)$) is randomly generated between 0 and 1), in other words, the following condition should be satisfied:

$$nf(k) \le S(i) \qquad k \sim \text{randint}(1, N)$$

From the above "Pheromone and Sensitivity" model, some conclusions can be drawn: Firstly, because of randomness of sensitivity, theoretically, any region can be searched, which can avoid trapping into local optimal to a great extent. Secondly, this selection model must satisfy step *b)*, so it makes the direction of algorithm evolution clear and algorithm converge in search space. The "Pheromone and Sensitivity" model of choosing region in FS is similar to the method of selecting an appropriate food source for onlookers, so the model may substitute the roulette wheel selection scheme.

B. New searching model

To converge to the optimal Pareto front and distribute uniformly, searching models of employed bees and onlookers are improved in this paper.

a) Improved searching model for employed bees

In ABC algorithm, the role of employed bees is to explore new solutions. It performs crossover operations between an randomly chosen individual and itself, so there exists blindness for algorithm evolution. Considering the particularity of MOPs, there are various non-dominated sorting ranks at the early generation. More excellent the individuals are, lower the sorting ranks are. Individuals at different non-dominated sorting ranks have different effects on population evolution, therefore formula (3) (called as exploration probability formula) is proposed to guide excellent and general individuals evolution respectively.

$$CR(i) = 1 - \frac{px(i) - min(px)}{max(px) - min(px)} \tag{3}$$

Where *px(i)* represents the non-dominated sorting rank of employed bee *i*. Generate a value randomly between 0 and 1. If the value is less than *CR* and the non-dominated sorting ranks of each onlooker are almost different, this employed bee may be at the poor storing rank. To accelerate convergence rate, the general solutions at the poorer sorting rank should learn more from excellent ones at the lower sorting rank, while the excellent individuals at the lower rank should learn more from themselves. Meanwhile, population should be dynamically adjusted according to evolutional generation. We hope individuals in initial stage can learn more from better ones to accelerate convergence rate, while at the final stage new solutions are explored in a small region to improve the ability of local searching. So a new adaptive searching model is proposed for employed bees in this paper, the details are as formula (4).

$$V_{ij} = (\frac{1}{px(i)})x_{ij} + (1 - \frac{1}{px(i)})x_{k1j} + (-1 + 2 \times (e^{(\frac{g}{g_{\max}}\ln(2))} - 1)) \times (x_{ij} - x_{kj}) \tag{4}$$

Where g and $g_{\max}$ represent respectively this iteration and the maximum iteration, and $k1$ is chosen randomly from individuals of better non-dominated storing rank.

On above search strategy, the effects of excellent individuals at the better ranks are emphasized. Although it can make the algorithm fast convergence, it's easy to fall into local optimal. So random disturbance has to be introduced to guide evolution of those better individuals (the better individuals should meet the following conditions that a randomly generated value, between 0 and 1, is more than the probability calculated by formula (3), and all employed bees are at same non-dominated sorting rank), the details are as following formula (5).

$$V_{ij} = x_{ij} + (-1 + 2 \times rand \times 2^{e^{(\frac{g}{g_{\max}})}}) \times (x_{ij} - x_{kj}) \quad (5)$$

The pseudo code of searching model of employed bees is presented in Fig.1. The following notations are adopted: *POP* represents the population composed by employed bees, *CRCompute* is adopted to compute CR, *VCompute1* and *VCompute2* are applied to compute new locations as formula (4) and (5) respectively.

```
function searching model of employed bees
begin
for each i
if min(px)~=max(px)
CR_i= CRCompute(POP(i))
if rand<CR
V_i= VCompute1(POP(i))
else
V_i= VCompute2(POP(i))
end if
else
V_i= VCompute2(POP(i))
end if
end for
```

Fig.1: Representation of pseudo code for searching model of employed bees

b) Improved crossover operation for onlookers

In ABC algorithm, onlookers perform crossover operations between a more excellent individual and itself to search new food source, which leads to fast convergence. Because both convergence criterion and diversity criterion should be met, it is too difficult to determine which individual is superior in MOPs.

At the early trials, most individuals are at different non-dominated ranks, and the superior individual should be at lower rank, so sorting rank can be used to evaluate superiority-inferiority of individuals. However, at the later stage, almost all the candidates are at the same non-dominated sorting rank and already close to the true front, so crowding distance representing diversity can be used as the standard of superiority-inferiority of evaluating individuals. From the above analysis, evolution generations can be used to choose superior individuals in MOPs. Through theoretical analysis and repeated tests, formula (6) based on evolutional generation is proposed in this paper.

$$CR = 2 - e^{\frac{g_{\max} - g}{g_{\max}} \log(2)} \quad (6)$$

If a randomly generated value, between 0 and 1, is less than *CR* and the non-dominated sorting ranks of onlookers are almost different, it may be at the early generations, so the superior individual *k* is chosen according to non-dominated sorting ranks by new selection scheme of Part 4.1, otherwise, *k* is selected according to crowding distance. Meanwhile, considering the effect of evolution generation on the moving step length during the searching process, adaptive adjustment of the moving step length is beneficial to improve algorithm evolution rate, so the searching model of onlookers is modified as formula (7).

$$V_{ij} = x_{kj} + (-1 + 2 \times rand \times 2^{e^{(\frac{g}{g_{\max}})}}) \times (x_{kj} - x_{ij}) \quad (7)$$

For onlookers, new selection scheme in Part 4.1, which is used to choose better individual *k,* and adaptive crossover model described as formula (7) containing random disturbance are simultaneously employed as a new searching model of onlookers. Convergence rate and diversity are both considered in the new searching model. The proposed searching model not only converges to the optimal Pareto front as fast as possible, but also avoids advanced stagnation.

The pseudo code of searching model of onlookers is presented in Fig.2. The following notations are adopted: *POP* represents the current population, *KCompute1* and *KCompute2* represents respectively the better individual selected according to non-dominated sorting ranks and crowding distance by the novel selection method in Part 4.1, *CRCompute* and *VCompute* are respectively used to compute CR and new locations as formula (6) and formula (7).

```
function searching model of onlookers
begin
for each i
CR_i= CRCompute(POP(i))
if rand<CR
k=KCompute1(POP)
else
k=KCompute2(POP)
end if
V_i= VCompute(POP(k))
end for
```

Fig.2: Representation of pseudo code for searching model of onlookers

C. New searching model

The method to determine elite population has an effect on convergence rate, and is an important factor of superiority-inferiority of algorithm. In NSGA-II algorithm, elite population is determined according to non-dominated sorting ranks and crowding distance, so the non-dominated sorting approach and the model of computing crowding distance are the keys to NSGA-II algorithm. Some improved methods of evaluating crowding distance have been proposed. To make the proposed algorithm gain the best performance, this paper applies formula (8) from refs 9 to calculate the crowding distance, the details are as follows.

$$Dc(B) = \sum_{i=1}^{m}\left(\left|f_i(A) - f_i(C)\right| - \left|f_i(B) - f_i(O)\right|\right)$$
$$= \sum_{i=1}^{m}\left(\left|f_i(A) - f_i(C)\right| \times 0.5 + \min[\left|f_i(A) - f_i(B)\right|, \left|f_i(B) - f_i(C)\right|]\right) \quad (8)$$

Where m is dimension of targets, individual A, B and C are at the same non-dominated rank, individual A and C are close to individual B in one target, individual O is the center between individual A and C.

An important idea is proposed in [10]: individuals selected one by one have an effect on diversity of the rest, in other words, diversity of the remaining individuals is changing constantly, so the way to select elite individuals according to the constant crowding distance was redefined. This paper proposes an improved method to determine elite population based on this idea, and the details are as follows: Firstly, apply formula (8) to calculate the crowding distance of individuals. Secondly, because the individual with the smallest crowding distance is poorest in the diversity, it is first removed. Finally, remove individuals according to the above method until the number of remaining individuals reaches predetermined size of elite population.

D. The procedure of multi-objective artificial bee colony algorithm

Multi-objective artificial bee colony algorithm (MABC) is proposed based on the above improvements from three aspects in this paper. The procedure is described as follows:

Step 1 Initialization: related parameters are given: the size of population is *2N*, the number of employed bees is *N,* and so is onlookers, the maximum generation is G_{max};

Step 2 Generate an initial population of *2N* individuals randomly;

Step 3 Obtain all information of each individual, including fitness of each target, the fast non-dominated sorting rank and crowding distance according to formula (8);

Step 4 Randomly select two individuals and store the better one according to championship selection method in NSGA-II algorithm;

Step 5 Employed bees search self-adaptively and generate *N* individuals by the means of Part 3.2.1, if an individual exceeds the searching boundary, restrict it within the boundary;

Step 6 Choose the better *N* individuals from population generated from step 4 and 5, according to the new method of determining elite population;

Step 7 Onlookers search adaptively and generate *N* individuals by the means of Part **B**, if an individual exceeds the searching boundary, restrict it within the boundary;

Step 8 Choose the better *2N* individuals to form new population from the individuals generated from **step 2, 4** and **7** by method of **Part C**;

Step 9 Repeat **Step 3** to **8** G_{max} times, then output Pareto front.

IV. EXPERIMENT RESULTS AND ANALYSIS

The performance of MABC is tested on classical ZDT problems of two targets from Ref. [16] in this paper. Convergence metric and diversity metric in Ref. [1] are used to evaluate superiority-inferiority of multi-objective optimization algorithms in this paper.

All simulations are performed in MATLAB 7.5 with 2Ghz Intel Centrino Duo PC, in our experiments, the parameters of MABC algorithm are set as below: the size of population is set to be 100, in other words, the number of employed bees is 50, and so is onlookers. The maximum number of generations is set to 250.

MABC is compared with several state-of-the-art multi-objective optimization algorithms in this paper. To avoid harmful effect on algorithm evaluation caused by randomness, 30 independent experiments have been carried on to calculate average and standard deviations of convergence criterion and diversity criterion of every algorithm. The experimental results of ZDT are listed in Table 1-Table5.

Table 1: The experimental results of ZDT1

algorithm	*Y*	Δ
	mean(STD)	mean(STD)
NSGA-II[1]	0.03348(0.00475)	0.39031(0.00187)
PESA-II[11]	0.00105(0.00000)	0.84816(0.00287)
σ-MOPSO[12]	0.01638(0.00048)	0.39856(0.00731)
NSPSO[3]	0.00642(0.00000)	0.90695(0.00000)
MPSO[13]	0.00133(0.00000)	0.68132(0.01335)
AMPSO[14]	0.00099(0.00000)	0.31826(0.00060)
INSGA-II[10]	0.00057(0.00000)	0.24073(0.00017)
PDE[6]	0.00118(0.00000)	0.374037(0.001381)
MPDE[15]	0.00199(0.00071)	0.76643(0.23280)
MABC	**0.00006**(0.00000)	**0.13155**(0.00253)

Table 2: The experimental results of ZDT2

algorithm	*Y*	Δ
	mean(STD)	mean(STD)
NSGA-II[1]	0.07239(0.03168)	0.43077(0.00472)
PESA-II[11]	0.00074(0.00000)	0.89292(0.00574)
σ-MOPSO[12]	0.00584(0.00000)	0.38927(0.00458)
NSPSO[3]	0.00951(0.00000)	0.92156(0.00012)
MPSO[13]	0.00089(0.00000)	0.63922(0.00114)
AMPSO[14]	0.00074(0.00000)	0.31996(0.00068)
INSGA-II[10]	0.00074(0.00000)	0.40057(0.15938)
PDE[6]	0.00132(0.00000)	0.38334(0.00166)
MPDE[15]	0.00576(0.00021)	0.27996(0.04204)
MABC	**0.00007**(0.00000)	**0.12114(**0.00192)

Table 3: The experimental results of ZDT3

algorithm	*Y*	Δ
	mean(STD)	mean(STD)
NSGA-II[1]	0.11450(0.00794)	0.73854(0.01971)
PESA-II[11]	0.00789(0.00011)	1.22731(0.02925)
σ-MOPSO[12]	0.10205(0.00238)	0.76016(0.00349)
NSPSO[3]	0.00491(0.00000)	061072(0.00069)
MPSO[13]	0.00418(0.00000)	0.83195(0.00892)
AMPSO[14]	0.00391(0.00000)	0.53154(0.00036)
INSGA-II[10]	0.00330(0.00000)	0.56963(0.00136)
PDE[6]	0.001425(0.00000)	0.57199(0.00102)
MPDE[15]	**0.00032(**0.00012)	0.52308(0.01392)
MABC	0.00257(0.00001)	**0.42182**(0.00195)

Table 4: The experimental results of ZDT4

algorithm	Y	Δ
	mean(STD)	mean(STD)
NSGA-II[1]	0.51305(0.11846)	0.70261(0.06465)
PESA-II[11]	9.98254(20.1340)	1.01136(0.00072)
σ-MOPSO[12]	3.83344(1.87129)	0.82842(0.00054)
NSPSO[3]	4.95775(7.43601)	0.96462(0.00156)
MPSO[13]	7.37429(5.482860)	0.96194(0.00114)
AMPSO[14]	0.40311(0.01259)	0.65060(0.00376)
INSGA-II[10]	**0.00873**(0.00000)	0.53590(0.12042)
PDE[6]	42.22191(2.08852)	0.94539(0.14223)
MPDE[15]	2.04314(0.09510)	0.33441(0.14223)
MABC	0..11712(0.00180)	**0.14608**(0.00589)

Table5: The experimental results of ZDT6

algorithm	Y	Δ
	mean(STD)	mean(STD)
NSGA-II	7.80680(0.001667)	0.64448(0.03504)
PESA-II	0.23255(0.004945)	1.04422(0.15811)
MMDD-DE	0.00598(0.000032)	0.42158(0.02684)
MOEO	0.00630(0.000033)	0.22547(0.03389)
DEMO	0.57403(0.02934)	0.86252(0.04453)
ε -DEMO	0.54372(0.61489)	0.51864(0.27761)
ε -ODEMO	0.00062(0.000014)	0.18415(0.01512)
MABC	**0.00007**(0.00000)	**0.12114**(0.00192)

As shown in Table 1-Table 5, the average value of diversity criterion of MABC on all problems are all closer to 0 than other algorithms, which shows the Pareto front obtained by MABC algorithm is the most uniform, and MABC algorithm has evident advantage in diversity. The mean of convergence criterion of MABC is slightly larger than MPDE and INSGA-II on ZDT2 and ZDT3, which indicates that the convergence performance of MABC is slightly inferior to MPDE and INSGA-II. Except ZDT2 and ZDT3, the mean of convergence criterion of MABC is significantly lower than the other algorithms, which shows that MABC is significantly superior to most multi-objective algorithms in convergence. In addition, smaller variances of convergence and diversity criterion of MABC demonstrate the robustness of its own. In each experiment, solutions obtained in finite iterations can almost converge to the true Pareto front and distribute uniformly, it is proved that MABC is effective in MOPs and can obtain uniform and true Pareto front. That is because that in the initial stage of evolution, MABC treats excellent and general individuals respectively according to sorting ranks, which makes MABC converge quickly to optimal Pareto front. In the later stage of evolution, while most of individuals are at better Pareto sorting rank, crowding distance can make distribution of solutions uniform.

V. CONCLUSIONS

A multi-objective algorithm based on artificial bee colony algorithm has been proposed in this paper. New adaptive searching operations are designed for employed bees and onlookers, which makes it not difficult to determine the better individuals in MOPs. Meanwhile, the approach of determining elite population is improved. The results on several general test functions have shown that solutions obtained by MABC algorithm are closer to the true Pareto non-dominated front and distribute more uniformly than other multi-objective optimization algorithms. It has been proved that MABC is an effective multi-objective optimization algorithm with popularization value in practical application. Meanwhile, application field of artificial bee colony algorithm is also widen here.

REFERENCES

[1] A. Seshadri. *A fast and elitist multi-objective genetic algorithm: NSGA-II*. IEEE Trans on Evolutionary Computation, 6(2), pp.182-197, 2002.

[2] S. Bleuler, M. Brack, and L. Thiele etc. *Multiobjective genetic programming: reducing bloat using SPEA2*. Computation CEC2001. 2001:536-543.

[3] S. Mostaghim, J. Teich. *Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO)*. Swarm Intelligence Symp 2003. 2003:26-33.

[4] R H Shang, L C Jiao, and W P Ma etc. *An immune memory clone algorithm for constrained multi-objective optimization*. ACTA ELECTRONI- CASINCA . 37(6), pp.1289-1294,2009.

[5] D D Yang, L C Jiao, and M G Gong, etc. *Clone Selection Algorithm to solve preference multi-objective optimization.* Journal of Software,21(1), pp.14-33,2010.

[6] R H Lei, Y H Cheng. *A Pareto-Based Differential Evolution Algorithm for Multi-objective Optimization Problems*. 2010 Chinese Control and Decision Conference. 2010: 1608-1613.

[7] D Karaboga , B Basturk , *On the performance of artificial bee colony (ABC) algorithm.* Applied Soft Computing , 8(1), pp.687-697,2008.

[8] K Penev, G Littlefair, *Free Search—a comparative analysis*. Information Sciences, 172(1), pp. 173-193,2005.

[9] H. J. Sun, C. H. Peng, J. F. Guo. *Non-dominated Sorting Differential Evolution Algorithm for Multi-objective Optimal Integrated Generation Bidding and Scheduling*. 2009 IEEE International Conference on Intelligent Computing and Intelligent Systems, 2009:372-376.

[10] C Y LUO, M Y Chen, C Y Zhang. *Improved NSGA-II algorithm with circular crowed sorting*. Control and Decision. 25(2), pp.227-231, 2010.

[11] D. W. Corne, N.R. Jerram, J. D. Knowles, et al. *PESA-II Region-based selection in evolutionary multi-objective optimization*. Proc of the Genetic and Evolutionary Computing Conf. 2001:283-290.

[12] C. A. Coello, G. T. Pulido, M. S. *Lechuga. Handling multiple objectives with particle optimization.* IEEE Trans on Evolutionary Computation, 8(3), pp.256-279, 2004.

[13] P. K. Tripathi, S. Bandyopadhyay. *Adaptive multi-objective particle swarm optimization algorithm*. Proc of IEEE Congress on Evolutionary Computation. Singapore, 2007:2281-2288.

[15] X Q Chen, Z X Hou, J X Liu. *Multi-objective Optimization with Modified Pareto Differential Evolution*. 2008 International Conference on Intelligent Computation Technology and Automation. 2008:90-95.

[16] T Liu. *Evolutionary Algorithms for Multi-objective Optimization Problem*. Xidian University.2010.