Adaptive neighbourhood size adjustment in MOEA/D-DRA

Meng Xu

Complex System and Computational Intelligence Laboratory,
Taiyuan University of Science and Technology,
Taiyuan, China
and
College of Computer Science and Technology,
Beijing University of Technology,
Beijing, China
Email: mengxu0805@163.com

Maoqing Zhang, Xingjuan Cai* and Guoyou Zhang

Complex System and Computational Intelligence Laboratory,
Taiyuan University of Science and Technology,
Taiyuan, China
Email: maoqing_zhang@163.com
Email: xingjuancai@163.com
Email: guoyouzh@sohu.com
*Corresponding author

Abstract: Multi-objective optimisation algorithm based on decomposition (MOEA/D) is a well-known multi-objective optimisation algorithm, which was widely applied for solving multi-objective optimisation problems (MOPs). MOEA/D decomposes a multi-objective problem into a set of scalar single objective sub-problems using aggregation function and evolutionary operator. A further improved version of MOEA/D with dynamic resource allocation strategy (MOEA/D-DRA) has exhibited outstanding performance on CEC2009 in terms of the convergence. However, it is very sensitive to the neighbourhood size. In this paper, a new enchanted MOEA/D-ANA strategy based on the adaptive neighbourhood size adjustment (MOEA/D-ANA) was presented to increase the diversity, which mainly focuses on the solutions density around sub-problems. The experiment results demonstrate that MOEA/D-ANA performs the best compared with other five classical MOEAs on the CEC2009 test instances.

Keywords: MOEA/D; diversity; neighbourhood size; CEC2009 test instances.

Reference to this paper should be made as follows: Xu, M., Zhang, M., Cai, X. and Zhang, G. (2021) 'Adaptive neighbourhood size adjustment in MOEA/D-DRA', *Int. J. Bio-Inspired Computation*, Vol. 17, No. 1, pp.14–23.

Biographical notes: Meng Xu received his MS from Taiyuan University of Science and Technology. He is currently pursuing his PhD degree at the College of Computer Science and Technology, Beijing University of Technology (BJUT), Beijing, China. His main research areas are evolutionary computation, deep learning and medical image processing.

Maoqing Zhang is currently working toward his PhD in Control Science and Engineering from Tongji University. His current research interests include evolutionary computation, deep learning and their applications.

Xingjuan Cai received her PhD in Control Science and Engineering from the Tongji University, China in 2017. She researches interest includes computational intelligence, stochastic algorithm and combinatorial optimisation. She has published over 40 peer reviewed journal papers.

Guoyou Zhang received his PhD in Control Science and Engineering from the Lanzhou jiaotong University, China in 2013. He researches interest is computational intelligence. He has published over ten peer reviewed journal papers.

1 Introduction

Multi-objective optimisation problems are the problems with two or three objectives, which can be seen in many research areas (Ma, 2018; Arloff et al., 2018), such as: engineering economics, image processing, scheduling and so on. In complex application, two or three objectives problem than one objective is more suitable for the practical requirement. However, these objectives in MOPs often conflict with each other and there does not exist one solution which can minimise all objectives at the same time. To tackle the problem, a set of trade-off candidate solutions are usually needed to find the approximate Pareto front in MOPs.

In the past decades, many efficient algorithms have been proposed, such as pigeon-inspired optimisation algorithm (Zhang et al., 2019), bat algorithm (Cai et al., 2016; Cui et al., 2018a; Cai et al., 2018), cuckoo search algorithm (Niu et al., 2018; Zhang et al., 2018b; Abdel-Baset et al., 2018) firefly algorithm (Yu and Feng, 2018; Wang et al., 2017; Lv et al., 2018), particle swarm optimisation (Bougherara et al., 2018), artificial bee colony algorithm (Amiri and Dehkordi, 2018). In addition, they have been successfully applied into different practical issues (Cui et al., 2017a, 2017b, 2018b). However, these methods are population-based algorithms and exhibit less efficient performance than evolutionary algorithms, which are more applicable to MOPs.

Researchers have contributed a lot to multi-objective evolutionary algorithms (MOEAs). Three frameworks have become mainstreams in MOEAs:

1 MOEAs based on dominance, like NSGA-II (Deb et al., 2002) and SPEA2 (Zitzler et al. 2001), they select and update individuals based on dominant relationship among the solutions;

2 MOEAs based on indicator, like HypE (Emmerich et al., 2005; Bader and Zitzler, 2014), are proposed to guide a fast search to approximate the real hypervolume value (HV). In addition, IBEA (Wagner et al., 2007) is also a MOEA based on indicator.

3 MOEAs based on decomposition, like MOEA/D (Zhang and Li, 2007), are based on the one decomposition approach such as the TCH, which convert MOPs into a number of scalar optimisation sub-problems. Different from other MOEAs, MOEA/D has less computational cost because all the sub-problems are solved simultaneously.

A neighbourhood relation defined among the sub-problems to help to increase efficiency. For each sub-problem, its neighbourhood relation consists of a certain amount of sub-problems. Compared with other algorithms, the most obvious features of MOEA/D are that it not only replaces itself, but also replaces its neighbourhood. It is a co-evolution mechanism. In the course of evolution, the 'good pheromone' from neighbourhood serves to precipitate algorithmic rate of convergence.

MOEA/D has shown great performance in CEC2009 competitions (Zhang et al., 2008). After that, a growing number of researchers have made contributions to its improvements and applications. For instance, Li and Zhang (2009) proposed MOEA/D-DE, which used differential evolution (DE) operator (Dong et al., 2017; Wang and Wang, 2018) to replace simulated binary crossing operator at the level of evolution operation. Further, to increase diversity, Li and Zhang employed the controlling parameter to control maximum number of replacement solutions. Zhang et al. (2009) employed dynamic resource allocation to compute utility rate of different sub-problems so as to decrease computation complexity and improve the performance of MOEA/D, which is called MOEA/D-DRA. Since then, researchers' major works focus on balancing exploration, exploitation and designing adaptation strategies. In addition to the above evolution operator and dynamic resource allocation, there are still some improvement strategies which are worth learning:

1 Adjustment weight vectors: like UMOEA/D (Tan et al., 2013), it is a strategy that designs uniform distribution weight vector rather than simplex lattice structure. MOEA/D-AWA (Guo et al., 2016), is an adaptive weight vector adjustment strategy, based on the geometric relationship of weights and corresponding solutions;

2 Matching selection mechanism: MOEA/D-SMS (Li et al., 2014) is first proposed to establish a stable matching relationship between solutions and sub-problems. MOEA/D-IR (Li et al., 2015), is proposed to build an interrelationship between sub-problems and solutions based on the mutual preference;

3 Neighbourhood size selection: for instance, MOEA/D-ENS (Zhao et al., 2012), employed an ensemble of different neighbourhood size (NS) based on the probability selection.

Also, there are some other improvement strategies. MOEA/D-M2M (Liu et al., 2014) decomposes one MOPs into many multi-objective sub-problems, which is similar to the method of cluster (Narasimhan et al., 2018, Reddy and Panigrahi, 2017). The adaptive reference point designs (Wang et al., 2017), investigates the effect of reference point specified in three representative manners: optimistic, pessimistic and dynamic. Moreover, MOEA/D has been successfully applied to some other areas, like Wireless Sensor Networks (Konstantinidis et al., 2009), image edge segment (Wang, Li et al. 2014), antenna design (Ding and Wang, 2013) and so on.

The framework of MOEA/D and the later improved versions are sensitive to neighbourhood size. A proper neighbourhood size is very important for improving the performance of the algorithm. Hence, we propose a new idea to adaptively adjust neighbourhood size and call the algorithm MOEA/D-ANA. MOEA/D-ANA makes full use of the density relationship between sub-problems and solutions to adaptively change neighbourhood size. During the initialisation, the neighbourhood size of sub-problems is a fixed value. At each generation, the neighbourhood size of sub-problems is no longer a default value and each sub-problem has different neighbourhood size.

In this paper, we design an adaptive adjustment neighbourhood size strategy for MOEA/D-DRA. Our main concerns are unconstraint MOPs. The paper is organised as follows. In Section 2, the related background of our work is presented, which consists of three parts: the basic definition of MOPs, a briefly introduction of decomposition approach and neighbourhood and the details of MOEA/D-DRA. In Section 3, we improved MOEA/D-DRA with adaptive adjustment neighbourhoods. Test instances and performance measures are introduced in Section 4. Finally, brief discussion and conclusion are studied with some future research.

2 Related background

2.1 Basic definition of MOPs

Multi-objective optimisation problems (MOPs) can be defined as follows:

$$\begin{aligned} &\textit{minimize} \quad L(x) = \left(l_1(x), l_2(x), \cdots, l_m(x)\right)^T \\ &\textit{subject to} \quad x \in \Omega \end{aligned} \tag{1}$$

where Ω is the decision space (variable), $\Omega \in P^n$, $x = (x_1, x_2, \ldots, x_n)^T \in P$ is a candidate solution. $L{:}\Omega \to P^m$ constitutes m objectives functions.

Pareto-optional: A solution x^* is said to be Pareto-optional if and only if $\neg\exists x \in X{:}\ x \succ x^*$.

Pareto-dominance: A solution x8, if there is no other solution x which can dominate x*, as defined by $\forall$i = 1, 2, …, n.

Pareto front (PF): The set of all the Pareto optimisation solution constitutes curved surface.

2.2 The decomposition approach and neighbourhood concept

Since Zhang proposed MOEA/D in 2007 (Zhang and Li, 2007), MOEA/D has been wildly mentioned by researchers. Two pieces of details are indispensable elements in MOEA/D: decomposition approach and neighbourhood. These characteristics are essential for improving algorithm.

- Decomposition

In MOEA/D, some decomposition approaches have been used, such as the weight aggregation approach, the Tchebycheff aggregation approach and the PBI aggregation approach (Shukla, 2007). The Tchebycheff aggregation approach is wildly used in variants of MOEA/D and it can be defined as follows:

$$\begin{aligned} &\textit{minimize} \quad g\left(x \mid \lambda, z^*\right) = \max\left\{\lambda_i \left|l_i(x) - z_i^*\right|\right\} \\ &\textit{subject to} \quad x \in \Omega \end{aligned} \tag{2}$$

where $\lambda = (\lambda_1, \lambda_2, \cdots, \lambda_m)$ is the weight vector, $\lambda_i \geq 0$ and $\sum_{i=1}^{m} \lambda_i = 1$, z^* is an ideal point. z^* is difficult to solve. Thus, it is replaces with z. z is the minimum value of each objective and $z = \min\{l_i(x) | x \in \Omega\}$ for each $i = 1, 2, \cdots, m$. MOEA/D decomposes a multi-objectives optimisation problems by the Tchebychedd aggregation approach. N weight vectors correspond to N sub-problems. The N weight vectors stands for N sub-problems and they are uniformly distributed on the PF. It has been proven that more uniform distribution of the weight vectors can ensure better diversity of the obtained solutions (Qi et al., 2014).

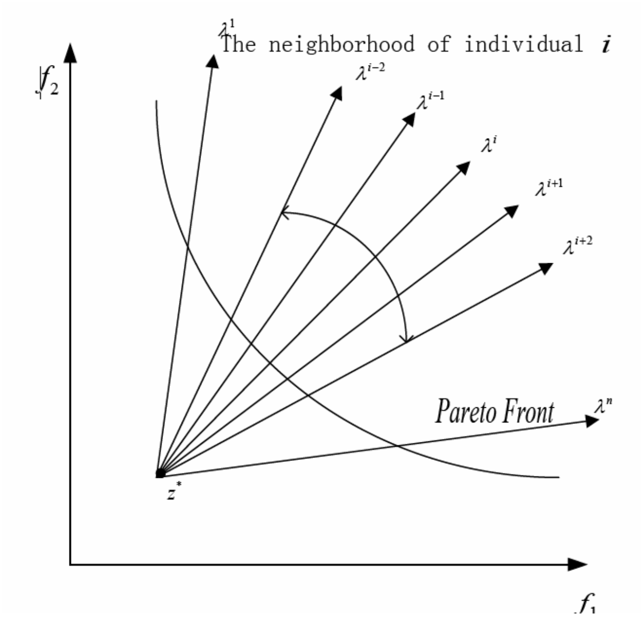
- Neighbourhood

MOEA/D defines the concept of the neighbourhood. Neighbourhood concept defines that distance between neighbouring weight vectors is close and the optimal solution is similar. In other words, two sub-problems are neighbours if their weight vectors are close in geometrics. For a sub-problem, its neighbourhood defines the T closest weight vectors to the weight vectors of all sub-problems on account of Euclidean distance as Figure 1. For simple computation, we think each sub-problem is the neighbour of themselves. In the version of MOEA/D-DE (Li and Zhang, 2009), the new solution substitutes a few of the best solutions of neighbourhood to increase convergence.

2.3 The framework of MOEA/D-DRA

In this paper, the proposed MOEA/D-ANA is derived from the MOEA/D-DRA, which is the new version of MOEA/D with dynamic resource allocation frame, in the runtime, the algorithm just computes the sub-problems with the high utility to increase convergence. The algorithm exhibits outstanding performance on the CEC2009 (Li and Zhang, 2009).

Figure 1 The neighbourhood of individual *i*



The difference between proposed MOEA/D-ANA and MOEA/D-DRA only lies in the selection neighbourhood size process. The framework of MOEA/D-DRA is presented in Algorithm 1 and in the next section, we will introduce the MOEA/D-ANA in detail.

Algorithm 1 MOEA/D-DRA framework

Step 1 Initialisation: Generate weight vectors of sub-problems, define neighbourhood, initialise population and update the ideal point.

Step 2 Update: Update the utility function and apply the genetic operator to select solutions by the utility function, then generate a new solution y and evaluate *F*(*y*).

Step 3 Replace the ideal point and sub-problems' objective function.

The utility functions define the

$$\Delta^i = \frac{g\left(x^i_{gen-50} \mid \lambda, z\right) - g\left(x^i_{new} \mid \lambda, z\right)}{g\left(x^i_{gen-50} \mid \lambda, z\right)} \tag{3}$$

The utility of individual of *i*:

$$\pi_i = \begin{cases} 1 & \text{if } \Delta^i > 0.001 \\ \left(0.95 + 0.05\dfrac{\Delta^i}{0.001}\right) & \text{otherwise} \end{cases} \tag{4}$$

Further details of MOEA/D-DRA can be found in (Zhang, Liu et al. 2009).

3 Adaptive neighbourhood size adjustment

As we can see from Figure 2, convergence and diversity are two major goals in improving the performance of MOEA (Laumanns et al., 2014). In the utopian condition, the individuals should quickly converge to the true Pareto Front and have a uniform distribution in the whole evolution process. In the MOEAs, convergence and diversity need to be balanced. On one hand, if individual pays more attention to the convergence, it will lose the diversity on the all search space. On the other hand, if it spends more time on the diversity, maybe the accuracy is not enough and it cannot be able to get close to the true Pareto very well. So, the current variants of MOEA/D consider balancing these two goals.

However, MOEA/D-DRA computes the utility of objective function by aggregating function results of each sub-problem, as shown in Figure 3. And it considers the convergence and ignores the diversity of each sub-problem. In the ideal situation, each individual should just correspond to one sub-problem. Many individuals are around them. The individuals stand for the solutions. This situation results in that more time is wasted in improving disadvantageous solutions of sub-problems and that good solutions is ignored. It will inevitably block the accuracy of the algorithm and may affect the performance of algorithms.

To improve the performance of MOEA/D-DRA, we propose a strategy of adaptive neighbourhood size adjustment based on former algorithm, MOEA/D-ANA.

Figure 2 Two goals of convergence and diversity

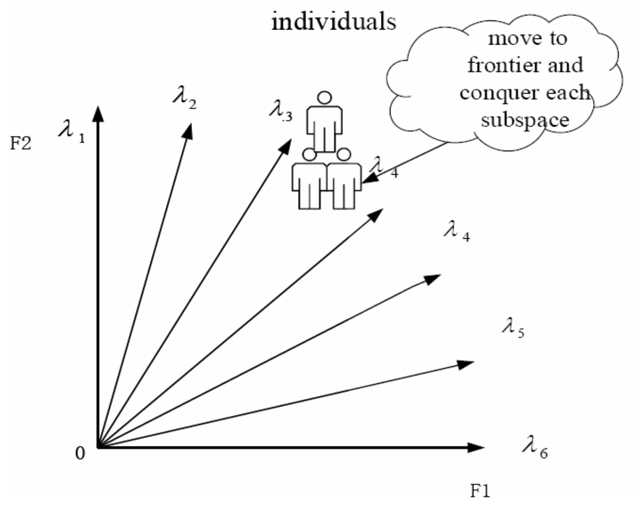
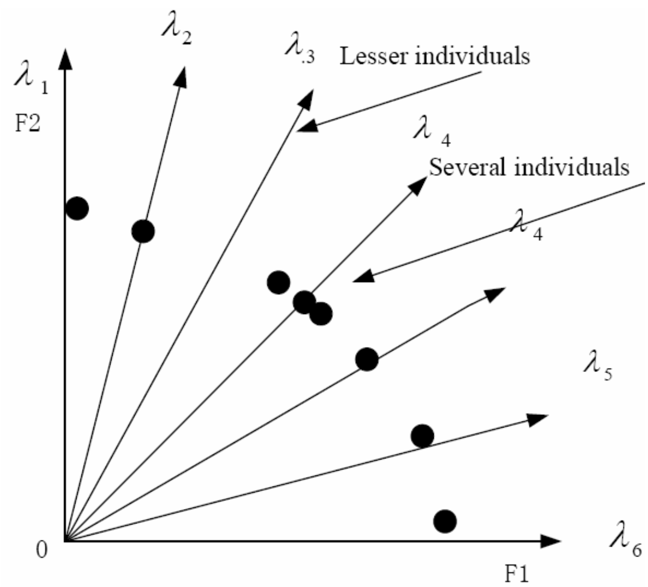


Figure 3 Reproducing the sub-problems in MOEA/D-DRA



Notes: For λ_3 sub-problem, the no individuals are around. For λ_4 sub-problem, several individuals are together.

In MOEA/D-ANA, we hope the neighbourhood size could be adaptively adjusted. In the initial phase, we assume that general neighbourhood size is the same with the neighbourhood size in MOEA/D-DRA. During the evolution, for the sub-problem with no individual, we think this kind of sub-problem lacks more search space to find the suitable candidate solutions. For this reason, we wish the neighbourhood size could be larger than the general neighbourhood size. On the contrary, for the sub-problem with several individuals, we think these sub-problems are solved very well. In this case, we want the neighbourhood size be smaller than general sub-problems. Thus, we help the MOEA/D-ANA to deal with the poor solutions with less time and help the good solutions with higher accuracy.

How do we know the individual corresponds to the sub-problems one by one? In this paper, the individual density of sub-problems is considered (Li et al. 2014).

Figure 4 Standard neighbourhood size

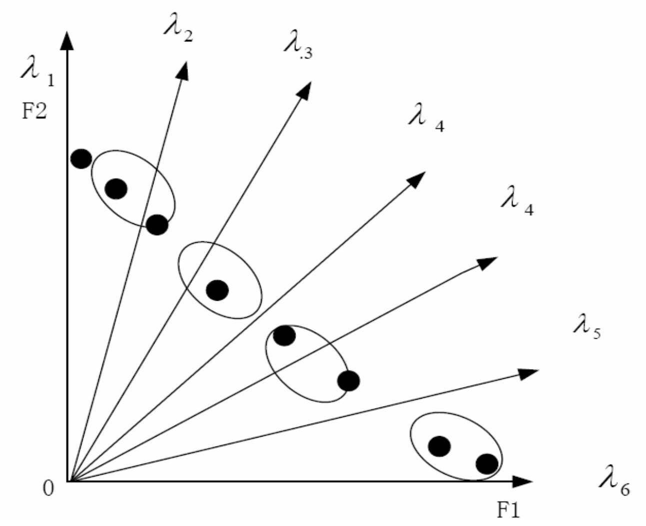
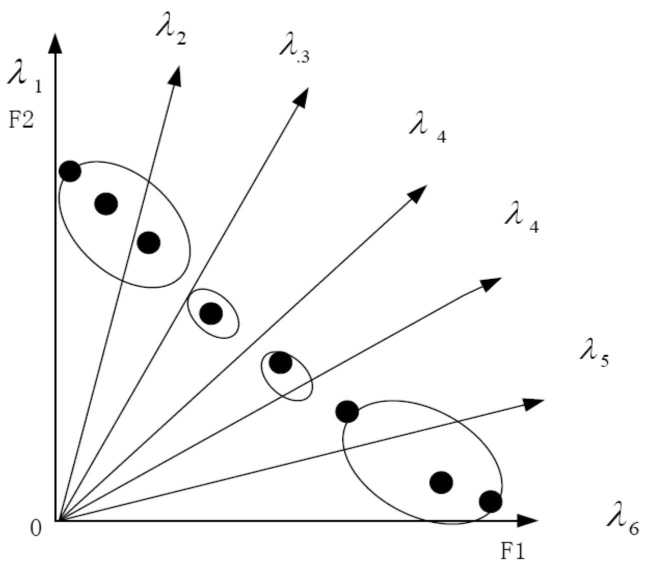


Figure 5 Adaptive neighbourhood size adjustment



For this problem, we think that when one individual is just within spatial distance of the sub-problem, the individual belongs to these sub-problems. While the weight vector of sub-problem corresponds to this sub-problem, we could compute the vertical distance from the individual to the sub-problem. If each individual is far from this sub-problem, we think no individual belongs to this sub-problem. In this case, we will enlarge the neighbourhood size to explore new search space. Otherwise, if the vertical distance of each individual to this sub-problem is close, we think several individuals belong to this sub-problem and we will narrow the range of neighbourhood to exploit the search space. As shown in Figures 4 and 5. The Algorithm 2 introduces this process, where *id* stands for the density of individual and $Ner(r^i)$ stands for the neighbourhood size of each sub-problems. In this paper, we think that the $id = 1$ and the general neighbourhood size is T; If $id = 0$, we think that the density of solutions is very low and we enlarge the neighbourhood size to increase the diversity. If $id > 1$, we think that the density of solutions is very high and we reduce the neighbourhood size to increase the convergence. The pseudo code of MOEA/D-ANA is as follows Algorithm 2 and Algorithm 3.

Algorithm 2 the framework of MOEA/D-ANA

Input

- N: the number of sub-problems.
- m: the number of objectives.
- The weight vectors of each sub-problem $\lambda = (\lambda_1, \lambda_2, \cdots, \lambda_N)^T$.
- T: the pool of nearest weight vectors in the neighbourhood $T = (T_1, T_2, \cdots, T^n)$

Output

- PS: $\{x_1, x_2, \cdots, x_n\}$.
- PF: $\{l(x_1), l(x_2), \cdots, l(x_n)\}$.

1 Initialisation:

1.1 To seek out T nearest weight vectors to sub-problems by computing the Euclidean distances of any two weight vectors. Set $Nei(r^i) = \{r_1, r_2, \cdots, r_T\}$. For each weight λ_i, the nearest weight vectors $\lambda_{r1}, \lambda_{r2}, \cdots, \lambda_{rT}$;

1.2 Initialise population, $x = \{x_1, x_2, \cdots, x_n\}$;

1.3 Initialise ideal point, $z = (z_1, z_2, \cdots, z_m)^T$, $z = \min_{1 \le j \le m, 1 \le i \le N} \{l_j(x^i)\}$;

1.4 Set the utility rates of each sub-problems $\pi_i =$, $gen = 0$;

2 Select sub-problems with high utility rate: Let all the sub-problems whose objectives are MOP individual objectives l_i form the initial I. By using 10-tournament selection based on π_i, select other $\left\lceil \frac{N}{5} \right\rceil - m$ indexes and add them to I.

3 For i $\in I$ do:

3.1 Mating selection

If $rand < \delta$

then $matingPool \leftarrow Nei(r^i)$

Else $matingPool \leftarrow \{1, 2, \cdots, N\}$'

Endif

3.2 Produce offspring

Randomly select three solutions x^{r_1}, x^{r_2} and x^{r_3} form *matingPool*. Then, they could generate a new offspring solution x_{new} with equation (3) to equation (5)

$$x_{new} = \begin{cases} x^{r_1} + F \times (x^{r_2} - x^{r_3}), & \text{if } rand < CR \\ x^{r_1} & \text{otherwise} \end{cases} \quad (5)$$

3.3 Evaluate the value of x_{new};

3.4 Update z: $j = 1, 2, \cdots, m$,

If $z_j \geq l_j(x_{new})$, then

$z_j = l_j(x_{new})$;

End if

3.5 Update solutions: set c= 0 and do the followings:

If $c = n$, or *matingPool* is empty, then

go to stop criteria;

Else if $g(x_{new} \mid \lambda, z) \leq g(x_r \mid \lambda, z)$ then

set $x_r = x_{new}$, $l_j = l(x_{new})$, $c = c + 1$;

End if

4 If mod (iteration, 50) == 0 then

Update the utility of each sub-problems with (3)(4)

End if

5 Update $Nei(r^i)$ (the details are presented in Algorithm 3).

return Step 3

End for

Algorithm 3 Strategy of adaptive neighbourhood size adjustment

Input the current point of each sub-problem $x = \{x_1, x_2, \cdots, x_n\}$; the weight vectors $\lambda_{r_1}, \lambda_{r_2}, \cdots, \lambda_{r_T}$;

Output $Ner(r^i)$

For $i = 1 \cdots N$;

$id^i = 0$;

$z^{nad} = \max_{1 \leq j \leq m, 1 \leq j \leq N} \{l_j(x^i)\}$

End for

For $i = 1 \cdots N$

For $j = 1 \cdots N$

Distance(i, j) = $D(x, \lambda_{r_i})$;

End for

$k = min$(Distance(i, j));

id^k ++

End for

If $id = 0$, then

$Ner(r^i) = Ner(r^i) + N$

Else if $id = 1$ then

$Ner(r^i) = Ner(r^i)$

Else

$Ner(r^i) = Ner(r^i) - N$

End if

where the vertical distance $D(x, \lambda_{r_i})$ is as follows:

$$D(x, \lambda_{r_i}) = \left\| \bar{L}(x) \frac{\lambda^T L(x) \lambda}{\lambda^T \lambda} \lambda \right\| \quad (6)$$

where $\bar{L}(x)$ is the normalised objective value and it can be defined as follows:

$$\bar{L}(x) : \bar{L}(x) = \frac{l(x) - z}{z - z^{nad}} \quad (7)$$

4 Experiment setting and results

In this section, the experiments are conducted to verify the proposed algorithm. The CEC2009 is the challenging instance. The CEC2009 benchmark (Zhang et al., 2008) is employed and there are ten instances in total. UF1–UF7 are bi-objective instances and UF8-UF10 are tri-objective instances. The dimension of the search space is 30 for all instances.

4.1 Experiment settings

Five classical multi-objective optimisation algorithms, including NSGA-II, HypE, MOEA/D, MOEA/D-DE and MOEA/D-DRA, are compared with MOEA/D-ANA. For a fair comparison, MOEA/D-DRA and MOEA/D-DE have the same parameters. They are all implemented in MATLAB. The parameter settings of the proposed MOEA/D-ANA are summarised as follows.

1 Setting produce offspring operator: the mutation rate $p_m = \frac{1}{n}$ and n = 20 (Zitzler et al., 2000). And for the DE operator, CR = 1 and F = 0.5.

2 We set the population sixe for UF1–UF7 to 600 and UF8–UF10 to 1,000.

3 Each algorithm will run 30 times independently.

4 Stopping condition: the function evaluation 300,000 for UF1–UF10.

5 Neighbourhood size: the neighbourhood size set $T = 0.1 \times N$ on MOEA/D-DE, MOEA/D-DRA, MOEA/D and neighbourhood size $T = 0.1 \times N$ and neighbourhood size pool $T = \{0.5 \times T, T, 1.5 \times T\}$. For how to select neighbourhood size, for more details, refer to Zhao et al. (2012).

6 For each sub-problems, the number of update neighbourhood's solutions is $n_r = 0.01 \times N$.

4.2 Performance metric

In the experiment part of this paper, the inverted generational distance (IGD) and hypervolume (HV) are two comprehensive indexes of diversity and convergence. So these metrics could be used to evaluate the performance of algorithms (Zitzler et al., 2003).

Assume p^* are the sample points of true Pareto, which are uniformly sampled. And x is the solutions obtained by algorithms. The IGD value is shown as the following:

$$IGD\left(x, p^*\right) = \frac{\sum_{x \in p^*} distance(x, p)}{\left|p^*\right|} \tag{8}$$

where $distance(x, p)$ is the minimum Euclidean distance between the x to the approximate to p. Therefore, the small IGD-metric value is, the better performance of the solutions is. We use the algorithm population as the p^* to compute IGD value. Each of UF1–UF7 has 1,000 uniformly distributed sample points. And each of UF8–UF10 has 10,000 sample points, like UK1–UF7.

Let $z' = \left(z_1', z_2', \cdots, z_m'\right)^T$. In objective space, z is a reference point dominated by Pareto objective vectors and x is the approximation points. It needs to meet $x_i \prec z$. So the metrics can be defined:

$$HV(x, z') = VOL\left(\bigcup_{i=1}^{N} \left[l_1(x), z'\right] \times \cdots \left[l_m(x), z'\right]\right) \tag{9}$$

where *vol*() is the Lebesgue measure (Zitzler and Thiele, 1999; Zitzler et al., 2000). In our measure, $z' = (2.0, 2.0)^T$ for UF1–UF7 and $z' = (2.0, 2.0, 2.0)^T$ for UF8–UF10.

4.3 Experiment result

In this part, ten instances of CEC2009 are tested. The IGD values of MOEA/D-ANA and other five MOEAs are listed in Table 1. IGD_mean is the mean of IGDs in 30 times. And IGD_std is the standard deviation of IGDs in 30 times. The best results are shows in boldface.

In Table 1, the IGD presents the final solutions of evolution process in conducted experiment. From this table, we can see the IGD results of MOEA/D-ANA and other five MOEAs. It is obvious that MOEA/D-ANA shows excellent performance on the CEC2009 test instances. MOEA/D-ANA is better than other algorithms on UF1–UF3, UF4 and UF6-UF8. And particularly for challenging UF4 and UF6, MOEA/D-ANA has better IGD values. While for the bi-objective instance UF5, with discrete points, both MOEA/D-ANA and MOEA/D-DRA show nearly the same performance. And for the tri-objective instances, UF9 and UF10, there are not significant differences between MOEA/D-ANA and MOEA/D-DRA, but both are not getting best performances. In addition, the MOEA/D-ANA get the best performance on UF8. In conclusion, the performance of MOEA/D-ANA in seven instances gets the better results than other algorithms.

Table 1 IGD result of MOEA/D-ANA and five other MOEAs on UF test instance

Instance	*Metrics*	*NSGA-II*	*HypE*	*MOEA/D*	*MOEA/D-DE*	*MOEA/D-DRA*	*MOEA/D-ANA*
UF1	IGD_mean	8.8534E-02	1.0062E-01	1.6171E-01	1.6576E-03	3.4500E-03	1.0920E-03
	IGD_std	1.18E-02	1.27E-02	9.38E-02	2.93E-04	4.52E-04	7.10E-05
UF2	IGD_mean	2.4654E-02	2.5129E-02	5.1777E-02	6.5679E-03	6.2990E-03	3.0150E-03
	IGD_std	5.29E-03	6.81E-03	3.89E-02	1.86E-03	2.26E-03	1.57E-03
UF3	IGD_mean	1.0700E-01	2.0788E-01	2.8125E-01	7.7375E-03	2.0731E-02	1.9640E-03
	IGD_std	4.88E-02	5.55E-02	3.16E-02	7.24E-03	1.48E-02	8.45E-04
UF4	IGD_mean	4.0342E-02	4.5096E-02	5.6310E-02	6.5966E-02	5.7473E-02	5.5996E-02
	IGD_std	3.59E-04	3.51E-03	4.42E-03	5.14E-03	4.16E-03	2.46E-03
UF5	IGD_mean	2.4529E-01	2.6515E-01	5.1287E-01	4.8404E-01	2.4221E-01	2.7968E-01
	IGD_std	4.76E-02	8.38E-02	1.12E-01	1.78E-01	7.15E-02	8.05E-02
UF6	IGD_mean	1.1619E-01	1.5182E-01	4.2483E-01	8.0708E-02	6.7855E-02	5.8669E-02
	IGD_std	1.36E-02	7.29E-02	1.32E-01	1.39E-01	1.72E-02	1.66E-02
UF7	IGD_mean	3.9008E-02	4.8449E-02	5.3477E-01	2.4031E-01	3.0410E-03	1.2460E-03
	IGD_std	1.10E-02	4.11E-02	1.24E-01	2.97E-01	7.21E-04	1.19E-04
UF8	IGD_mean	2.4207E-01	4.3922E-01	1.1450E-01	6.3398E-02	5.3024E-02	4.1114E-02
	IGD_std	1.31E-02	1.82E-02	4.99E-02	1.07E-02	1.62E-02	7.51E-03
UF9	IGD_mean	1.7224E-01	2.5835E-01	2.4637E-01	5.4433E-02	8.1060E-02	8.4664E-02
	IGD_std	7.42E-01	9.44E-02	1.35E-02	4.33E-02	5.56E-02	5.87E-02
UF10	IGD_mean	2.9920E-01	3.3266E-01	5.2249E-01	5.4759E-01	3.7634E-01	4.0986E-01
	IGD_std	3.95E-02	2.86E-02	3.51E-02	7.97E-02	5.82E-02	3.97E-02
Sum		1/10	0/10	0/10	1/10	1/10	7/10

Table 2 IGD and HV value of MOEA/D-ANA and MOEA/D-DRA

IGD	*Metrics*	*MOEA/D-DRA*	*MOEA/D-ANA*	*HV*	*Metrics*	*MOEA/D-DRA*	*MOEA/D-ANA*
UF1	IGD_mean	3.4500E-03	1.0920E-03	UF1	HV_mean	3.158	3.6612
	IGD_std	4.52E-04	7.10E-05		HV_std	6.75E-03	2.45E-03
UF2	IGD_mean	6.2990E-03	3.0150E-03	UF2	HV_mean	3.3216	3.6230
	IGD_std	2.26E-03	1.57E-03		HV_std	1.92E-02	4.87E-02
UF3	IGD_mean	2.0731E-02	1.9640E-03	UF3	HV_mean	3.4554	3.6615
	IGD_std	1.48E-02	8.45E-04		HV_std	1.86E-02	4.77E-03
UF4	IGD_mean	5.7473E-02	5.5996E-02	UF4	HV_mean	2.9887	3.1734
	IGD_std	4.16E-03	2.46E-03		HV_std	1.65E-02	1.45E-02
UF5	IGD_mean	2.4221E-01	2.7968E-01	UF5	HV_mean	2.4013	2.0191
	IGD_std	7.15E-02	8.05E-02		HV_std	1.48E-01	1.64E-01
UF6	IGD_mean	6.7855E-02	5.8669E-02	UF6	HV_mean	3.1051	3.2273
	IGD_std	1.72E-02	1.66E-02		HV_std	1.59E-02	1.51E-02
UF7	IGD_mean	3.0410E-03	1.2460E-03	UF7	HV_mean	2.7454	3.2069
	IGD_std	7.21E-04	1.19E-04		HV_std	5.89E-01	5.37E-01
UF8	IGD_mean	5.3024E-02	4.1114E-02	UF8	HV_mean	3.6756	6.0026
	IGD_std	1.62E-02	7.51E-03		HV_std	3.87E-01	1.53E-02
UF9	IGD_mean	8.1060E-02	8.4664E-02	UF9	HV_mean	4.59954	4.0338
	IGD_std	5.56E-02	5.87E-02		HV_std	5.48E-01	5.86E-01
UF10	IGD_mean	3.7634E-01	4.0986E-01	UF10	HV_mean	2.3354	2.0240
	IGD_std	5.82E-02	3.97E-02		HV_std	4.55E-01	6.63E-01

Table 2 gives experimental results of MOEA/D-DRA and MOEA/D-ANA in terms of IGD and HV. For the reason that the MOEA/D-DRA and MOEA/D-ANA have the same framework expect for adaptive neighbourhood adjustment strategy we proposed in MOEA/D-ANA. As we can see, MOEA/D-ANA is better than MOEA/D-DRA from the results. HV value mainly reflects the diversity of the evolutionary algorithms. It is worth noting that the better HV is, the better result of IGD value. The results of MOEA/D-ANA illustrate convergence and diversity in UF1-UF10. It proves the effectiveness of our algorithms.

5 Conclusions

In this paper, we proposed the MOEA/D-ANA algorithm, which aims to improve the performance of MOEA/D-DRA. The major modification of the proposed MOEA/D-ANA is the adaptive neighbourhood size adjustment for MOEA/D-DRA. To maintain the diversity of the population, we introduce the density of solutions for each sub-problem to insure uniform distribution of the obtained solutions on the Pareto front. The diversity of population also helps to improve the accuracy of algorithm. The simulation results prove that the strategy of adaptive neighbourhood size adjustment is very useful.

In the future, we will do further research with our approach, some methods of machine learning are used in MOEA/D to enhanced the accuracy of operator selection, like CPS strategy (Zhang et al., 2018a).

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant No. 61806138, Natural Science Foundation of Shanxi Province under Grant No. 201801D121127 and Taiyuan University of Science and Technology Scientific Research Initial Funding under Grant No. 20182002.

References

Abdel-Baset, M. et al. (2018) 'An improved cuckoo search algorithm for integer programming problems', *International Journal of Computing Science and Mathematics*, Vol. 9, No. 1, pp.66–81.

Amiri, E. and Dehkordi, N. (2018) 'Dynamic data clustering by combining improved discrete artificial bee colony algorithm with fuzzy logic', *International Journal of Bio-Inspired Computation*, Vol. 12, No. 3, pp.164–172.

Arloff, W. et al. (2018) 'A parameter estimation method for stiff ordinary differential equations using particle swarm optimisation', *International Journal of Computing Science and Mathematics*, Vol. 9, No. 5, pp.419–432.

Bader, J. and Zitzler, E. (2014) 'HypE: an algorithm for fast hypervolume-based many-objective optimization', *Evolutionary Computation*, Vol. 19, No. 1, pp.45–76.

Bougherara, M. et al. (2018) 'IP assignment for efficient NoC-based system design using multi-objective particle swarm optimisation', *International Journal of Bio-Inspired Computation*, Vol. 12, No. 4, pp.203–213.

Cai, X. et al. (2016) 'Improved bat algorithm with optimal forage strategy and random disturbance strategy', *International Journal of Bio-Inspired Computation*, Vol. 8, No. 4, pp.205–214.

Cai, X. et al. (2018) 'Bat algorithm with triangle-flipping strategy for numerical optimization', *International Journal of Machine Learning and Cybernetics*, Vol. 9, No. 2, pp.199–215.

Cui, Z. et al. (2017a) 'A novel oriented cuckoo search algorithm to improve DV-Hop performance for cyber–physical systems', *Journal of Parallel and Distributed Computing*, Vol. 103, pp.42–52.

Cui, Z. et al. (2017b) 'Optimal LEACH protocol with modified bat algorithm for big data sensing systems in internet of things', *Journal of Parallel and Distributed Computing*, DOI: 10.1016/j.jpdc.2017.12.014.

Cui, Z. et al. (2018a) 'Bat algorithm with principal component analysis', *International Journal of Machine Learning and Cybernetics*, DOI: 10.1007/s13042-018-0888-4.

Cui, Z. et al. (2018b) 'Detection of malicious code variants based on deep learning', *IEEE Transactions on Industrial Informatics*, Vol. 14, No. 7, pp.3187–3196.

Deb, K. et al. (2002) 'A fast and elitist multi-objective genetic algorithm: NSGA-II', *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, pp.182–197.

Ding, D. and Wang, G. (2013) 'MOEA/D-GO for fragmented antenna design', *Progress in Electromagnetics Research M*, Vol. 33, pp.1–15.

Dong, X. et al. (2017) 'Dynamic differential evolution with oppositional orthogonal crossover for large scale optimisation problems', *International Journal of Computing Science and Mathematics*, Vol. 8, No. 5, pp.414–424.

Emmerich, M. et al. (2005) 'An EMO algorithm using the hypervolume measure as selection criterion', *International Conference on Evolutionary Multi-Criterion Optimization*, pp.62–76.

Guo, X. et al. (2016) 'MOEA/D with adaptive weight vector design', *Evolutionary Computation*, Vol. 22, No. 2, pp.231–264.

Konstantinidis, A. et al. (2009) 'A sub-problem-dependent heuristic in MOEA/D for the deployment and power assignment problem in wireless sensor networks', *Proceedings of the Eleventh conference on Congress on Evolutionary Computation*, Trondheim, Norway, pp.2740–2747.

Laumanns, M. et al. (2014) 'Combining convergence and diversity in evolutionary multi-objective optimization', *Evolutionary Computation*, Vol. 10, No. 3, pp.263–282.

Li, H. and Zhang, Q. (2009) 'Multi-objective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II', *IEEE Transactions on Evolutionary Computation*, Vol. 13, No. 2, pp.284–302.

Li, K. et al. (2014) 'Stable matching-based selection in evolutionary multi-objective optimization', *IEEE Transactions on Evolutionary Computation*, Vol. 18, No. 6, pp.909–923.

Li, K. et al. (2015) 'Interrelationship-based selection for decomposition multi-objective optimization', *IEEE Transactions on Cybernetics*, Vol. 45, No. 10, pp.2076–2088.

Liu, H.L. et al. (2014) 'Decomposition of a multi-objective optimization problem into a number of simple multi-objective sub-problems', *IEEE Transactions on Evolutionary Computation*, Vol. 18, No. 3, pp.450–455.

Lv, L. et al. (2018) 'Object tracking with improved firefly algorithm', *International Journal of Computing Science and Mathematics*, Vol. 9, No. 3, pp.219–231.

Ma, C.X. (2018) 'Network optimisation design of Hazmat based on multi-objective genetic algorithm under the uncertain environment', *International Journal of Bio-Inspired Computation*, Vol. 12, No. 4, pp.236–244.

Narasimhan, M. et al. (2018) 'EGA-FMC: enhanced genetic algorithm-based fuzzy k-modes clustering for categorical data', *International Journal of Bio-Inspired Computation*, Vol. 11, No. 4, pp.219–228.

Niu, Y. et al. (2018) 'Adaptive two-SVM multi-objective cuckoo search algorithm for software defect prediction', *International Journal of Computing Science and Mathematics*, Vol. 9, No. 6, pp.547–554.

Qi, Y. et al. (2014) 'MOEA/D with adaptive weight adjustment', *Evolutionary Computation*, Vol. 22, No. 2, pp.231–264.

Reddy, S.S. and Panigrahi, B.K. (2017) 'Optimal power flow using clustered adaptive teaching learning-based optimisation', *International Journal of Bio-Inspired Computation*, Vol. 9, No. 4, pp.226–234.

Shukla, P.K. (2007) 'On the normal boundary intersection method for generation of efficient front', *Proceedings of the 7th international conference on Computational Science, Part I: ICCS 2007*, Beijing, China, pp.310–317.

Tan, Y.Y. et al. (2012). 'A modification to MOEA/D-DE for multi-objective optimization problems with complicated Pareto sets', *Information Sciences*, Vol. 213, No. 23, pp.14–38.

Wagner, T. et al. (2007) *Pareto-, Aggregation- and Indicator-Based Methods in Many-Objective Optimization*, Springer Berlin Heidelberg.

Wang, H. et al. (2018) 'Firefly algorithm with neighbourhood attraction', *Information Sciences*, DOI: 382/383:374-387.

Wang, Q. et al. (2014) 'A multi-objective optimization method based on MOEA/D and fuzzy clustering for change detection in SAR images', *2014 IEEE Congress on Evolutionary Computation (CEC)*, Beijing, China, pp.3024–3029.

Wang, R. et al. (2017) 'On the effect of reference point in MOEA/D for multi-objective optimization', *Applied Soft Computing*, Vol. 58, pp.25–34.

Wang, W. and Wang, H. (2018) 'A new differential evolution based on Gaussian sampling for forecasting urban water resources demand', *International Journal of Computing Science and Mathematics*, Vol. 9, No. 2, pp.155–162.

Yu, G. and Feng, Y. (2018) 'Improving firefly algorithm using hybrid strategies', *International Journal of Computing Science and Mathematics*, Vol. 9, No. 2, pp.163–170.

Zhang, J. et al. (2018a) 'Preselection via classification: a case study on global optimisation', *International Journal of Bio-Inspired Computation*, Vol. 11, No. 4, pp.267–281.

Zhang, M. et al (2018b) 'Hybrid multi-objective cuckoo search with dynamical local search', *Memetic Computing*, Vol. 10, No. 2, pp.199–208.

Zhang, Q. and Li, H. (2007) 'MOEA/D: A multi-objective evolutionary algorithm based on decomposition', *IEEE Transactions on Evolutionary Computation*, Vol. 11, No. 6, pp.712–731.

Zhang, Q. et al. (2008). *Multi-objective optimization Test Instances for the CEC 2009 Special Session and Competition*, University of Essex.

Zhang, Q. et al. (2009) 'The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances', *IEEE Congress on Evolutionary Computation, CEC '09*.

Zhao, S.Z. et al. (2012) 'Decomposition-based multi-objective evolutionary algorithm with an ensemble of neighbourhood sizes', *IEEE Transactions on Evolutionary Computation*, Vol. 16, No. 3, pp.442–446.

Zitzler, E. and Thiele, L. (1999) 'Multi-objective evolutionary algorithms: a comparative case study and the strength Pareto approach', *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 4, pp.257–271.

Zitzler, E. et al. (2000) 'Comparison of multi-objective evolutionary algorithms: empirical results', *Evolutionary Computation*, Vol. 8, No. 2, p.173.

Zitzler, E. et al. (2001) 'SPEA2: Improving the strength Pareto evolutionary algorithm', *Parallel Problem Solving from Nature – PPSN VIII. PPSN 2004, Lecture Notes in Computer Science*, Vol. 3242, pp.742–751.

Zitzler, E. et al. (2003) 'Performance assessment of multi-objective optimizers: an analysis and review', *IEEE Transactions on Evolutionary Computation*, Vol. 7, No. 2, pp.117–132.