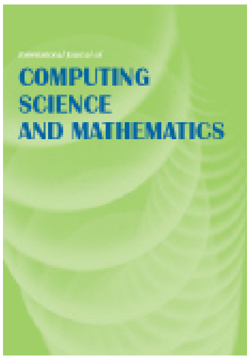


International Journal of Computing Science and Mathematics

ISSN online: 1752-5063 - ISSN print: 1752-5055
https://www.inderscience.com/ijcsm

An enhanced multi-objective particle swarm optimisation with Levy flight

Hai-ying Lan, Gang Xu, Yu-qun Yang

DOI: 10.1504/IJCSM.2023.10055625

Article History:

Received:	06 February 2021
Last revised:	07 May 2021
Accepted:	20 July 2021
Published online:	20 April 2023

An enhanced multi-objective particle swarm optimisation with Levy flight

Hai-ying Lan

School of Mathematics and Statistics,
Jiangxi Normal University,
Nan Chang, Jiang Xi, 330022, China
Email: lanhaiying@163.com

Gang Xu*

Department of Mathematics,
Nanchang University,
Nan Chang, Jiang Xi, 330031, China
Email: xgang_csu@ncu.edu.cn
*Corresponding author

Yu-qun Yang

The Middle School attached to Nanchang University,
Nan Chang, Jiang Xi, 330047, China
Email: yangyuqun@163.com

Abstract: In the scope of multi-objective particle swarm optimisation (MOPSO) research, avoiding premature convergence remains a challenge. To address this issue, the article develops an enhanced multi-objective particle swarm optimisation with Levy flight (LF-MOPSO). In LF-MOPSO, swarm is made to evolve based on the original MOPSO to accelerate convergence. Then, Levy flight is adaptively activated to maintain diversity, so as to deal with the premature convergence when Pareto frontier is stagnant. It realises the transformation between shrinkage and divergence of population diversity by self-adaptive conversion mechanism, which further improves the search ability of MOPSO. LF-MOPSO has been contrasted with some recently improved MOPSOs, the experimental outcomes indicate that LF-MOPSO ensures the better approximation to the Pareto optimal frontier, and gains the non-dominated solutions with good diversity and distribution.

Keywords: MOP; multi-objective optimisation; PSO; particle swarm optimisation; Pareto solution; external archive; Levy flight; conversion mechanism; population diversity.

Reference to this paper should be made as follows: Lan, H-y., Xu, G. and Yang, Y-q. (2023) 'An enhanced multi-objective particle swarm optimisation with Levy flight', *Int. J. Computing Science and Mathematics*, Vol. 17, No. 1, pp.79–94.

Biographical notes: Hai-ying Lan is currently a Lecturer at the School of Mathematics and Statistics from Jiangxi Normal University (China). She received her Master's degree in Mathematics and Statistics at Jiangxi Normal

University. Her current research interests are focuses on swarm intelligence algorithm and evolutionary algorithm.

Gang Xu received his PhD in Applied Mathematics at South China University of Technology in 2008. He is currently a Professor at the Department of Mathematics from Nanchang University (China). His current research interests include particle swarm optimisation and evolutionary algorithm theory.

Yu-qun Yang is currently a teacher at the middle school attached to Nanchang University. She received her Master's degree in Mathematics and Statistics at Jiangxi Normal University (China). Her current research interests are focuses on particle swarm optimisation.

This paper is a revised and expanded version of a paper entitled 'An enhanced multi-objective particle swarm optimization with levy flight' presented at *9th International Workshop on Swarm Intelligent Systems (IWSIS2020) & 7th International Workshop on Mobile and Wireless Computing (IWMWC2020)*, Ganzhou, Jiang Xi, China, 11–13 December, 2020.

1 Introduction

There are many multi-objective optimisation problems (MOPs) in the real world, multi-objective particle swarm optimisation (MOPSO) algorithms for MOPs are increasingly concerned in these years (Wang et al., 2018; Liu et al., 2020; Hu and Gong, 2021; Garg et al., 2020; Li et al., 2020; Xue et al., 2020, 2021). A lot of researches on MOPSO have been accepted for publication in different ways (Coello et al., 2004; Wang et al., 2013; Xu et al., 2015; Niu et al., 2019; Xu et al., 2020a; Zhang et al., 2020a, 2020b). In MOPSO, contrasted with single objective optimisation, getting a solution set in which the non-dominated solutions are diverse and well-distributed is very important for the ultimate compromise of MOPs.

However, the fast convergence of particle swarm optimisation (PSO) (Kennedy and Eberhart, 1995) often means that it will lose diversity rapidly during the iteration, which necessarily brings about premature convergence. Due to the above shortcomings of PSO, although MOPSO possesses fine global search capability, the original MOPSO (Coello and Lechuga, 2002) cannot guarantee that the global convergence probability is one (Xu et al., 2020b). In the vicinity of Pareto solution, the optimisation effect of MOPSO abates significantly now and then. In order to solve problem of premature convergence, a lot of effective tactics are introduced in MOPSO, the improved MOPSOs can guarantee good approximation to the Pareto optimal frontier, and obtain non-dominated solutions with fine variousness and distribution.

Coello et al. (2004) introduced a secondary repository of particles into MOPSO, and also added a special mutation operator to enrich the exploration ability. Kazuhiro et al. (2008) proposed a new MOPSO to solve structural problems. In this algorithm, a optimisation method base on gradient is united with MOPSO for reducing the difficulty of constraint processing. Chen et al. (2010) proposed a clonal selection principle based mixed immunisation multi-objective optimisation. In the improved MOPSO, by keeping an ideal balance between extensive search and small scale search, it raises search capability and accelerates the convergence rate to the Pareto optimal frontier. Xu et al.

(2015) presented a business like multi-objective dichotomy line search based MOPSO to attain satisfied results in the matter of quality of solution. Zhu et al. (2018) developed an new external archive-piloted MOPSO. This proposed algorithm accelerates the convergence speed of MOPSO. Han et al. (2018) developed an adaptive gradient MOPSO algorithm to enhance computation capability. Zhang et al. (2020a) proposed a modified particle swarm optimisation (AMPSO) to solve the multimodal multi-objective problems. An efficient multi-objective optimisation algorithm based on level swarm optimiser is presented to address the problem that the diversity and convergence of non-dominated solutions are difficult to balance (Zhang et al., 2020b).

Although a lot of works have been done to raise search capability and convergence for MOPSO, there is still space for improvement in MOPSO's performance. In this research, an enhanced multi-objective particle swarm optimisation with Levy flight (LF-MOPSO) is presented to show that LF-MOPSO can get closer to the Pareto optimal frontier and the ultimate non-dominated solutions have good diversity and distribution. The remainder of this paper is arranged as below: Section 2 presents the original MOPSO algorithm; Section 3 gives the proposed approach; Section 4 gives simulation results and discussions; finally, Section 5 summarises this paper.

2 Original MOPSO algorithm

MOPSO is a random optimisation algorithm based on population. The particle's position can be defined as $x_i = (x_{i1}, x_{i2}, ..., x_{iD})$, and $v_i = (v_{i1}, v_{i2}, ..., v_{iD})$ is the velocity. $p_i^{(t)} = (p_{i1}^{(t)}, p_{i2}^{(t)}, ..., p_{iD}^{(t)})$ is the best place of the *i*th particle in history during the iteration, $p_g^{(t)} = (p_{g1}^{(t)}, p_{g2}^{(t)}, ..., p_{gD}^{(t)})$ is the best place for the whole population so far.

A successive minimising MOP is considered. It includes D decision variables and m objectives.

$$\begin{cases} \min\ f(x) = (g_1(x), g_2(x), ..., g_m(x)), \\ \quad s.t.\ l_j(x) \le 0, \quad j = 1, \cdots, q, \end{cases} \tag{1}$$

where $x = (x_1, ..., x_D) \in S_x \subset \mathbb{R}^D$ is the decision vector with *D* dimensions, S_x is the decision scope. $f_i(x)$ is the *i*th target for MOP, $l_j(x)$ is the *j*th constraint. $f(x) \in S_y \subseteq \mathbb{R}^m$ is the target vector and S_y is the target scope. For convenience, this paper does not consider the constraint problems.

One decision vector $a \in S_x$ is called dominating the other $b \in S_x$, it must satisfy $g_i(a) \le g_i(b)$ and any *i* belongs to set $\{1, ..., m\}$, moreover, the existence of *i* belonging to set $\{1, ..., m\}$ makes $g_i(a) < g_i(b)$ hold. This dominant relationship is denoted by $a \prec b$.

In addition, if one solution which is called Pareto solution is not dominated by all other solutions. These Pareto solutions are deposited into an external archive $A(t)$. Before algorithm running, the external archive is initialised as an empty set, denoted by $A(0)$. the best place of the *i*th particle in history $p_i^{(t)}$ is given as follows:

$$p_i^{(t)} = \begin{cases} x_i^{(t)}, & if\ x_i^{(t)} \prec p_i^{(t-1)}, \\ p_i^{(t-1)}, & otherwise. \end{cases} \tag{2}$$

During the search, the archive $A(t)$ is renewed by using $A(t-1)$ and $p_i^{(t)}$ as follows.

$$A(t) = A(t-1) \cup p_i^{(t)}, \text{ if } \lambda_i(t-1) \prec\succ p_i^{(t)}, \tag{3}$$

where $A(t) = \{\lambda_1(t), \lambda_2(t), ..., \lambda_H(t)\}$, H is the maximal allowable capacity of the external archive $A(t)$. $\prec\succ$ denotes that neither of $\lambda_i(t-1)$ and $p_i^{(t)}$ is dominated by the other.

Moreover, the speeds and locations of particles at next step are renewed as follows:

$$\begin{aligned} v_i^{(t+1)} &= wv_i^{(t)} + c_1 r_1 \otimes (p_i^{(t)} - x_i^{(t)}) + c_2 r_2 \otimes (p_g^{(t)} - x_i^{(t)}), \\ x_i^{(t+1)} &= x_i^{(t)} + v_i^{(t+1)}, \end{aligned} \tag{4}$$

where i belongs to set $\{1, 2, ..., n\}$, w is the inertance weight, c_1 and c_2 are both the acceleration modulus, r_1 is stochastic number vector in which each component has a uniform distributed in [0,1], so is r_2, $\otimes$ is component-wise multiplications. $v_{id}^{(t)} \in [-v_M, v_M]$ with $d = 1, 2, ..., D$, v_M and is the upper bound of speed, $-v_M$ is the lower bound of speed. $x_{id}^{(t)} \in [x_m, x_M]$, x_M is the upper limit of search, and x_m is the lower bound of search.

3 Proposed algorithm

To avoid premature convergence and improve the diversity and distribution of non-dominated solutions, respectively, an enhanced LF-MOPSO has been proposed. In LF-MOPSO, Levy flight is employed to prevent premature convergence, moreover, a self-adaptive conversion method is designed for obtaining better equilibrium between exploitation and exploration. The pseudocode of LF-MOPSO is given in Table 1**.**

Table 1 Proposed LF-MOPSO algorithm

1. *t*=0, initializing Levy flight parameters, population size, positions and velocities of particles.
2. **For** *t*=1 to *t*= maximal number of iterations
   (1) Calculating the fitness value.
   (2) Obtaining the non-dominated solutions.
   (3) Storing non-dominated solutions found in archive according to Eq. (3) and calling adaptive grid mechanism.
   (4) **If** (counter exceeding maximum allowed value)
   Levy flight is self-adaptively activated.
   End
   (5) **If** (new non-dominated solutions entering empty grids)
   Levy flight stops.
   End
   (6) Selecting p_g from the archive.
   (7) Updating position and speed of particles according to Eq. (4).
3. Outputting the solutions in *A*(t) and stop.

3.1 Levy flight

Levy flight (Chechkin et al., 2008) is a kind of non- Gaussian stochastic motion in which the random walk comes from Levy stationary distribution. The distribution is a straightforward power-law expression $L(s) \sim |s|^{-1-\beta}$, where $0 < \beta < 2$ is an exponent. In general, Levy distribution can be simply expressed as:

$$L(s,\gamma,\mu) = \begin{cases} \sqrt{\dfrac{\gamma}{2\pi}} \exp[-\dfrac{\gamma}{2(s-\mu)}] \dfrac{1}{\sqrt{(s-\mu)^3}}, & 0 < \mu < s < \infty, \\ 0, & s \le 0, \end{cases} \tag{5}$$

where μ is conversion parameter, $\gamma > 0$ is scale (control distribution scope) parameter.

Generally speaking, Levy distribution ought to be expressed by Fourier transform as follow:

$$F(x) = \exp(-\alpha |x|^{\beta}), \ 0 < \beta \le 2 \tag{6}$$

where $\alpha \in [-1,1]$ is a parameter. Stability index $\beta \in (0,2]$ is also called Levy index. Except for a few special cases, the analytical expression of integral cannot be known for general β.

On stochastic walk, Step size s can be computed by using Mantegner algorithm as follows:

$$s = \frac{q}{\sqrt[\beta]{|p|}}, \tag{7}$$

where q and p are derived from normal distributions. Namely,

$$q \sim N(0,\sigma_q^2), \ p \sim N(0,\sigma_p^2) \tag{8}$$

where $\sigma_q = \left\{ \dfrac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma[1+\beta/2]\beta 2^{(\beta-1)/2}} \right\}^{1/\beta}$. The step width is computed by $step\ size = 0.01\,s$.

The coefficient 0.01 mentioned above is from the fact that $L/100$ should be a classic walking step width, in which L is a classic length scale; On the other hand, Levy flight can get too radical, so that it can cause new solutions to escape from local optima.

3.2 Self-adaptive conversion mechanism

The external archive plays a very important role, because $\boldsymbol{p}_g$ solution is selected from the external archive. In this paper, to generate well-distributed Pareto frontiers, the adaptive grid mechanism for the archive (Coello et al., 2004) is introduced into LF-MOPSO. If the obtained non-dominated solutions has exceeded maximal allowable capacity for $A(t)$, then the adaptive grid procedure is called. After some iterations, if no one new non-dominated solution goes into empty grids near the current Pareto frontier, it means that the current Pareto frontier is stagnant, that is to say, the MOPSO algorithm become premature convergence or trapped in local optima. In order to detect such a situation, a counter is equipped to track how many times the current Pareto frontier has not changed. If the counter exceeds a maximal allowed value (say, *NA*), this means that MOPSO may be premature convergence or trapped in local optima, then Levy flight is self-adaptively activated to guide the algorithm to escape from the premature convergence or the local optima. If a new non-dominated solution gets into the empty grids near the current Pareto frontier, Levy flight stops automatically and the current algorithm rerun the original MOPSO algorithm, meanwhile, the counter is set to 0.

3.3 Enhanced MOPSO with levy flight

Besides the original MOPSO, LF-MOPSO mainly includes Levy flight and the self-adaptive conversion mechanism. In LF-MOPSO, first, the particle's position and velocity are updated by equation (4), the archive $A(t)$ followed according to equation (3). If the current non-dominated solutions are larger than the maximal allowable capacity of $A(t)$, the adaptive grid mechanism (Coello et al., 2004) is applied. Then, if the counter exceeds a maximal value which is set to *NA*=20 in this paper, Levy flight is self-adaptively activated. By employing Levy flight method to renew the particle's location, the new particle's location (Haklı and Uguz, 2014) is calculated as follows:

$$x_i^{(t+1)} = x_i^{(t)} + rand(size(D) \oplus Levy(\beta), \tag{9}$$

where $rand(size(D)$ is random digit for dimension of space. According to equation (9), particles can flee from local optima by broad jump, which enhance the swarm diversity, so that LF-MOPSO can improve the global search ability. In Levy flight, parameter β plays a significant role to change random distribution. According to characteristics of Levy flight, if β is set different values, the distribution will change accordingly. In this paper, the constant value for β is set to 1.5.

Finally, once a new non-dominated solution enters the empty grids near the current Pareto frontier, this means that the current Pareto frontier is closer to the Pareto optimal frontier, Levy flight stops automatically, the original MOPSO rerun. The pseudocode of LF-MOPSO is presented in Table 1.

4 Simulation results and discussions

In this section, two two-objective functions (ZDT1 and ZDT2) and one three-objective function (DLTZ1) (Deb et al., 2002) are used for evaluating the search capability of LF-MOPSO. To testify the advantages of the proposed method, LF-MOPSO is contrasted with three recently developed MOPSOs, that is, MOPSO (Coello et al., 2004), MOLS-MOPSO (Xu et al., 2015) and AgMOPSO (Zhu et al., 2018). On the same test problem, each algorithm is simulated for twenty times to study its statistical performance.

4.1 Performance metrics

To assess the search capability of LF-MOPSO, three different quantitative performance metrics (Deb et al., 2006) are hired in the simulation study.

1 *Generational distance* (*GD*): It is a good index of interval between resultant Pareto frontier (*PF*) and Pareto optimal frontier (PF_{true}). *GD* is expressed as

$$GD = \sqrt{\frac{1}{N}\sum_{i=1}^{N} r_i^2}\ , \tag{10}$$

where *N* is the amount of non-dominated solutions in *PF*, r_i is Euclidian distance between *i*th non-dominated solution in *PF* and the non-dominated solution in PF_{true}, which is closest to *i*th non-dominated solution in *PF*. The smaller *GD* value means that the Pareto frontier is better convergence.

2 *Spacing* (*SP*): It gives the degree of uniformity for the distribution of non-dominated solutions in the resultant Pareto frontier, *SP* is expressed as

$$S = \left[\frac{1}{N}\sum_{i=1}^{N}(d_i - \bar{d})^2\right]^{\frac{1}{2}} / \bar{d}, \quad \bar{d} = \frac{1}{N}\sum_{i=1}^{N} d_i \tag{11}$$

The smaller *SP* value means that the distribution of solution in *PF* is more uniform.

3 *Maximal spread* (*MS*): It evaluates how much PF_{true} is overlapped by *PF* by using hyper-boxes which is constructed by the maximum and minimum function values obtained in PF_{true} and *PF*. *MS* is expressed as

$$MS = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left[\frac{\min(g_{iM}, G_{iM}) - \max(g_{im}, G_{im})}{G_{iM} - G_{im}}\right]^2}\ , \tag{12}$$

where *m* is objective amount in a multi-objective function. For the *i*th objective in *PF*, g_{iM} is the maximal value and g_{im} is the minimal value. Moreover, For the *i*th objective in PF_{true}, G_{iM} is the maximal value and G_{im} is the minimal value. The higher the value for *MS*, the wider the scope of the solutions.

4.2 Parameter settings

In addition to using the personality parameters shown in their original papers, MOPSO, MOLS-MOPSO, AgMOPSO and LF-MOPSO have three common parameters. Population size is 100; maximal allowable capacity for $A(t)$ is 100 on two-objective functions and 200 on three-objective functions, respectively; the maximal iteration number is 3000.

4.3 Experimental results

Tables 2–4 present average and standard deviation of *GD*, *SP* and *MS* metrics for MOPSO, MOLS-MOPSO, AgMOPSO and LF-MOPSO on three multi-objective functions, respectively. For presenting the convergence, distribution and diversity of the non-dominated solutions in the last Pareto frontier, Figures 1–3 show the resultant Pareto frontier generated by MOPSO, MOLS-MOPSO, AgMOPSO and LF-MOPSO on three benchmark functions in an arbitrary run.

Table 2 Average (A) and standard deviation (SD) for *GD* on test problems

Approach	*ZDT1*	*ZDT2*	*DLTZ2*
MOPSO (A)	0.00281	0.00521	0.11824
(SD)	0.00865	0.00872	0.00042
MOLS-MOPSO (A)	0.00253	0.00339	0.04217
(SD)	0.00764	0.00328	0.00037
AgMOPSO (A)	0.00305	0.00436	0.03928
(SD)	0.00453	0.01456	0.00055
LF-MOPSO (A)	0.00225	0.00129	0.01723
(SD)	0.00231	0.00087	0.00018

Table 3 Average (A) and standard deviation (SD) for *SP* on test problems

Approach	*ZDT1*	*ZDT2*	*DLTZ2*
MOPSO (A)	0.73291	0.42753	0.78243
(SD)	0.00049	0.00326	0.00835
MOLS-MOPSO (A)	0.61974	0.22535	0.35382
(SD)	0.00062	0.00439	0.00544
AgMOPSO (A)	0.93283	0.32862	0.32845
(SD)	0.00275	0.00382	0.00738
LF-MOPSO (A)	0.49267	0.24298	0.30324
(SD)	0.00023	0.00224	0.00239

Table 4 Average (A) and standard deviation (SD) for *MS* on test problems

Approach	*ZDT1*	*ZDT2*	*DLTZ2*
MOPSO (A)	0.96749	0.98435	0.99548
(SD)	0.02387	0.00374	0.00725
MOLS-MOPSO (A)	0.98492	0.99342	0.99783
(SD)	0.00982	0.00297	0.00628
AgMOPSO (A)	0.99021	0.99452	0.99921
(SD)	0.00993	0.00245	0.00426
LF-MOPSO (A)	0.99873	0.99967	0.99978
(SD)	0.00382	0.00093	0.00167

4.3.1 GD index comparison

From Table 2, first, LF-MOPSO can obtain better average and standard deviation of *GD* than other three MOPSOs on ZDT1 and ZDT2, respectively. Especially for ZDT2, the average of *GD* gained by LF-MOPSO is much better than those of other three algorithms.

Second, LF-MOPSO surpasses other three algorithms in the matter of the average and standard deviation of *GD* on DLTZ2 too. It can be found out that LF-MOPSO is better than other three MOPSOs in the matter of the value of *GD*. The results show that Levy flight has played an important role in LF-MOPSO. According to the comparison results, it is easy to see that LF-MOPSO is the closest to the optimal Pareto frontier. Moreover, from Figures 1–3, LF-MOPSO encloses the entire frontier nearly, it means that LF-MOPSO can achieve the best convergence. Therefore, it is obvious from Table 2 that LF-MOPSO is very competitive in comparison with MOPSO, MOLS-MOPSO and AgMOPSO in terms of convergence.

Figure 1 Pareto frontiers produced by: (a) MOPSO; (b) MOLS-MOPSO; (c) AgMOPSO and (d) LF-MOPSO on ZDT1

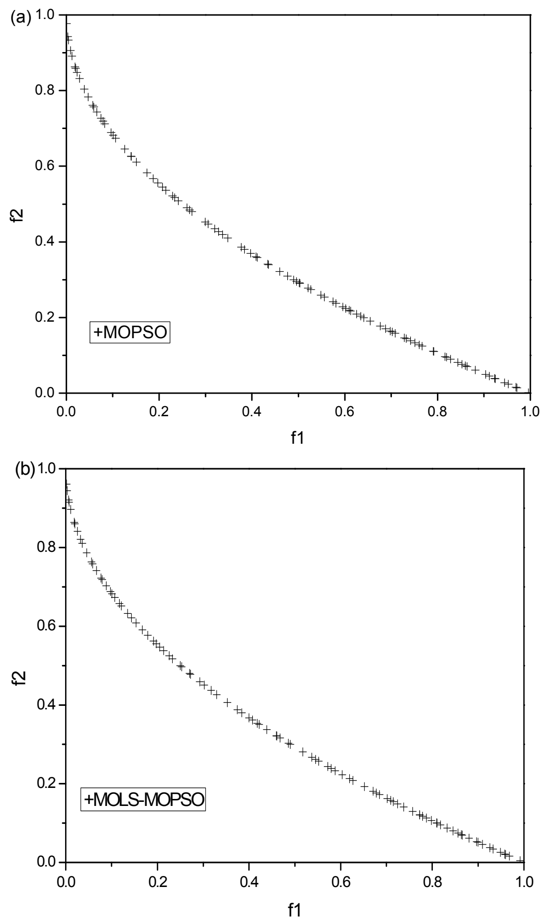
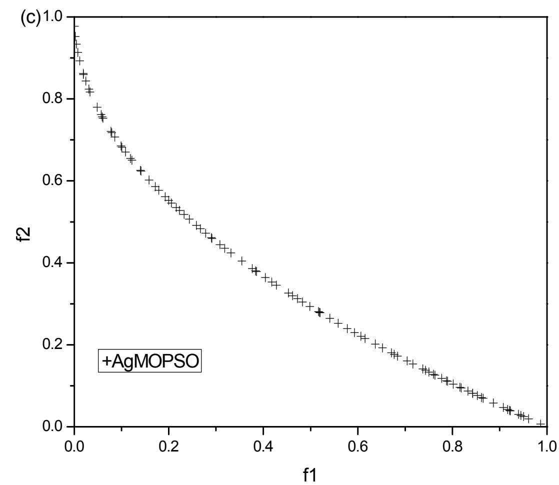
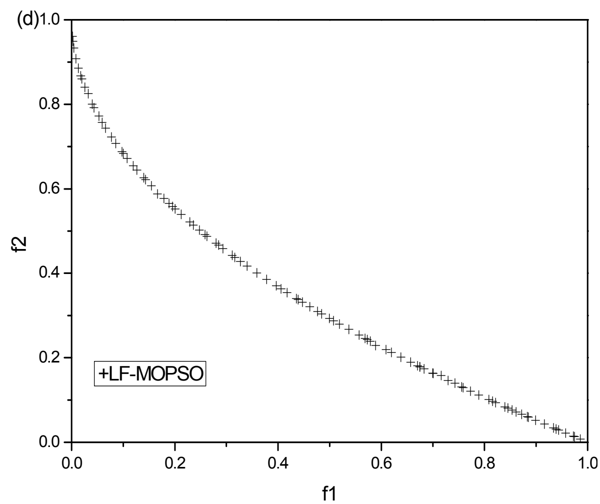


Figure 1 Pareto frontiers produced by: (a) MOPSO; (b) MOLS-MOPSO; (c) AgMOPSO and (d) LF-MOPSO on ZDT1 (continued)

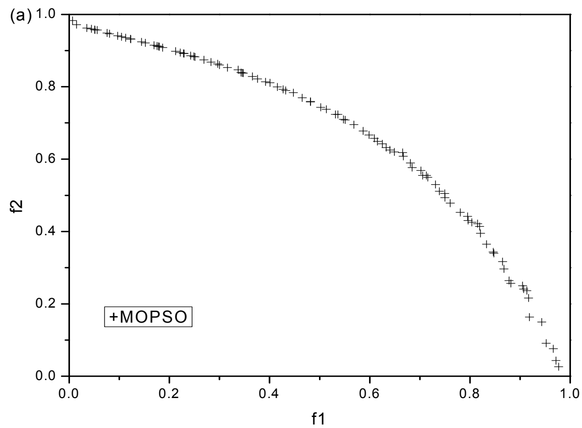


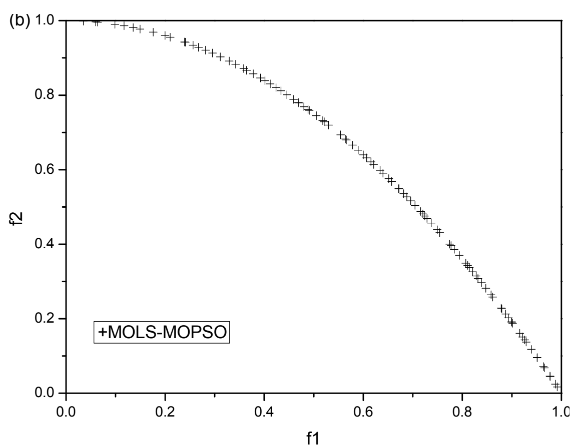


4.3.2 SP index comparison

The *SP* comparison between LF-MOPSO and other three compared algorithms is shown in Table 3. On the one hand, it is obvious that LF-MOPSO can own better average and standard deviation for *SP* than other three algorithms on ZDT1, moreover, LF-MOPSO far surpasses AgMOPSO. LF-MOPSO slightly performs worse than MOLS-MOPSO on ZDT2. Although LF-MOPSO does not perform the best for *SP* metric, it is very close to the best and ranks second; on the other hand, LF-MOPSO has the best the values of *SP* on DTLZ2 than other three algorithms, specially, the average of *SP* obtained by LF-MOPSO is much better than that of MOPSO. Therefore, on the basis of the above analysis, the *SP* performance of LF-MOPSO is better than other three algorithms on the most ZDTs and DTLZs.

Figure 2 Pareto frontiers produced by: (a) MOPSO; (b) MOLS-MOPSO; (c) AgMOPSO and (d) LF-MOPSO on ZDT2





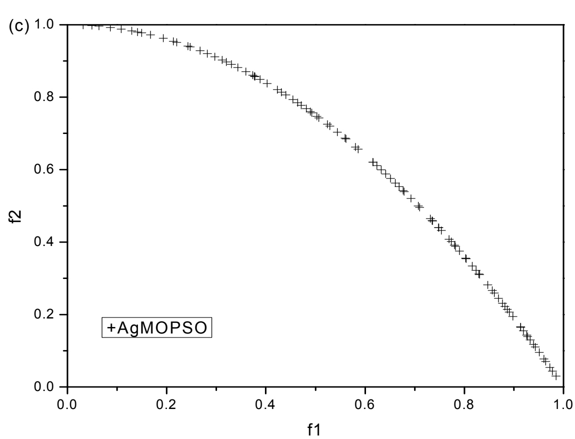


Figure 2 Pareto frontiers produced by: (a) MOPSO; (b) MOLS-MOPSO; (c) AgMOPSO and (d) LF-MOPSO on ZDT2 (continued)

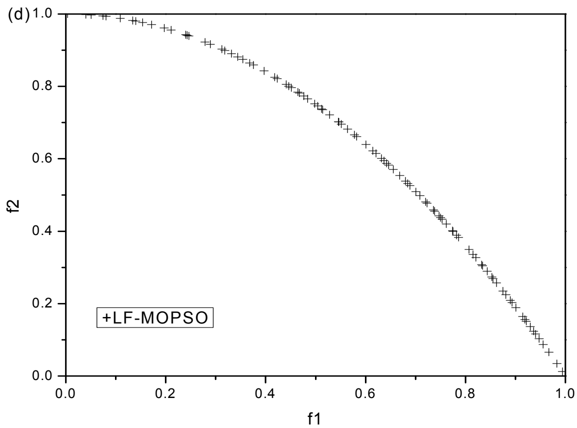
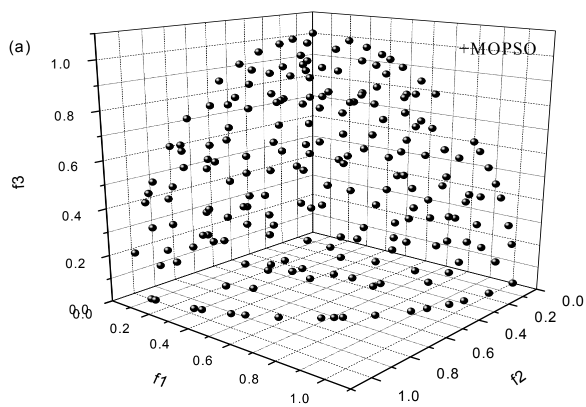


Figure 3 Pareto frontiers produced by: (a) MOPSO; (b) MOLS-MOPSO; (c) AgMOPSO and (d) LF-MOPSO on DLTZ2



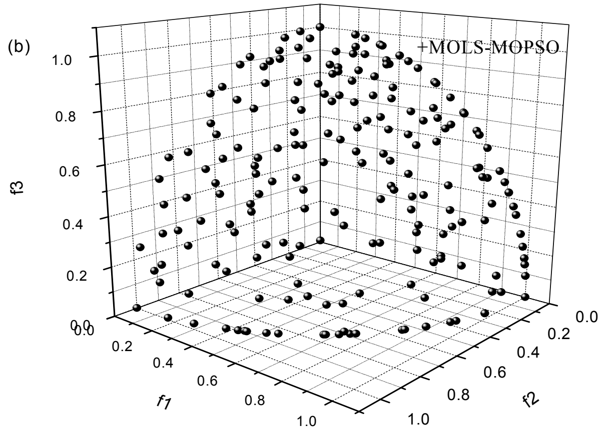
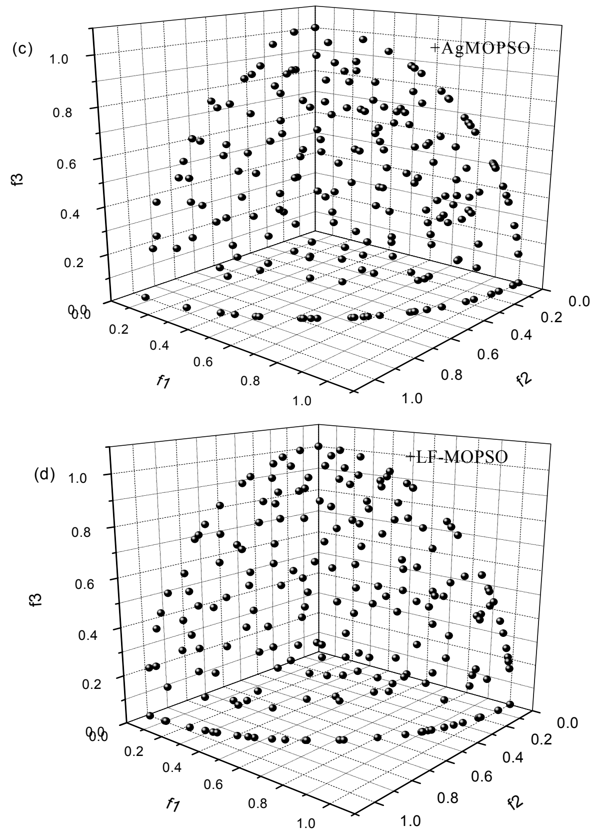


Figure 3 Pareto frontiers produced by: (a) MOPSO; (b) MOLS-MOPSO; (c) AgMOPSO and (d) LF-MOPSO on DLTZ2 (continued)



4.3.3 MS index comparison

The average and standard deviation of *MS* metric on three test problems are presented in Table 4. According to Table 4, first, for the average of *MS* metric, the order of four algorithms is LF-MOPSO, MOLS-MOPSO, AgMOPSO, MOPSO on ZDT1; the order of four algorithms is LF-MOPSO, AgMOPSO, MOLS-MOPSO, MOPSO on ZDT2. Obviously, LF-MOPSO can obtain better average and standard deviation of *MS* metric than other three MOPSOs on ZDT1 and ZDT2. Second, LF-MOPSO gains the best results among the four compared algorithms in the matter of average and standard deviation of *MS* metric on DTLZ2. For the average of *MS* metric, the order of four algorithms is LF-MOPSO, AgMOPSO, MOLS-MOPSO, MOPSO. The experimental results make known that LF-MOPSO has best spread of non-dominated solutions on ZDT1, ZDT2 and DTLZ2, respectively. By the results of *MS* in Table 4, LF-MOPSO can get a non-dominated solution set with better spread of solutions than other three algorithms.

4.4 Discussions

The aim of LF-MOPSO is to make the last Pareto frontier approach the Pareto optimal frontier better, moreover, a solution set in which the non-dominated solutions are diverse and well-distributed is obtained. According to the above analysis, it obvious that LF-MOPSO can obtain the best results among the four MOPSOs considered, Figures 1–3 also proves the conclusion. The good performance of LF-MOPSO may be attributed to Levy flight and the self-adaptive conversion mechanism.

In LF-MOPSO, The population diversity reduces when the original MOPSO runs. Levy flight is used to prevent premature convergence and improve the population diversity. Once the diversity reaches the lower bound conditions by the original MOPSO, a self-adaptive conversion mechanism activates Levy flight. When the diversity reaches the upper bound conditions, the self-adaptive conversion mechanism activates the original MOPSO and Levy flight stops. The self-adaptive conversion mechanism is designed to activate the original MOPSO and Levy flight periodically. Therefore, LF-MOPSO can keep the swarm diversity periodically and retain the balance between Levy flight and the original MOPSO during the iteration.

Experimental results present that Levy flight is more suitable for MOPSO to maintain the swarm diversity because Levy flight can get better exploration effect in search space. However, the original MOPSO is more inclined to exploit the search space, which may lead to the premature convergence. By employing the self-adaptive conversion mechanism, Levy flight and the original MOPSO can switch automatically. Therefore, LF-MOPSO realises the periodic transformation from the exploration to the exploitation, which make better balance between the shrinkage and divergence during the iteration, so that LF-MOPSO avoids premature convergence and a solution set in which the non-dominated solutions are diverse and well-distributed is acquired. Therefore, the combination of Levy flight and the original MOPSO improves the performance of LF-MOPSO. Generally speaking, the proposed algorithm improves the performs on the test problems used.

5 Conclusions

In this paper, a new method, called LF-MOPSO, is proposed. The present method which integrates Levy flight and the self-adaptive conversion mechanism into the original MOPSO is used for dealing multi-objective problems. In LF-MOPSO, the self-adaptive conversion mechanism can realise the periodical conversion between the original MOPSO and Levy flight. The original MOPSO has exploitation to improve the small scale search abilities, and Levy flight has wide-ranged exploration to maintain the population diversity. The balance is kept between the shrinkage and divergence during the iteration. LF-MOPSO is compared with three recently improved MOPSOs on three multi-objective benchmark functions, including two two-objective functions and one three-objective function. Based on three performance indicators, the experimental results show that LF-MOPSO is not only closer to the Pareto optimal frontier, also a solution set in which the non-dominated solutions are diverse and well-distributed is obtained.

Acknowledgement

The authors wish to acknowledge the National Nature Science Foundation of China (Grant 61866023), the Science and Technology Plan Project of Jiangxi Provincial Education Department China (Grant GJJ12093) and Nanchang University Fund China (Grant 13002490) for the financial support.

References

Chechkin, A.V., Metzler, R., Klafter, J. and Gonchar, V.Y. (2008) 'Introduction to the theory of Lévy flights', in Klages R, Radons, G. and Sokolov, I.M. (Eds.): *Anomalous Transport: Foundations and Applications*, John Wiley & Sons, pp.129–162.

Chen, J.Y., Lin, Q.Z. and Zhen, J. (2010) 'A hybrid immune multiobjective optimization algorithm', *European Journal of Operational Research*, Vol. 204, pp.294–302.

Coello, C.A.C. and Lechuga, M.S. (2002) 'MOPSO: a proposal for multiple objective particle swarm optimization', *Proceedings of IEEE Congress on Evolutionary Computation*, Honolulu, HI, USA, No. 2, pp.1051–1056.

Coello, C.A.C., Pulido, G.T. and Lechuga, M.S. (2004) 'Handling multiple objectives with particle swarm optimization', *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 3, pp.256–279.

Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2006) 'A fast and elitist multi-objective genetic algorithm: NSGA-II', *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, pp.182–197.

Deb, K., Thiele, L., Laumanns, M. and Zitzler, E. (2002) 'Scalable multi-objective optimization test problems', *Congress on Evolutionary Computation*, Honolulu, HI, USA, pp.825–830.

Garg, A., Singh, S., Gao, L., Xu, M.J. and Tan, C.P. (2020) 'Multi-objective optimisation framework of genetic programming for investigation of bullwhip effect and net stock amplification for three-stage supply chain systems', *International Journal of Bio-Inspired Computation*, Vol. 16, No. 1, pp.241–251.

Haklı, H. and Uguz, H. (2014) 'A novel particle swarm optimization algorithm with levy flight', *Applied Soft Computing*, Vol. 23, pp.333–345.

Han, H.G., Lu, W., Zhang, L. and Qiao, J.F. (2018) 'Adaptive gradient multiobjective particle swarm optimization', *IEEE Transactions on Cybernetics*, Vol. 48, No. 11, pp.3067–3079.

Hu, Y. and Gong, D.W. (2021) 'Multiobjective particle swarm optimization for feature selection with fuzzy cost', *IEEE Transactions on Cybernetics*, Vol. 51, pp.874–888.

Kazuhiro, I., Shinji, N., Masataka, Y. and Masahiko, N. (2008) 'Enhanced multiobjective particle swarm optimization in combination with adaptive weighted gradient-based searching', *Engineering Optimization*, Vol. 40, No. 9, pp.789–804.

Kennedy, J. and Eberhart, R.C. (1995) 'Particle swarm optimization', *Proceedings of the IEEE International Conference on Neural Networks*, Perth, WA, Australia, pp.1942–1948.

Li, Y.Y., Tian, M.Z., Wang, Y., Qingfu Zhang, Q.F., Saxena, D.K. and Jiao, L.C. (2020) 'A new replica placement strategy based on multi-objective optimisation for HDFS', *International Journal of Bio-inspired Computation*, Vol. 16, No. 1, pp.13–22.

Liu, X.G., Du, Y.Z. and Zeng, X.X. (2020) 'Multiobjective particle swarm optimization based on network embedding for complex network community detection', *IEEE Transactions on Computational Social Systems*, Vol. 7, pp.437–449.

Niu, B., Liu, J. and Tan, L.J. (2019) 'Multi-swarm cooperative multi-objective bacterial foraging optimisation', *International Journal of Bio-Inspired Computation*, Vol. 13, No. 1, pp.21–31.

Wang, H., Sun, H., Li, C., Rahnamayan, S. and Pan, J. (2013) 'Diversity enhanced particle swarm optimization with neighborhood search', *Information Sciences*, Vol. 223, pp.119–135.

Wang, H., Wang, W.J., Cui, L.Z., Sun, H., Zhao, J., Wang, Y. and Yu Xue, Y. (2018) 'A hybrid multi-objective firefly algorithm for big data optimization', *Applied Soft Computing*, Vol. 69, pp.806–815.

Xu, C.H., Ding, A.S. and Liao, S.S. (2020a) 'A privacy-preserving recommendation method based on multi-objective optimisation for mobile users', *International Journal of Bio-Inspired Computation*, Vol. 16, No. 1, pp.23–32.

Xu, G., Luo, K., Jing, G.X., Yu, X., Ruan, X.J. and Song, J. (2020b) 'On convergence analysis of general multi-objective particle swarm optimization algorithm', *European Journal of Operational Research*, Vol. 286, No. 1, pp.32–38.

Xu, G., Yang, Y.Q., Liu, B.B., Xu, Y.H. and Wu, A.J. (2015) 'An efficient hybrid multi-objective particle swarm optimization with a multi-objective dichotomy line search', *Journal of Computational and Applied Mathematics*, Vol. 280, pp.310–326.

Xue, F. and Wu, D. (2020) 'NSGA-III algorithm with maximum ranking strategy for many-objective optimisation', *International Journal of Bio-Inspired Computation*, Vol. 15, No. 1, pp.14–23.

Xue, Y., Tang, Y.H., Xu, X. and Liang, J.Y. (2021) 'Multi-objective feature selection with missing data in classification', *IEEE Transactions on Emerging Topics in Computational Intelligence,* doi: 10.1109/TETCI.2021.3074147.

Zhang, X.W., Liu, H. and Tu, L.P. (2020a) 'A modified particle swarm optimization for multimodal multi-objective optimization', *Engineering Applications of Artificial Intelligence*, Vol. 95, pp.103905.

Zhang, X.W., Liu, H., Tu, L.P. and Zhao, J. (2020b) 'An efficient multi-objective optimization algorithm based on level swarm optimizer', *Mathematics and Computers in Simulation*, Vol. 177, pp.588–602.

Zhu, Q.L., Lin, Q.Z., Chen, W.N., Wong, K.C., Coello, C.C.A., Li, J.Q., Chen, J.Y. and Zhang, J. (2018) 'An external archive-guided multi-objective particle swarm optimization algorithm', *IEEE Transactions on Cybernetics*, Vol. 47, No. 9, pp.2794–2807.