A New Multi-objective Evolutionary Optimisation Algorithm: The Two-Archive Algorithm

Kata Praditwong[1] and Xin Yao[2]
The Centre of Excellence for Research in Computational Intelligence and Applications(CERCIA),
School of Computer Science,
The University of Birmingham,
Edgbaston, Birmingham B15 2TT, UK
{kxp[1],xin[2]}@cs.bham.ac.uk

Abstract

Many Multi-Objective Evolutionary Algorithms (MOEAs) have been proposed in recent years. However, almost all MOEAs have been evaluated on problems with two to four objectives only. It is unclear how well these MOEAs will perform on problems with a large number of objectives. Our preliminary study [1] showed that performance of some MOEAs deteriorates significantly as the number of objectives increases. This paper proposes a new MOEA that performs well on problems with a large number of objectives. The new algorithm separates non-dominated solutions into two archives, and is thus called the Two-Archive algorithm. The two archives focused on convergence and diversity, respectively, in optimisation. Computational studies have been carried out to evaluate and compare our new algorithm against the best MOEA for problems with a large number of objectives. Our experimental results have shown that the Two-Archive algorithm outperforms existing MOEAs on problems with a large number of objectives.

1 Introduction

Evolutionary algorithms (EA) used as a powerful tool to search solutions in complex problems during the recent past. The history of multi-objective evolutionary algorithms (MOEA) began in the mid 1980's. In [2], the authors have classified MOEAs into three categories; aggregating functions, population-based approaches, and Pareto-based approaches. A hybrid method uses a set of solutions from evolutionary computation to build a model, which presents Parato front such as ParEGO [3]. In recent years the Pareto-based approaches have become a popular design and several techniques have been proposed. One popular technique is, elitism, which uses an external storage, called an archive, to keep useful solutions. Other component that develops with an archive is a removal strategy. As an archive has limited size, removal strategy is required to trim exceeding members of archive. This operator is important because it might delete some members and keep the Pareto front in the same time. This is challenging for researchers.

Generally, multi-objective problems involve two sets of variables. The first is a set of decision variables as input of a particular problem and the other is a set of objective functions or objective values as output of a problem. Thus, the solution's quality will be measured in a space of objective functions. One factor that made the multi-objective optimisation too difficult is the number of objectives to optimise. Thus, Deb [4] suggests that the number of objectives makes an impact on the difficulty of the optimisation problems. Obviously, the dimensions of the solution space vary according to the number of objectives. For example, the shape of a solution space in two-dimensional problems is plain, while the shape of the space on three-dimensional problems is a three-dimensional surface.

Many researchers have invented several MOEAs, problem generators, and performance measurement. A small amount of publications are available that relate to performance on many objectives. Almost all simulated results still focus on two or three objectives and the behaviour of MOEAs in solving high-dimensional problems is still scrutiny.

An important comparison result in many objectives from [1] is that the scalability of objectives in each algorithm is different. The Pareto Envelope-based Selection Algorithm (PESA) [5] has powerful scalability while Non-dominated Sorting Genetic Algorithm II (NSGA-II) [6] has a lack of scalability. Generally, NSGA-II has performed well in two or three objectives. The behaviour of NSGA-II in many objectives problem is interesting. This fact helps researchers design new MOEA that can efficiently solve problems in many objectives.

This paper proposes and implements a new concept of archiving. PESA and *Two-Archive* algorithm are compared in four scalable objective problems. The experimental results are measured in convergence and diversity metrics, and they are analysed by statistical method.

The remainder of the paper is organised as follows. Section 2 will describe the concepts of non-domination with domination and the *Two-Archive* algorithm with the pseudocode. Section 3 will describe a set of scalable testing problems. Section 4 will present a set of performance measuring. The experimental setting and results will be explained in section 5. The conclusions should be shown in section 6.

2 The Two-Archive Algorithm

2.1 Non-dominated Solution with Domination

The new idea for improving convergence of archive is based on two factors. Firstly, the archive collects non-dominated solutions from a population. Next, the truncation will be applied if an archive overflows. The truncation can be applied in two ways, firstly, during collecting non-dominated solutions, as in PESA, and, secondly, after finishing the collection process, as in NSGA-II or SPEA2(Strength Pareto Evolutionary Algorithm 2) [7]. This operator can remove any member in an archive. On the other hand, all algorithms do not distinguish non-dominated solution. In the new concept, non-dominated solutions are categorised into two types according to comparison with members of the archive. The first type is a non-dominated solution with domination which dominates some existing members, and the other is ordinarily a non-dominated member without domination. Generally, the first type can improve convergence to Pareto front. Non-dominated solutions with domination should be kept in archive.

2.2 The Algorithm

The *Two-Archive* algorithm borrows the replacement of dominated solution by the new dominated solution from PESA [5]. The truncation was done at the end of collecting non-dominated solutions from NSGA-II [6] and SPEA2 [7]. The details of the proposed algorithm are shown in Algorithms 1 and 2.

The framework of the *Two-Archive* algorithm is shown by Algorithm 1. The proposed archiving method collects the new candidate solutions from a population, one by one. Firstly, a new candidate can be a new member of an archive if it is a non-dominated member in the population, and if no member in both archives can dominate it (as lines 2 and 3 in Algorithm 2). Secondly, the new member should check the domination relationship with other existing members. If the new member can dominate some other members, it should enter the convergence archive (CA) and the dominated members should be deleted. Otherwise, it should enter the diversity archive (DA) and no existing member will be deleted (line 2 to line 13 in Algorithm 2).

When the total size of both archives overflows, the removal strategy should be applied. All members in DA may calculate its distances to all members in CA, and keep the shortest distance among distances of them. The member with the shortest distance in DA should be deleted until the total size equals the threshold.

The total size of archives during the collecting process is not fixed. The size can increase over the threshold. However, the size should be reduced to the capacity of archives after the truncation process. The size of the CA never overflows because the new member entered the CA when at least one existing member in CA should be removed. In other words, the size of the CA is restricted by the number of members in both archives before collecting new members. Thus, only DA is an unlimited growing archive.

2.2.1 The Main Loop of The Algorithm

This algorithm starts with a random population and empty archives. The decision variables of each individual are assigned using the random number generator with a uniform distribution in an available decision space.

Each initial individual's objective variables are calculated using objective functions and decision variables. This algorithm uses original objective functions as fitness.

For each generation, the non-dominated members of population are kept in the archives and dominated members in the archives are deleted. The algorithm uses two archives, the convergence archive and the diversity. The total capacity is fixed but the size of each archive vary.

The mating population is built up from choosing individuals from both archives. The process to select an individual is following. Firstly, an archive is chosen with a probability. The probability is a pre-defined parameter that is a ratio to choose members from the convergence archive to the diversity archive. Secondly, a member in the chosen archive should be selected uniformly at random. Finally, the chosen parent goes to the mating population.

2.2.2 Collecting Non-dominated Solutions

The non-dominated solution collection is composed of two parts; the main part is obtaining non-dominated members from the population and the optional part is removing the further members in the diversity archive when the total size of archives is more than the capacity threshold.

Algorithm 1 The Two-Archive Algorithm

```
1: Initialise the population
2: Initialise archives to the empty set
3: Evaluate initial population
4: Set t=0
5: repeat
6:     Collect non-dominated individuals to archives
7:     Select parents from archives
8:     Apply genetic operators to generate a new population
9:     Evaluate the new population
10:    t = t + 1
11: until t == MAX_GENERATION
```

Collecting non-dominated solutions begins with fetching an individual from a population one by one. It is compared with the remainder of the population. If it is a non-dominated solution, it goes to the next step. Otherwise, it is discarded because of being dominated. The non-dominated solution from the population is compared with all members in the current archives. If it is dominated by a member of the archives, it is discarded, otherwise it is a new member of the archives. During this stage, any duplicated member is deleted. The remainder of the archives are then compared with the new member and two cases are possible. In the first case, the new member can dominate a member of the archives. The dominated member is removed and the new member is received by the convergence archive. A flag is set to the value of the convergence archive. In this case, the size of the convergence archive is possibly increased but the total size of archive is not increased because the new member enters the convergence archive by deleting at least one dominated member. In the second case, the new member cannot dominate any member and is not dominated by any archive member. The new member becomes a member of the diversity archive, and the size of the diversity archive is increased. The total size of both archives is increased because the diversity archive receives the new member without deleting any member. The above process performs until the last individual in population. The new members are separated according to their flag values.

If the total size of the archives overflows, the removal operation should be performed. The removal operator deletes only members in the diversity archive. This operator has no impact on the convergence archive. All members in the diversity archive calculate the shortest distance from themselves to the nearest member in the convergence archive. In other words, each member of the diversity archive calculates its Euclidean distance to all members of the convergence archive. The shortest one is then chosen. The member with the shortest distance among the diversity members is deleted until the total size equals the capacity.

Algorithm 2 Collect Non-Dominated Individuals To Archives

```
1: for $i$ = 1 to $popsize$ do
2:     if individual($i$) is non-dominated solution then
3:         if no member in both archives can dominates an individual($i$) then
4:             Set individual($i$).Dflag = 0
5:             if individual($i$) dominates any member in both archives then
6:                 Set individual($i$).Dflag = 1
7:                 Delete dominated member
8:             end if
9:             if member($i$).Dflag == 1 then
10:                Add individual($i$) to Convergence Archive(CA)
11:            else
12:                Add individual($i$) to Diversity Archive(DA)
13:            end if
14:        end if
15:    end if
16: end for
    {* Remove Strategy *}
17: if sizeof(CA)+sizeof(DA) $>$ limit then
18:     for $i$=1 to sizeof(DA) do
19:         DA($i$).length = maxreal
20:         for $j$=1 to sizeof(CA) do
21:             if DA($i$).length $>$ Dist( CA($j$), DA($i$) ) then
22:                 DA($i$).length = Dist( CA($j$), DA($i$) )
23:             end if
24:         end for
25:     end for
26:     repeat
27:         Delete the member of DA with the shortest length
28:     until sizeof(DA)+sizeof(CA) == limit
29: end if
```

Table 1. Population and Their Objective Values

Solution	f_1	f_2
1	0.45	0.78
2	0.51	0.75
3	0.53	0.62
4	0.72	0.49
A	0.47	0.68
B	0.78	0.44

2.3 An Example

This problem is a bi-objective minimisation problem as shown in Table 1.

For the *Two-Archive* algorithm using the example, the convergence archive consists of two members, solutions 1 and 2, and members in the diversity archive are solutions 3 and 4. For the first possible sequence, solution A will enter to archives before solution B. All members compare with candidate A and no existing members can dominate the new one. Solution A enters the convergence archive because it can dominate member 2. Thus, current members of the convergence archive are solutions 1 and A, and the diversity archive remains the same as before. Solution B will enter the diversity archive because it cannot dominate any current members. The temporal total size of the two archives can be more than the capacity during the collecting process. When archives finish the collecting process, the exceeding members in the diversity archive will be deleted. In this case, solution B is the last candidate. The diversity archive has three solutions, 3, 4, and B.

The removal strategy is based on the shortest Euclidean distance in the objective space from members in the diversity archive to members in the convergence archive. Firstly, calculating the distance from itself to all members of the convergence archive, the distance from solution 3 to solution 1 is 0.17 and the distance from solution 3 to solution A is 0.08. Secondly, choosing the shortest distance, of 0.08, the process of finding the shortest distance should be applied to the members (solutions 4 and B) in the diversity archive. The shortest distance of solution 4 is 0.31, and 0.39 for solution B.

In this case, solution 3 is deleted because it has the shortest distance. After archive-update, solutions 1 and A are in the convergence archive and the diversity archive has solutions 4 and B.

2.4 Difference Between One and Two Archives

Implementation of two archives is based on the inequality of non-dominated solutions in an archive. The comparison of the new solution and a set of members in an archive can be separated into three cases. Firstly, no member can dominate the new solution and it can dominate some members in archive. Secondly, the members and the new solution cannot dominate each other. Finally, the new solution is dominated by some members. The new solution in the last case is discarded because a property of archive is domination free. The new archive that includes the new solution in the first case is better than the previous archive because the new one is better than the old member in terms of domination relationship. In the second case, the archives do not interfere with each other before and after collecting.

After collecting the new solutions to an archive, MOEA with one archive manages all members in the same way. If the archive overflows, all members have a chance to remove. The *Two-archive* algorithm separates solutions into two archives and manages them in different ways.

The member in the convergence archive is removed because it can only be dominated by the new solution. The member of diversity archive is deleted in two cases, when the member is dominated by the new solution, or when the archives overflow. When the archives overflow, the removal strategy is only applied to the diversity archive. This is the reason why this algorithm uses two archives to store the solutions.

3 Scalable Testing Problems

A set of four scalable testing problems [8] are used in this experimental simulation. These problems are designed to achieve many features. They are easy to construct. The number of decision variables and the number of objective variables can be scaled to any number. The Pareto fronts of these problems are exactly known in terms of shape and position in objective space. These problems are invented carefully, although this experiment choose only problems DLTZ1, DTLZ2, DTLZ3 and DTLZ6, to simulate. The main reason is that the Pareto front of these problem can be easily written into mathematical expression. Some problems share the same mapping functions or shapes of the Pareto front. It is useful to remove some redundant testing problems. The DTLZ4 and the DTLZ5 use the same meta-variable mapping function with DTLZ2. The Pareto front with curve uses DTLZ6 instead of DTLZ5 because DTLZ6 has a difference mapping function from the remainder problems.

The number of objectives vary in a range from 2 to 8 objectives. The global Pareto fronts have many shapes: linear hyper-plane, unit spherical surface, and curved.

Table 2. Convergence Metric (Minimisation). The value of a two-tailed t-test with 58 degrees of freedom: T-Test (PESA-TwoArch)

	Objs	PESA	TwoArch	T-Test
DTLZ1	2	2.86948 ± 0.00591	2.48684 ± 4.29603	1.01046
	3	0.04419 ± 0.12320	0.53283 ± 2.37626	-1.69289
	4	0.02317 ± 0.09059	0.53937 ± 0.79182	**-3.00987**
	6	0.00117 ± 0.00089	0.15170 ± 0.31683	-1.45862
	8	0.00407 ± 0.00015	0.40247 ± 0.73347	**-2.54713**
DTLZ2	2	0.00008 ± 0.00019	0.00002 ± 0.00001	0.02377
	3	0.00035 ± 0.00013	0.00027 ± 0.00008	0.02939
	4	0.00170 ± 0.00039	0.00164 ± 0.00034	0.01291
	6	0.00301 ± 0.00040	0.00294 ± 0.00038	0.00906
	8	0.00689 ± 0.00109	0.00904 ± 0.00115	-0.17718
DTLZ3	2	22.52023± 22.9048	35.26955 ± 27.67275	**-9.81903**
	3	1.80296 ± 5.78546	4.23237 ± 9.39880	**-3.41480**
	4	1.16736 ± 3.50522	0.53312 ± 1.25334	1.59248
	6	0.15035 ± 0.12692	0.24030 ± 0.61444	-0.49384
	8	7.23062 ± 2.25611	19.84626 ± 16.61913	**-14.28823**
DTLZ6	2	0.79397 ± 0.32237	0.20647 ± 0.04337	**5.32092**
	3	0.20528 ± 0.21199	0.17652 ± 0.05096	0.30716
	4	3.60430 ± 0.38084	2.56216 ± 0.26565	**7.09909**
	6	5.30454 ± 0.31227	3.06482 ± 0.16873	**11.66720**
	8	6.32247 ± 0.10668	4.16521 ± 0.17995	**16.71019**

4 Metrics

The convergence metric has been proposed by Deb and Jain [9]. This metric computes averaging the smallest normalised Euclidean distance of all points in obtained Pareto front to reference set.

The convergence metric [9] is calculated by averaging the smallest Euclidean distance, d_i, from point i to the global Pareto front as

$$C = \frac{\sum_{i=1}^{n} d_i}{n} \tag{1}$$

where n presents a number of points in the obtained set.

In these testing problems, the Pareto fronts are used as the reference sets.

The concept of diversity metrics [9] is calculating distribution of projection of obtained solutions on an objective axis. The objective axis is divided into small areas according to a number of solutions. The diversity measurement is successful if all small areas have one or more representative points. The number of areas with representative point indicates a quality of diversity metric.

5 Experiments and Results

5.1 Experiment Setting

The experimental setting was based on Khare *et al.* [1]. The population size varied according to a number of objectives. The archive size is equal to the population size. In two, three, and four objectives of DTLZ1 and DTLZ2, there were 300 generations and in that of DTLZ3 and DTLZ6, 500 generations. Furthermore, the number of generations in six and eight objectives was doubled. All experiments of *Two-Archive* algorithm were repeated independently 30 times.

5.2 Results

Convergence Metric: Table 2 summaries the convergence values of the obtained solution set. In DTLZ1, PESA performed slightly better than *Two-Archive* algorithm. However, only two sets of experiments showed statistically significant differences. For the other three experiments, the convergence metrics of both archives are comparable. In DTLZ2, they performed as the same behaviour. No statistically significant differences between them were detected in this problem. In DTLZ3, PESA had better convergence values than *Two-Archive* algorithm. In three experiments, PESA outperformed *Two-Archive* algorithm and significant differences were also detected. In DTLZ6, *Two-Archive* algorithm outperformed PESA according to the convergence metric. In addition, almost all statistically significant differences (4 of 5 experiments) were obviously detected.
Diversity Metric: Diversity metric is shown in table 3. The average values of *Two-Archive* algorithm were somewhat better than that of PESA, however, a few experiments had statistically significant differences. The *Two-Archive* algorithm had better values than PESA in the diversity metric 17 of 20 experiments and only two experiments had statistically significant differences. In only three experiments, PESA performed better than *Two-Archive*, however, no statistically significant differences were detected.

This informs the analysis of performance according to the characteristics of problems. In the most simple problem, DTLZ2, both algorithms were comparable. It seems highly probable that the *Two-Archive* algorithm was prevented from the Pareto front on multi-modal, non-linear mapping function used in DTLZ1 and DTLZ3. However, the *Two-Archive* algorithm can produce a set of solutions in DTLZ6 near to the global Pareto front which its front is a curve.

6 Conclusions

In this paper, the concept of non-dominated solutions with domination was presented. We illustrated the *Two-Archive* algorithm, and compared its performance according to a set of convergence and diversity metrics on a set of scalable testing problems invented by Deb *et al.*

Table 3. Diversity Metric (Maximisation). The value of a two-tailed t-test with 58 degrees of freedom: T-Test (PESA-TwoArch)

	Objs	PESA	TwoArch	T-Test
DTLZ1	2	0.25093 ± 0.14059	0.40720 ± 0.19185	-1.48446
	3	0.42116 ± 0.07563	0.52340 ± 0.10649	-1.31218
	4	0.37605 ± 0.07125	0.42902 ± 0.06855	-0.77596
	6	0.33643 ± 0.04046	0.33463 ± 0.04299	0.02438
	8	0.25245 ± 0.00764	0.25037 ± 0.03623	0.04693
DTLZ2	2	0.57396 ± 0.09135	0.65979 ± 0.08791	-1.11032
	3	0.57163 ± 0.04344	0.63981 ± 0.03482	-1.33491
	4	0.52708 ± 0.03692	0.58181 ± 0.02321	-1.22245
	6	0.47099 ± 0.02660	0.51825 ± 0.01830	-0.82652
	8	0.43230 ± 0.04908	0.48221 ± 0.01037	-0.68861
DTLZ3	2	0.14023 ± 0.14497	0.16460 ± 0.16208	-0.24091
	3	0.38965 ± 0.13220	0.40272 ± 0.17358	-0.12949
	4	0.31659 ± 0.09393	0.40721 ± 0.12635	-1.05756
	6	0.18813 ± 0.06554	0.40871 ± 0.02731	**-2.55314**
	8	0.02615 ± 0.00247	0.10324 ± 0.10686	-1.24915
DTLZ6	2	0.20191 ± 0.14198	0.68258 ± 0.07386	**-5.66679**
	3	0.41962 ± 0.06423	0.51650 ± 0.04737	-1.58848
	4	0.22558 ± 0.02790	0.24695 ± 0.02241	-0.52192
	6	0.27631 ± 0.02356	0.31483 ± 0.01462	-0.72240
	8	0.27328 ± 0.00488	0.24692 ± 0.00924	0.93449

[8]. It is not clear which algorithm is better in terms of convergence. However, the *Two-Archive* algorithm seems to have outperformed PESA in DTLZ6, which is difficult to convergence because of the shape of the Pareto front. It is virtually certain that the *Two-Archive* algorithm will have better diversity than PESA. However, the proposed algorithm was investigated in a limited set of testing problems. Further work is required to evaluate the usefulness of the *Two-Archive* algorithm.

Acknowledgement
The authors are grateful to Felicity Simon for proof-reading the paper.

References

[1] V. Khare, X. Yao, and K. Deb. Performance Scaling of Multi-objective Evolutionary Algorithms. In C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele, editors, *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*, volume 2632 of *Lecture Notes in Computer Science*, pages 376–390, Faro, Portugal, April 2003. Springer.

[2] C. A. Coello Coello and G. B. Lamont. An Introduction to Multi-objective Evolutionary Algorithms and Their Applications. In C. A. Coello Coello and G. B. Lamont, editors, *Applications of Multi-Objective Evolutionary Algorithms*, chapter 1, pages 1–28. World Scientific Publishing, London, England, 2004.

[3] J. Knowles. ParEGO: A Hybrid Algorithm With On-Line Landscape Approximation for Expensive Multiobjective Optimization Problems. *IEEE Transactions On Evolutionary Computation*, 10(1):50–66, 2006.

[4] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester, UK, 2001.

[5] D. W. Corne, J. D. Knowles, and M. J. Oates. The Pareto Envelope-based Selection Algorithm for Multiobjective Optimization. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, editors, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, number 1917 in Lecture Note in Computer Science, pages 839–848, Paris, France, 2000. Springer.

[6] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast and Elitist Multi-Objective Genetic Algorithm-NSGA-II. *IEEE Transactions On Evolutionary Computation*, 6(2):182–197, 2002.

[7] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, May 2001.

[8] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable Test Problems for Evolutionary Multi-Objective Optimization. Technical Report TIK-Report No.112, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, July 2001.

[9] K. Deb and S. Jain. Running Performance Metrics For Evolutionary Multi-Objective Optimization. Technical Report KanGAL Report Number 2002004, Indian Institute of Technology, Kanpur, India, May 2002.