Dynamic Weighted Aggregation for Evolutionary Multi-Objective Optimization: Why Does It Work and How?

Yaochu Jin, Markus Olhofer and Bernhard Sendhoff
Future Technology Research
Honda R&D Europe (D) GmbH
63073 Offenbach/Main, Germany
Email: {yaochu.jin, markus.olhofer, bernhard.sendhoff}@hre-ftr.f.rd.honda.co.jp

Abstract

Evolutionary Dynamic Weighted Aggregation (EDWA) has shown to be both effective and computationally efficient [1] for multi-objective optimization (MOO). Besides, it was also found empirically and surprisingly that EDWA was able to deal with multi-objective optimization problems with a concave Pareto front, which has proved to be beyond the capability of the Conventional Weighted Aggregation (CWA) methods [2]. In this paper, a theory on why CWA fails for multi-objective problems with a concave Pareto front is provided schematically. According to this theory, it can easily be explained why EDWA has worked well for both convex and concave multi-objective problems. Simulation examples are conducted on various test functions to support our theory. It is concluded that EDWA is an effective and efficient method for solving multi-objective optimization problems.

1 Introduction

Evolutionary multi-objective optimization has been widely investigated in the recent years [3, 4]. Generally speaking, there are three main approaches to evolutionary multi-objective optimization, namely, weighted aggregation approaches, population-based non-Pareto approaches and Pareto-based approaches [5].

Conventional weighted aggregation (CWA) based approaches two main weaknesses. Firstly, aggregation based approaches can provide only one Pareto solution from one run of optimization. Secondly, it has been shown that weighted aggregation is unable to deal with multi-objective optimization problems with a concave Pareto front [2].

One effort using weighted aggregation based approach for multi-objective optimization (MOO) was reported in [6]. In that work, the weights of the different objectives are encoded in the chromosome to obtain more than one Pareto solution. Phenotypic fitness sharing is used to keep the diversity of the weight combinations and mating restrictions are required so that the algorithm can work properly.

An efficient and effective method called evolutionary dynamic weighted aggregation (EDWA) was proposed in [1]. The original idea in EDWA was straightforward, i.e. if the weights for the different objectives are changing during optimization, the optimizer will go through all points on the Pareto front. If the found non-dominated solutions are archived, the whole Pareto front can be achieved. This has been shown to be working well for both convex and concave Pareto fronts.

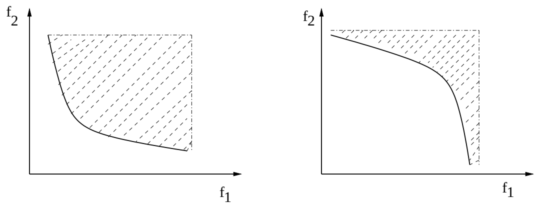
In this paper, a theory on evolutionary multiobjective optimization using weighted aggregation is suggested. Based on this theory, the reason why EDWA is able to deal with MOO is revealed. Simulations are carried out on different test functions both to support our theory and to demonstrate the effectiveness of EDWA.

2 Multi-objective Optimization with Weighted Aggregation

2.1 Definition of Multi-objective Optimization

Consider a multi-objective optimization problem with k objectives ($f_i, i = 1, 2, ..., k$) and n decision variables ($x_i, i = 1, 2, ..., n$):

$$\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), ..., f_k(\mathbf{x})), \tag{1}$$



(a) Convex Pareto front (b) Concave Pareto front

Figure 1: Convex and concave Pareto fronts.

The target of the optimization is to minimize $f_i(\mathbf{x}), i = 1, 2, ..., k$ subject to

$$g_i(\mathbf{x}) \leq 0, i = 1, 2, ..., m. \tag{2}$$

Since the k objectives may be conflicting with each other, it is usually difficult to obtain the global minimum for each objective at the same time. Therefore, the target of MOO is to achieve a set of solutions that are Pareto optimal. The related concepts of Pareto dominance, Pareto optimality, Pareto optimal set and Pareto front are defined as follows [4]:

Pareto dominance: A vector $\mathbf{u} = (u_1, ..., u_k)$ is said to dominate $\mathbf{v} = (v_1, ..., v_k)$ if and only if $u_i \leq v_i, i = 1, 2, ..., k$ and there exists at least one element with $u_i < v_i$.

Pareto optimality: A solution $\mathbf{x}$ is said to be Pareto optimal if and only if there does not exist another solution $\mathbf{x}'$ so that $\mathbf{f}(\mathbf{x})$ is dominated by $\mathbf{f}(\mathbf{x}')$. All the solutions that are Pareto optimal for a given multi-objective optimization problem are called the Pareto optimal set ($\mathcal{P}^\star$).

Pareto front: For a given multi-objective optimization problem and its Pareto optimal set $\mathcal{P}^\star$, the Pareto front ($\mathcal{PF}^\star$) is defined as:

$$\mathcal{PF}^\star = \{\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), ..., f_k(\mathbf{x}))|\mathbf{x} \in \mathcal{P}^\star\}. \tag{3}$$

There are generally convex and concave Pareto fronts. A Pareto front ($\mathcal{PF}^\star$) is said to be convex if and only if $\forall \mathbf{u}, \mathbf{v} \in \mathcal{PF}^\star, \forall \lambda \in (0,1), \exists \mathbf{w} \in \mathcal{PF}^\star : \lambda||\mathbf{u}|| + (1-\lambda)||\mathbf{v}|| \geq ||\mathbf{w}||$.

On the contrary, a Pareto front is said to be concave if and only if $\forall \mathbf{u}, \mathbf{v} \in \mathcal{PF}^\star, \forall \lambda \in (0,1), \exists \mathbf{w} \in \mathcal{PF}^\star : \lambda||\mathbf{u}|| + (1-\lambda)||\mathbf{v}|| \leq ||\mathbf{w}||$.

For example, Fig.1(a) is a convex Pareto front and Fig.1(b) is a concave Pareto front. Of course, a Pareto front can be partially convex and partially concave.

2.2 Conventional Weighted Aggregation for MOO

Conventional Weighted Aggregation (CWA) is a straightforward approach to multi-objective optimization. In this method, the different objectives are summed up to a single scalar with a prescribed weight

$$F = \sum_{i=1}^{k} w_i f_i(\mathbf{x}), \tag{4}$$

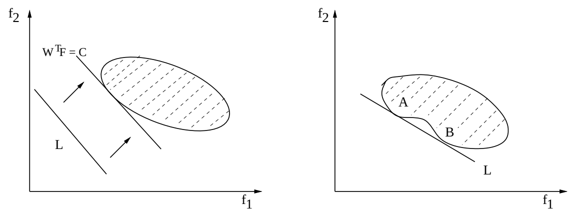
where w_i is the non-negative weight for objective $f_i(\mathbf{x}), i = 1, ..., k$. Usually, *a priori* knowledge is needed to specify the weights. During the optimization, the weights are fixed in conventional weighted aggregation method.

Using this method, only one Pareto optimal solution can be obtained with one run of the optimization algorithm. In other words, if one intends to obtain different Pareto solutions, one has to run the optimizer several times. This is of course not allowed in a lot of real world problems because it usually takes too much time to run the optimization more than once.

What is worse, efficiency is not the only problem for CWA. It was pointed out that CWA is not able to obtain the Pareto solutions that are located in the concave region of the Pareto front [2].

However, it is not as straightforward as one might imagine to explain the reason why the solutions in the concave region of the Pareto front cannot be obtained using CWA. One attempt to explain this problem is illustrated in Fig. 2, which was provided in [2]. In the figure, line L denotes solutions with the same cost and the slope of the line is determined by the weights. According to this theory, the solutions in the concave region between point A and B cannot be reached by CWA based methods. Unfortunately, this illustration is incorrect because the solutions outside the shaded area are unreachable anyway and therefore, it is impossible for the optimizer to proceed towards the Pareto front from the origin, in particular for minimization problems.

In [5], another illustration as shown in Fig. 3 is used. However, from this illustration, it is still unclear why the solutions in the concave region are not obtainable with CWA methods. Further explanations are provided as follows. In Fig. 4 (a), it can be seen that the line with equal cost will converge to a point on a convex Pareto front when the slant of the line is given, that is, when the weights are fixed. In contrast, the line will continue to move after it reaches a point (point C, which is corresponding to the given weights)



(a) Convex Pareto front. (b) Concave Pareto front.

Figure 2: Geometrical representation of weighted sum approach abstracted from [2].

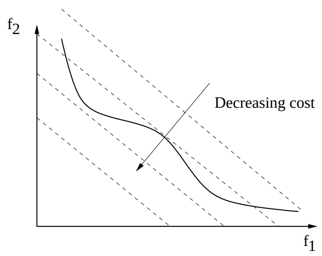
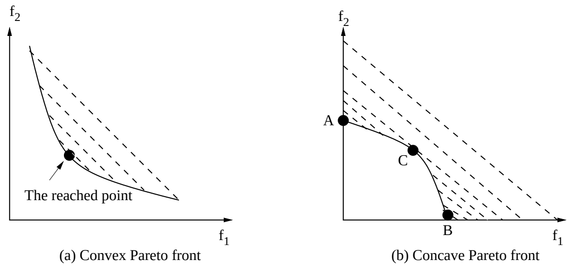


Figure 3: Line of equal cost introduced by the weighted-sum approach abstracted from [4].

in the concave region of the Pareto front, until no further minimization of the cost is possible. Finally, the obtained solution will be either A or B.

In the following, a new explanation for this problem is suggested. In our opinion, whether CWA is able to converge to a Pareto-optimal solution depends on the stability of the Pareto solution corresponding to the given weight combination. If the Pareto solution corresponding to a given weight combination is a stable minimum, then it can be obtained with CWA. To explain this further, let us have a look at the problem from another point of view. We first discuss a convex Pareto front. For a two-objective problem, if the Pareto front is presented in the conventional way, as shown in Fig. 5 (a), then point B is the stable minimum on the Pareto front. That is to say, point B will be the solution obtained by CWA given the weight combination of $(w_1, w_2) = (0, 1)$. If the weight combination is changed in optimization, it is equivalent to rotating the coordinate system together with the Pareto front. Thus, when w_1 decreases and w_2 increases, it is equal to rotate the coordinate system counter-clockwise. If $f_1^\star = f_2^\star$, then for a given weight combination of $(w_1, w_2) = (0.5, 0.5)$, the coordinate system rotates 45 degrees. In this case, C is the stable minimum of the Pareto front that will be obtained using CWA with $(w_1, w_2) = (0.5, 0.5)$, as shown in Fig. 5 (b). Obviously, for a weight combination of $(w_1, w_2) = (1, 0)$, A is the stable minimum and the coordinate system rotates 90 degrees. Therefore, different Pareto solutions will be obtained using the conventional weight aggregation with different weight combinations if the Pareto front is convex. Since the weights are always non-negative, the maximal rotation angle is 90 degree. Without considering the time consumption, the whole Pareto front can be obtained by running the optimizer as many times as possible.



(a) Convex Pareto front (b) Concave Pareto front

Figure 4: Conventional weighted aggregation for MOO. (a) Convex Pareto front, (b) Concave Pareto front.

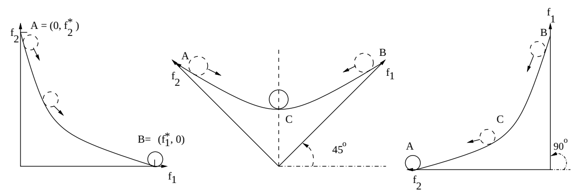


Figure 5: Convex Pareto front. All Pareto solutions are stable minimum when the coordinate system rotates. (a) 0 degree; (b) 45 degree; (c) 90 degree.

Now let us have a look at a concave Pareto front. As illustrated in Fig. 6, all solutions located in the concave region of the Pareto front are unstable when the weight combination changes. As explained above, Fig. 6 (a) corresponds to a weight combination of $(w_1, w_2) = (0, 1)$ and the solution will be point B. For all weight combinations that corresponds to a rotation angle between 0 and 45 degrees, the solution to be obtained will be B, whereas for all weight combinations that correspond to a rotation angle between 45 and 90 degrees, the solution to be obtained will be A. The weight combination that corresponds to a rotation angle of 45 degree (if $f_1^\star = f_2^\star$) is a dividing point (Point C). If the weight combination exactly corresponds to this dividing point, the result of the optimization can either be A or B, depending on the initial condition and dynamics of the optimizer. As a conclusion, only point A and B are stable minima on the Pareto front no matter how the weight combination changes.

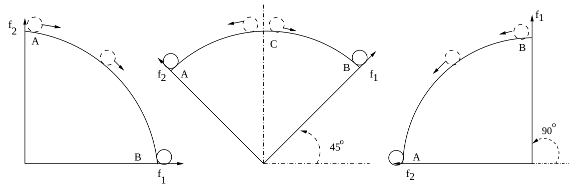


Figure 6: Concave Pareto front. All Pareto solutions are unstable minimum except the two points on both ends when the coordinate system rotates. (a) 0 degree; (b) 45 degree; (c) 90 degree.

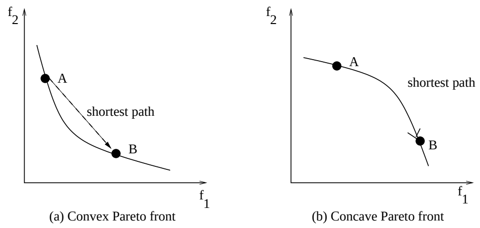


Figure 7: For a convex Pareto front (a), it is not the shortest path between two points, whereas for a concave Pareto front (b), it is the shortest.

According to the above discussions, we can draw the following conclusions:

- For a convex Pareto front, each weight combination corresponds to a stable minimum on the Pareto front.
- For a concave Pareto front, all solutions with exception to the two points on the two ends are unstable when the conventional weighted aggregation is used. Therefore, an optimizer is unable to converge to the Pareto solution corresponding to the weight combination.
- When the Pareto front is rotated slowly from 0 degree to 90 degree, the optimizer will go along the Pareto front from one stable minimum to another, *once it reaches any point of the Pareto front.* If the Pareto front is convex, the moving speed is determined by the change of weights. If the Pareto font is concave, the optimizer will stay on one stable minimum until this point becomes unstable. In this case, the optimizer will move *along the Pareto front* to the next stable minimum.

3 Evolutionary Dynamic Weighted Aggregation

As it is pointed out in the last section, if we rotate the Pareto front 90 degree, the optimizer will go from one stable optimum to another. This can be done in two ways:

- After the optimizer has converged to one stable minimum, the Pareto front is rotated 90 degrees abruptly. In the two-objective case, this corresponds to the situation where w_1 is changed from 0 to 1 and w_2 from 1 to 0. We call it Bang-bang Weighted Aggregation (BWA).
- The Pareto front is rotated gently, that is, the weights are changed gradually. In this case, the optimizer will traverse the whole Pareto front and all the solutions on the front will be obtained. This was denoted as Generation-based Periodical Variation of the Weights in [1]. In this paper, it is called Dynamic Weighted Aggregation (DWA).

In both cases, the weights are changed periodically. This may be helpful if the Pareto front is not uniform. By uniform, we mean that if the distance in the weight space is the same, then the distance on the Pareto front is also the same.

3.1 Bang-bang Weighted Aggregation

Bang-bang weighted aggregation (BWA) can be seen as a test of our theory proposed in the last section. According to our theory, it is also *possible* to obtain the whole Pareto front when we rotate it 90 degrees abruptly, no matter whether it is convex or concave. However, we expect that the optimizer may not necessarily keep moving along the Pareto front if it is convex, because the Pareto front is not the shortest feasible path from one stable point to another, refer to Fig. 7 (a). Very interestingly, if the Pareto front is concave, the optimizer should keep moving along the Pareto front because it provides the shortest feasible path from one stable point to another, as illustrated in Fig. 7 (b).

A bang-bang change of weights can be realized in the following way for a two-objective minimization problem:

$$w_1(t) = \mathrm{sign}(\sin(2\pi t/F)) \quad (5)$$
$$w_2(t) = 1.0 - w_1, \quad (6)$$

where t is the generation index and F is the frequency of the weight change. It is clear that F should be large enough to allow the optimizer to move from one stable point to another.

3.2 Dynamic Weighted Aggregation

In dynamic weighted aggregation (DWA), the weights are changed gradually. This slow change of the weights will force the optimizer to keep moving on the Pareto front if it is convex. If it is concave, the performance of DWA may not have much difference from that of the BWA. This can be realized as follows:

$$w_1(t) = |\sin(2\pi t/F)|, \tag{7}$$

$$w_2(t) = 1.0 - w_1(t), \tag{8}$$

where t is the number of generation. It is noticed that $w_1(t)$ changes from 0 to 1 periodically. The change frequency can be adjusted by F. The frequency should not be too high so that the algorithm is able to converge to a minimum. On the other hand, it seems reasonable to let the weight change from 0 to 1 twice during the whole optimization. In the simulation described in Section 5.3 and 5.4, F is set to 100 for BWA and 200 for DWA, so that for both methods, the Pareto front rotates three times in 150 generations.

3.3 An Archive of Pareto Solutions

In both BWA and DWA, the population is not able to keep all the found Pareto solutions, although it is able to traverse the Pareto front dynamically. Therefore, it is necessary to record the Pareto solutions that have been found so far. To this end, it is necessary to maintain an archive of the Pareto-optimal solutions. The pseudo-code for building the archive is listed in Algorithm 1. The similarity is measured by the Euclidean distance in the fitness space.

Algorithm 1 Pseudo-code for maintaining an archive of Pareto solutions.

```
for each individual o in offspring population do
  if (o dominates an individual in parent population
  p) and (o is not dominated by any solutions in the
  archive) and (o is not similar to any solutions in
  the archive) then
    if archive is not full then
      add o to the archive
    else if o dominates any solution a in the archive
    then
      replace a with o
    else if any solution a1 in the archive dominates an-
    other solution a2 then
      replace a2 with o
    else
      discard o
    end if
  else
    discard o
  end if
end for
for each solution in the archive do
  if solution a1 dominates a2 then
    remove a2
  end if
end for
```

4 Test Functions

To evaluate our theory and to demonstrate the effectiveness of our methods, simulations are carried out on five test functions. The first three test functions are taken from [7, 8] and the fourth test function is adapted from test functions F_2 and F_3 so that its Pareto front is partially convex and partially concave. F_5 has a discontinuous Pareto front, which is used to test how the methods behave when the Pareto front is discontinuous. Note that for all test functions, $x_i \in [0, 1]$.

- The first test function (F_1) used here is the second function in [8] and we extend it to an n-dimensional. The Pareto front of this function is uniform.

$$f_1 = \frac{1}{n}\sum_{i=1}^{n} x_i^2 \tag{9}$$

$$f_2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - 2.0)^2 \tag{10}$$

- The second test function (F_2) is the first function in [7], which has a convex but non-uniform Pareto front:

$$f_1 = x_1 \tag{11}$$

$$g(x_2, ..., x_n) = 1.0 + \frac{9}{n-1}\sum_{i=2}^{n} x_i \tag{12}$$

$$f_2 = g \times (1.0 - \sqrt{f_1/g}). \tag{13}$$

- The third test function (F_3) is the second function in [7], which has a concave Pareto front:

$$f_1 = x_1 \tag{14}$$

$$g(x_2, ..., x_n) = 1.0 + \frac{9}{n-1}\sum_{i=2}^{n} x_i \tag{15}$$

$$f_2 = g \times (1.0 - (f_1/g)^2). \tag{16}$$

- The fourth test function (F_4) adapted from F_2 and F_3, which has a Pareto front that is neither purely convex nor purely concave:

$$f_1 = x_1 \quad (17)$$

$$g(x_2, ..., x_n) = 1.0 + \frac{9}{n-1}\sum_{i=2}^{n} x_i \quad (18)$$

$$f_2 = g \times (1.0 - \sqrt[4]{f_1/g} - (f_1/g)^4). \quad (19)$$

- The fifth test function (F_5) is the third function in [7], whose Pareto front consists of a number of separated convex parts:

$$f_1 = x_1 \quad (20)$$

$$g(x_2, ..., x_n) = 1.0 + \frac{9}{n-1}\sum_{i=2}^{n} x_i \quad (21)$$

$$f_2 = g \times (1.0 - \sqrt{f_1/g} - (f_1/g) \sin(10\pi f_1)). \quad (22)$$

5 Simulation Studies

5.1 The Evolution Strategies

In the standard evolution strategy, the mutation of the objective parameters is carried out by adding an $N(0, \sigma_i^2)$ distributed random number. The step size σ_i is also encoded in the genotype and subject to mutations. A standard evolution strategy can be described as follows:

$$\mathbf{x}(t) = \mathbf{x}(t-1) + \tilde{\mathbf{z}} \quad (23)$$

$$\sigma_i(t) = \sigma_i(t-1)\exp(\tau' z)\exp(\tau z_i) \quad (24)$$

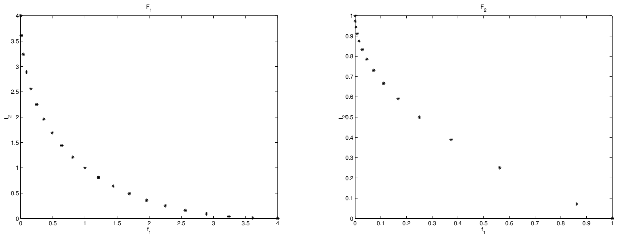
where $\mathbf{x}$ is an n-dimensional parameter vector to be optimized, $\tilde{\mathbf{z}}$ is an n-dimensional random number vector with $\tilde{\mathbf{z}} \sim N(\mathbf{0}, \boldsymbol{\sigma}(t)^2)$, z and z_i are normally distributed random numbers with $z, z_i \sim N(0,1)$. Parameters τ, τ' and σ_i are the strategy parameters, where σ_i is mutated as in equation (24) and τ, τ' are constants as follows:

$$\tau = \left(\sqrt{2\sqrt{n}}\right)^{-1}; \ \tau' = \left(\sqrt{2n}\right)^{-1} \quad (25)$$

There are several extensions to the above standard ES. In this work, the standard (μ, λ)-ES [9] is employed.

5.2 Conventional Weighted Aggregation

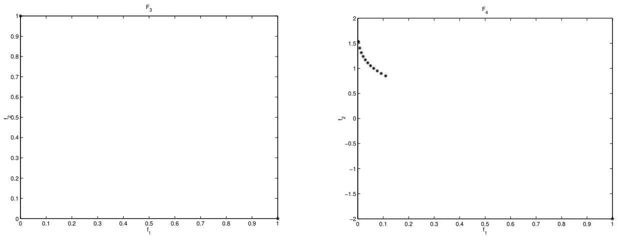
We first employ CWA to obtain the Pareto front. As we mentioned above, we have to run the optimizer more than once if we attempt to obtain more than one solution. In this work, the algorithm is run for 20 times for test functions F_1, F_2 and F_3, and 40 times for test function F_4. Since the Pareto solutions are not uniformly distributed on the Pareto front corresponding to a uniformly distributed weight combinations, smaller weight change is needed to obtain the solutions in the convex region of the Pareto front in function F_4. In all the simulations, the dimension n is set to 2.



(a) (b)

Figure 8: CWA for F_1 and F_2. The results are collected from 20 runs of the optimization.

The results on F_1 and F_2 are given in Fig. 8. Since both functions are convex, the CWA based approach is able to obtain different Pareto solutions with different weights. We see that the distribution of the solutions from F_2 is not uniform, although the distribution of the weights is uniform.



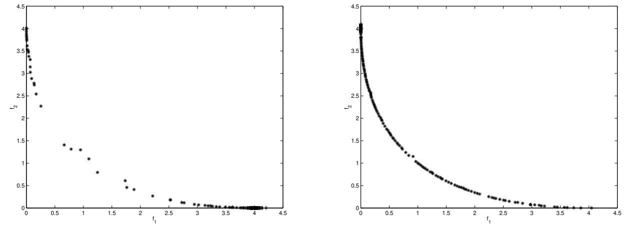
(a) (b)

Figure 9: CWA for F_3 and F_4. 20 runs are carried out for F_3 and 40 runs are carried for F_4.

Fig. 9 provides the results on F_3 and F_4. Since the Pareto front of F_3 is concave, we can only obtain two solutions, whereas for F_4, the solutions in the convex region are obtained and those in the concave region are not obtained, which is expected from our discussion in Section 3.

5.3 Bang-bang Weighted Aggregation and Dynamic Weighted Aggregation

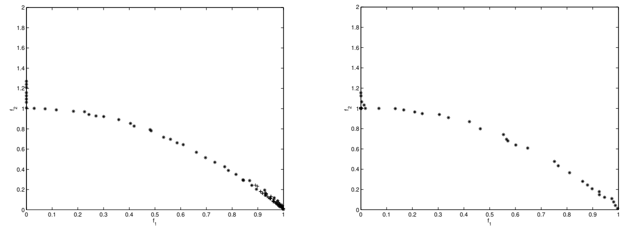
In this part, we intend to empirically support our theory on multi-objective optimization by showing that bang-bang weighted aggregation is able to obtain the Pareto set, in particular for concave Pareto fronts. At the same time, the performance of both BWA and DWA are compared for different situations. For both methods, 150 generations are run so that the weight can switch three times during optimization, as mentioned in Section 3.2.



(a) (b)

Figure 10: Results on F_1: (a) BWA and (b) DWA.

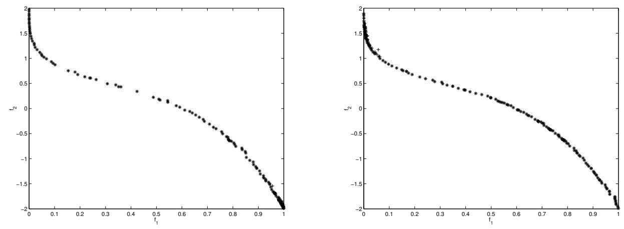
According to our theory, BWA may perform worse than DWA for convex Pareto fronts, because the Pareto front between two stable points is not the shortest feasible path. This can be seen from the results on F_1, which are shown in Fig. 10. However, when the Pareto front is concave, the performances of BWA and DWA are similar, as shown in Fig. 11. This is consistent with our theory.



(a) (b)

Figure 11: Results on F_3: (a) BWA and (b) DWA.

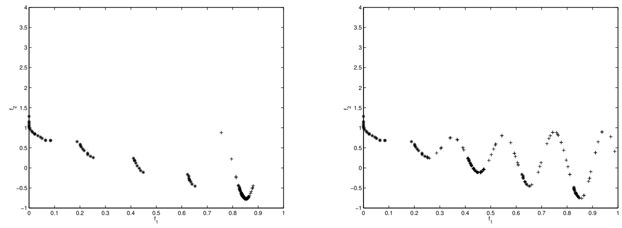
Test function F_4 has a partially convex and partially concave Pareto front. Since its convex part is relatively small, there is no essential discrepancy between the results from BWA and DWA, see Fig. 12.



(a) (b)

Figure 12: Results on F_4: (a) BWA and (b) DWA.

All the above test functions, F_1, F_3 and F_4 have a continuous Pareto front. It is desirable to investigate how our methods work for discontinuous Pareto fronts, particularly when BWA is used. In Fig. 13 (a), we see that BWA has successfully obtained the discontinuous Pareto front. Amazingly, the algorithm is able to move from one part of the Pareto front to another through a *bridge* that connects the different parts of the Pareto front, which is shown in Fig. 13 (b) caught by a snapshot during optimization.



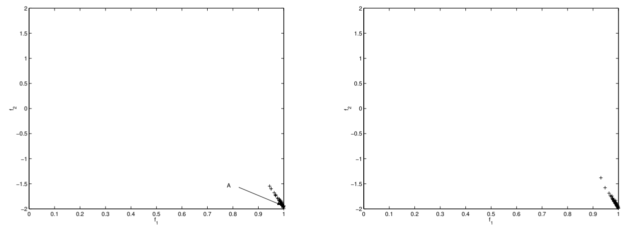
(a) (b)

Figure 13: BWA for a discontinuous Pareto front (F_5): (a) The obtained Pareto front; (b) A snapshot showing how the BWA moves between different parts of the Pareto front.

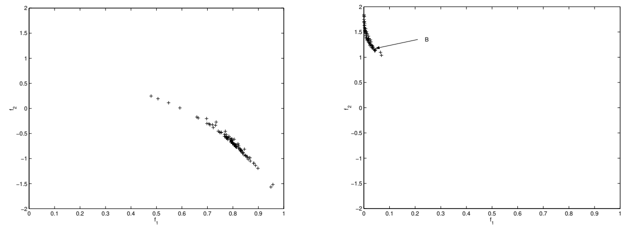
Similar results have been obtained on F_5 using DWA.

5.4 Discussions

From the simulation results, we can make the following observations:



(a) (b)



(c) (d)

Figure 14: DWA for F_4: The optimizer starts to move along a concave Pareto front only when the rotation angle reaches a dividing point. (a) $w_1 = 0.0$; (b) $w_1 = 0.9$; (c) $w_1 = 0.92$ and (d) $w_1 = 1.0$.

- For convex Pareto fronts, DWA exhibits better performance than BWA. The reason is that the optimizer will not necessarily keep moving along the Pareto front if BWA is used.

- For concave Pareto fronts, BWA and DWA show similar performance. However, BWA may be more efficient than DWA when the Pareto front is concave. This is due to the fact that when the Pareto front rotates, the optimizer stays at one stable solution until a rotation angle corresponding to the dividing point has been reached, as discussed in Section 2.2. Let us have a look at the results from applying the DWA to F_4. In generation 100, $w_1 = 0.0$ and the optimizer is around the stable point A, see Fig. 14 (a). When the evolution proceeds, the optimizer remains on point A until in generation 135, when $w_1 = 0.90$, see Fig. 14 (b). In generation 137, the optimizer has moved through the most part of the concave region (Fig. 14 (c)), where $w_1 = 0.92$. Finally, in generation 150, the optimizer is around the other stable point of the concave region, i.e., point B in Fig. 14 (d). It should be noticed that the solutions in the archive are not shown in Fig. 14.

In application, if one does not know in advance if the Pareto front is convex or concave, DWA is recommended to ensure that the optimizer will move along the Pareto front to obtain the whole set of Pareto solutions. However, if one knows that the Pareto front is concave, the BWA may need less time to achieve the whole Pareto set.

6 Conclusion

Multi-objective optimization using weighted aggregation based approaches is revisited. The problem of concave Pareto fronts in MOO is discussed and a theory why CWA based approaches are unable to obtain the solutions in the concave region of the Pareto front is proposed. An evolutionary dynamic weighted aggregation (EDWA) is proposed to obtain Pareto solutions in one run, no matter whether they are located in the convex or concave region of the Pareto front. The proposed method is shown to be not only efficient, meaning it is able to obtain Pareto solutions in one run of the optimization, but is also able to obtain the solutions located in the concave region of the Pareto front. This is very encouraging because the EDWA is computationally efficient and all existing evolutionary algorithms can be employed with minor modifications to change the weight dynamically during optimization.

However, theoretic work may be necessary to ascertain that when the weights changes and the optimizer moves from one stable minimum to another stable minimum, all the solutions in the concave region between the two minima are reached. The conclusion that local optima are concentrated in a very small region of the solution space [10] may be one support for the EDWA and vice versa, the successful of the EDWA is also a support for this conclusion.

ACKNOWLEDGMENTS

The authors would like to thank E. Körner and W. von Seelen for their support, T. Arima for his helpful comments and M. Hasenjäger and T. Okabe for stimulating discussions.

References

[1] Y. Jin, T. Okabe, and B. Sendhoff. Adapting weighted aggregation for multi-objective evolution strategies. In K. Deb, L. Thiele, and E. Zitzler, editors, *First International Conference on Evolutionary Multi-criterion Optimization*, Lecture Notes in Computer Science, pages 96–110, Zurich, Switzerland, 2001. Springer.

[2] P.J. Fleming. Computer aided control systems using a multi-objective optimization approach. In *Proc. IEE Control'85 Conference*, pages 174–179, Cambridge, U.K., 1985.

[3] C.A.C. Coello. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems*, 1(3):269–308, 1999.

[4] D. A. Van Veldhuizen and G. B. Lamont. Multiobjective evolutionary algorithms: Analyzing the state-of-art. *Evolutionary Computation*, 8(2):125–147, 2000.

[5] C. M. Fonseca and P. J. Fleming. Multiobjective optimization. In Th. Bäck, D. B. Fogel, and Z. Michalewicz, editors, *Evolutionary Computation*, volume 2, pages 25–37. Institute of Physics Publishing, Bristol, 2000.

[6] P. Hajela and C. Y. Lin. Genetic search strategies in multicriteria optimal design. *Structural Optimization*, 4:99–107, 1992.

[7] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolution algorithms: empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.

[8] J. D. Knowles and D. W. Corne. Approximating the nondominated front using the Pareto archived evolution strategies. *Evolutionary Computation*, 8(2):149–172, 2000.

[9] H.-P. Schwefel. *Evolution and Optimum Seeking*. Sixth-Generation Computer Technologies Series. John Wiley & Sons, Inc., 1994.

[10] P.C. Borges and M.P. Hansen. A basis for future successes in multiobjective combinatorial optimization. Technical Report IMM-REP-1998-8, Department of mathematical Modeling, Technical University of Denmark, 1998.