The Importance of Scalability When Comparing Dynamic Weighted Aggregation and Pareto Front Techniques

Grzegorz Drzadzewski and Mark Wineberg

Computing and Information Science, University of Guelph,
Guelph, Ontario, N1G 2W1, Canada
gdrzadze@uoguelph.ca,
wineberg@cis.uoguelph.ca

Abstract. The performance of the Dynamic Weight Aggregation system as applied to a Genetic Algorithm (DWAGA) and NSGA-II are evaluated and compared against each other. The algorithms are run on 11 two-objective test functions, and 2 three-objective test functions to observe the scalability of the two systems. It is discovered that, while the NSGA-II performs better on most of the two-objective test functions, the DWAGA can outperform the NSGA-II on the three-objective problems. We hypothesize that the DWAGA's archive helps keep the searching population size down since it does not have to both search and store the Pareto front simultaneously, thus improving both the computation time and the quality of the front.

1 Introduction

At the present moment in the field of evolutionary computation there is very little research being conducted to investigate the behaviour of newly developed work by researchers other than the original creator. This is a major deficiency in the field. In most other disciplines of science the important aspect is the repeatability of experiments and confirmation of results by other independent research teams. In this paper we are performing an un-bias study reproducing the newly developed Dynamic Weight Aggregation Evolutionary Strategy (DWAES) algorithm and comparing it to a popular Pareto front style algorithm (NSGA-II).

There are two main approaches to evolutionary multi-objective optimization: weighted aggregation approaches and Pareto-based approaches.

The weighted aggregation approaches are easier to implement and understand, as well as being the first of the Evolutionary Multi-Objective Optimization (EMOO) algorithms created. However, recently they have been deemed flawed since they only produce a single solution along the Pareto front, and in many circumstances cannot find particular solutions along the front, no matter what weightings are used. Consequently Pareto-based approach has risen in popularity and now dominates the literature. This group of algorithms work by dividing its population into dominated and non-dominated solutions [1], where a non-dominated solution is one where no other solution is better than it across every objective. These groups of algorithms have often been analysed and compared with each other.

E. Talbi et al. (Eds.): EA 2005, LNCS 3871, pp. 143 – 154, 2006.

Recently, a modification to the simplistic weighted aggregation approach was proposed: the Dynamic Weight Aggregation (DWA) system [2], [3]. This system, while based on the weighted aggregation approach, was designed to overcome the two shortcomings mentioned above[1]. Experimentation on traditional EMOO problems seemed to verify the technique. However, the DWA system was never directly compared to the Pareto based EMOO methods.

In this paper we compare the DWA method as applied to the GA against a Pareto base EMOO system, the NSGA-II, to see if the DWA produces solutions of as high quality (as close to the Pareto front and covering the front as evenly).

2 The Two Systems

2.1 Non-dominated Sorting GA

One of the most popular of the Pareto-based approaches is the NSGA-II algorithm, which is an enhancement of the original non-dominated sorting GA (NSGA) proposed by Srinivas and Deb in 1994 [4]. The NSGA algorithm first sorts the solutions by fronts: each subset of the population that is not dominated by any other member of population is separated from those that are, with this definition recursively applied as each front is removed from the population. From this sorted population, standard reproduction techniques are applied using the front levels as fitness.

The NSGA-II uses a new non-dominated sorting approach, which is more efficient than the original method [5]. The old sorting algorithm used in NSGA has a complexity of $O(mN^3)$. The NSGA-II algorithm has improved the performance of the sort so it now has a complexity of $O(mN^2)$, where m is the number of objectives and N is the population size – this improves the execution time significantly. The NSGA-II also incorporates elitism and has a parameter-less diversity preservation mechanism.

2.2 The Dynamic Weighted Aggregation Systems

The conventional weighted aggregation (CWA) approach, which is a simple weighted sum of the different objective fitness values into a single fitness value, while being the simplest approach to Evolutionary Multi-Objective Optimization (and the first utilized), has been severely criticized on account of two main weaknesses [1]: First, the conventional weighted aggregation can provide only one Pareto solution from one run of optimization. Second, it has been shown that weighted aggregation is unable to deal with multi-objective optimization problems with a concave Pareto front.

Recently a new dynamic weight aggregation algorithm was proposed with the claim that it has eliminated the two problems associated with the conventional approach [2], [3]. The idea behind the algorithm is that "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"[3]. This works for both the convex and concave

[1] Similar dynamic weighting techniques have also been used in non-evolutionary search methods such as Pareto Simulated Annealing [10], and Multi-Objective Tabu Search [11].

Pareto fronts. A theory for why the CWA algorithm does not work on concave Pareto front is provided in [2], which states that the CWA can only converge to a Pareto-optimal solution if the Pareto solution corresponding to the given weight combination is stable. Since all points on convex Pareto front are stable CWA has no trouble with it, but it is unable to reach points on the concave Pareto front. DWA algorithm on the other hand is able to go through all the points on the concave and convex Pareto front.

Using the CWA approach, a total fitness value for the chromosome is computed from the multiple fitness functions by performing a weighted sum

$$f(c) = w_1 f_1(c) + w_2 f_2(c) = w_1 f_1(c) + (1 - w_1) f_2(c) \tag{1}$$

where w_1 and w_2 are constant weights (which must sum to 1).

In the DWA, the constant weights are changed to time varying weights, $w_1(t)$ and $w_2(t)$, where t is 'time' measured in generations. The equations used in [2] for the two dynamic weights are:

$$w_1(t) = \left|\sin(2\pi t / T)\right| \tag{2}$$

and

$$w_2(t) = 1.0 - w_1(t) \tag{3}$$

where T is the period, a user defined parameter that controls how rapidly the weights cycle from 0 to 1 and back again.

In the case of a three objective problem, the weights are computed similarly, except that now there is rotation about two axes instead of just one and the weights are determined based on variables α and β.

$$\begin{aligned} w_1(\alpha) &= |\sin(2\pi\alpha)| \\ w_2(\alpha,\beta) &= (1 - w_1(\alpha))\,|\sin(2\pi\beta)| \\ w_3(\alpha,\beta) &= 1 - w_1(\alpha) - w_2(\alpha,\beta), \end{aligned} \tag{4}$$

where $0 \le \alpha, \beta \le \pi/2$.

Since the fitness function changes from generation to generation, it becomes important to store good solutions found in each generation. These good solutions are stored in the *archive*. A solution is added to the archive if it is not Pareto-dominated by any member of the archive. If a new solution Pareto-dominates members of the archive then all the dominated solutions are removed from it while the new solution is added.

3 Experimental Design

3.1 Algorithms and Parameters

To compare NSGA-II with the DWA system, it is important to isolate the various features of the two systems. This is both to assure a fair comparison, and to prevent extraneous factors from obscuring the underlying differences or similarities. Consequently, we chose to keep the underlying evolutionary algorithms the same for both systems. This means that all the parameters, with the exception of any system specific parameters, are set in common.

Table 1. Parameter Settings

Common parameter

	2 obj	**3 obj**		**2 obj**	**3 obj**
Population	100	{600,800}	Length	10	{14,16}
Generations	150	{900,1200}	Tournament Sel Pres.	0.9	.9
Prob. of cross over	0.8	0.8	Uniform xover prob.	0.4	.4
Mutation Rate	0.1	{0.071,0.0625}	Alphabet Size	100	100

DWAGA only parameters

	2 obj	**3 obj**		**2 objectives**	**3 obj**
# of 90º rotations	2	n/a	Gen	{150, 250, 600, 900, 2250}	n/a
Archive Size	100	1000			

To accomplish this uniformity for comparison we had to choose which evolutionary algorithm to base the two systems on. The NSGA, as its name implies, was designed to work on top of a Genetic Algorithm. The DWA, on the other hand, was originally written for an Evolutionary Strategy system. Since the DWA is just a modification of the fitness weights, which can be trivially used for either ES or GA, we chose to implement a Dynamic Weighted Aggregation Genetic Algorithm (DWAGA) to compare against the NSGA-II system.

3.1.1 Two Objective Problems

The performance of DWAGA was examined using 5 different period values (*T*). The values for the period length varied all the way from 200 to 7500 depending on the test function. It was discovered that DWAGA worked best when the period was set to a value that makes the number of 90º rotations equal to 2 (using equation 2).

Using 150 generations the DWAGA with a period of 600 will perform one 90º rotation; a period of 300 will result in two 90º rotations, 200 results in 3 rotations, 150 in 4 rotations, and 120 in 5 rotations.

When testing we discovered that our implementation of the DWAGA was, in general, faster than the NSGA-II. Therefore the DWAGA could perform more generations and improve the solutions that it had obtained and still finish at the same time as the NSGA-II. Consequently we ran DWAGA for a varying number of generations, making sure that the time equaled that of the NSGA-II.

The details for parameter values used for two objective problems can be found in table 1.

3.1.2 Three Objective Problems

When dealing with 3 objective problems we have to vary both α and β for the DWAGA system. Consequently, there are two periods for the 3-objective DWAGA system, with β cycling through its settings for every setting of α. Instead of complicating maters with two user-defined parameter both periods are set to be inversely proportional to the number of generations. Also the DWAGA system only goes through one 90º rotation for both α and β instead of 180º.

The details for parameter values used for three objective problems can again be found in table 1. All experiments are repeated 30 times for statistical accuracy.

Table 2. Function definitions for two tri-objective functions used in the test suite

F12	F13
$f_1 = x_1^2 + (x_2 - 1)^2$	$f_1 = 0.5(x_1^2 + x_2^2) + \sin(x_1^2 + x_2^2)$
$f_2 = x_1^2 + (x_2 + 1)^2 + 1$	$f_2 = \frac{(3x_1 - 2x_2 + 4)^2}{8} + \frac{(x_1 - x_2 + 1)^2}{27} + 15$
$f_3 = (x_1 - 1)^2 + x_2^2 + 2$	$f_3 = \frac{1}{x_1^2 + x_2^2 + 1} - 1.1\exp(-x_1^2 - x_2^2)$
where $-2 \le x_1, x_2 \le 2$	where $-3 \le x_1, x_2 \le 3$

3.1.3 Test Functions

Since we are reconstructing the experiments of the creators of DWA we are comparing the NSGA-II algorithm against the DWAGA on the same five test functions that were used by them in [3], which we similarly label F1 to F5 (three of them F2, F3 and F5 were also used in [6] and called T1 to T3.

In addition we are using an extra six multi-objective test functions that are used in test suites [6] and [7]. F6 – F8 corresponds to F3 – F5 as found in [7] and F9 to F11 corresponds to T4 – T6 as found in [6].

We then tried two tri-objective functions to see how the two algorithms scale, see Table 2 for function definitions.

3.1.4 Performance Measures

The performance of the EMOO systems is evaluated by examining the following measures as suggested by [8]: the spacing, diversity, coverage and execution time of the respective systems. Again, all measurement statistics are based on 30 repetitions.

Spacing is a measure of how evenly the solutions are spaced on the Pareto front. Each distance between neighbouring solutions is compared against the average of the distance between neighbours. If all solutions are evenly spaced, the measure will read 0, the more non-uniform the distribution along the Pareto front, the higher the number. The formula for Spacing is:

$$S = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n-1}(d_i - \bar{d})^2} \tag{5}$$

where d_i is the distance between two neighbouring solutions and $\bar{d}$ is the average distance between neighbours.

In the case of three objective problems the Pareto front is a plane instead of a line. As a result the distance there is measured between a solution and its closest neighbour.

Diversity is similar to Spacing, but instead of being based on the L_2-norm (associated with the Euclidean distance) it is based on the L_1-norm (associated with the Hamming distance). Also, Diversity is designed to take into account the full range of the Pareto front. With Spacing, the system could produce solutions that are evenly spaced but only cover a small section of the Pareto front, yet produce the same result as a system that evenly covers the entire Pareto front. Diversity compensates for this effect.

$$Diversity = \frac{d_f + d_l + \sum_{i=1}^{N-1} | d_i - \bar{d} |}{d_f + d_l + (N-1)\bar{d}} \tag{6}$$

Here d_f and d_l are the distances between the end points of the found Pareto front and the (known) extreme solutions of the true Pareto front. N is the size of the solution set.

In the case of three objective problems the corners of the Pareto front plane are taken as the extreme solutions.

Coverage of Two Sets: this measure compares the size of the Pareto front from one of the optimization techniques with the size of the Pareto front formed from the combined fronts of each of the two techniques.

$$Coverage\_1(\alpha) = \#(A \cap C) \,/\, \#(C) \tag{7}$$

$$Coverage\_2(\alpha) = \#(A \cap C) \,/\, \#(A) \tag{8}$$

$$Coverage\_3(\alpha,\beta) = (\#(A \cap C) - \#(A \cap B \cap C)) /\, \#(C) \tag{9}$$

where A is a Pareto front found by algorithm α, B is a Pareto front found by algorithm β, and C is a Pareto front formed when combining Pareto fronts A and B. Coverage_1(α) is the percentage of the combined Pareto front discovered by algorithm α and Coverage_2(α) is the percentage of the Pareto front discovered by α that is used in the combined Pareto front. Coverage_3(α,β) is the percentage of the combined Pareto front discovered by algorithm α that was not discovered by algorithm β.

Execution time: the time it took on the computer that executed the two algorithms. Both programs were written in Java and run on AMD Athlon XP 1800, with a CPU Clock speed of 1150Mhz and with 512MB of RAM DDR of memory.

Through experimentation it was discovered that the coverage-of-two-sets measurement was the most important measurement; often by itself it was informative enough to determine which algorithm is better. When the Coverage measurement did not indicate a clear winner, the diversity measurement was a good way of breaking the tie and determining the winner. When the diversity measurement did not indicate a clear winner, the spacing measurement was used to break the tie.

Finally, for statistical accuracy, all experiments have been run 30 times each for each setting, i.e. all statistics are based on 30 repetitions.

4 Results

4.1 Results When NSGA-II Is Victorious

The Coverage measurements indicate that for all these test functions the combined Pareto front consists entirely of the solutions found by NSGA-II algorithm (see Table 3). This clearly shows that DWA is inferior for these test functions. Since the performance difference on the coverage measurements between these two methods is so drastic, further measurements on diversity and spacing are not necessary.

Table 3. Coverage_1 and Coverage_2 measurements for the NSGA-II and DWAGA algorithms

	Coverage_1(DWA)				Coverage_1(NSGA)			
	avg	std	Conf Interval		avg	std	Conf Interval	
f2	0.0%	0.0%	0.0%	0.0%	100.0%	0.0%	100.0%	100.0%
f3	0.0%	0.0%	0.0%	0.0%	100.0%	0.0%	100.0%	100.0%
f4	0.1%	0.3%	-0.2%	0.3%	99.9%	0.3%	99.7%	100.2%
f5	0.0%	0.0%	0.0%	0.0%	100.0%	0.0%	100.0%	100.0%
f9	0.0%	0.0%	0.0%	0.0%	100.0%	0.0%	100.0%	100.0%
f10	7.7%	19.8%	-7.5%	22.9%	92.3%	19.8%	77.1%	107.5%
f11	0.0%	0.0%	0.0%	0.0%	100.0%	0.0%	100.0%	100.0%

	Coverage_2(DWA)				Coverage_2(NSGA)			
	avg	std	Conf Interval		avg	std	Conf Interval	
f2	0.0%	0.0%	0.0%	0.0%	100.0%	0.0%	100.0%	100.0%
f3	0.0%	0.0%	0.0%	0.0%	100.0%	0.0%	100.0%	100.0%
f4	0.1%	0.5%	-0.3%	0.5%	100.0%	0.0%	100.0%	100.0%
f5	0.0%	0.0%	0.0%	0.0%	100.0%	0.0%	100.0%	100.0%
f9	0.0%	0.0%	0.0%	0.0%	100.0%	0.0%	100.0%	100.0%
f10	11.2%	27.3%	-9.9%	32.2%	97.0%	16.3%	84.5%	109.6%
f11	0.0%	0.0%	0.0%	0.0%	100.0%	0.0%	100.0%	100.0%

avg = average std= standard deviation Conf Interval= Confidence Interval

4.2 Results When NSGA-II Is Challenged on Two Objective Problems

When the DWAGA and NSGA-II algorithms were tested on functions f1, f6, f7, and f8 it was observed that the NSGA-II was no longer a clear favorite and the DWAGA even had the superior performance on some test functions.

As the behavior of the algorithms on each of these four test functions is so diverse, each of the four test functions will be examined in detail one at a time.

4.2.1 F1 Comparison Results

For F1 the combined Pareto front consists half from DWA and half from NSGA-II. As can be seen from the confidence intervals for coverage in table 4 the NSGA-II slightly outperforms DWA, but since the difference is this small it is important to also evaluate Diversity and spacing in order to be sure which algorithm is better. It can be seen in Table 5 that NSGA-II is better in both spacing and diversity and as a result NSGA-II should be considered the better performer on F1 (but DWA is very close).

4.2.2 F6 Comparison Results

For F6 the combined Pareto front consists 1/3 from DWA and 2/3 from NSGA-II. As can be seen from the confidence intervals, the NSGA-II outperforms DWA in coverage, but it can be seen that DWA also contributes good solutions since 1/3 is a decent proportion, and so we evaluate diversity and spacing. In the Diversity and spacing the NSGA-II outperforms DWA. When these 3 measurements are considered together it is clearly seen that NSGA-II performs better.

Table 4. Comparing[2] the Coverage_1 and Coverage_2 measurements for the NSGA-II and DWAGA algorithms

	Coverage_1(DWA)				Coverage_1(NSGA)			
	avg	std	Conf Interval		avg	std	Conf Interval	
F1	46.9%	3.2%	3.2%	7.3%	53.1%	3.2%	50.6%	55.6%
F6	33.4%	3.5%	3.5%	6.3%	66.6%	3.5%	63.9%	69.4%
F7	69.1%	1.7%	67.8%	70.4%	30.9%	1.7%	29.6%	32.2%
F8	59.0%	21.3%	42.5%	75.4%	41.0%	21.3%	24.6%	57.5%

	Coverage_2(DWA)				Coverage_2(NSGA)			
	avg	std	Conf Interval		avg	std	Conf Interval	
F1	75.1%	7.3%	69.4%	80.7%	88.9%	3.6%	86.1%	91.7%
F6	44.2%	6.3%	39.3%	49.1%	92.4%	2.2%	90.7%	94.1%
F7	93.8%	2.2%	92.0%	95.5%	54.0%	4.0%	51.0%	57.1%
F8	71.1%	25.3%	51.7%	90.6%	66.2%	34.0%	40.0%	92.5%

Table 5. Comparing[3] the Spacing and diversity for the NSGA-II and DWAGA algorithms on four bi-objective functions

Spacing							
	Rank (D)	Rank (N)	s	p-value	bonf corr. p-value	Better	Statistically Significant
f1	45.5	15.5	2.37	4.1E-19	1.9E-17	NSGA	Yes
f6	42.0	19.0	3.20	1.4E-09	6.8E-08	NSGA	Yes
f7	8.0	45.5	2.44	4.1E-22	2.0E-20	DWA	Yes
f8	42.9	17.5	2.64	1.3E-13	6.0E-12	NSGA	Yes
Diversity							
	Rank (D)	Rank (N)	Pooled Std. Dev.	p-value	bonf corr p-value	Better	Statistically Significant
f1	45.5	15.5	2.27	4.1E-19	2.0E-17	NSGA	Yes
f6	45.5	15.5	3.20	3.2E-13	1.5E-11	NSGA	Yes
f7	25.5	27.5	2.44	4.1E-01	19.6	DWA	No
f8	15.8	45.2	2.64	5.3E-16	2.6E-14	DWA	Yes

4.2.3 F7 Comparison Results

For F7 the combined Pareto front consists 2/3 from DWA and 1/3 from NSGA-II. As can be seen from the confidence intervals for Coverage measure, this time the DWA outperforms NSGA-II. To be certain that DWA is in fact better than NSGA-II we first looked at Diversity, but since results of this test are inconclusive (the two algorithms can't be statistically differentiated based on this test), spacing becomes the determining factor. Here the results are in DWA favour. Based on these three measurements one can conclude that DWAGA is the better method for solving F7.

This is an important result for the research in DWA because F7 has a concave Pareto front. It has been assumed that DWA would have problems with solving this

[2] The confidence intervals are formed using the normal parametric approach as the results were found to be normally distributed when using a normality plot.

[3] The results were found to not be normally distributed, so the T test was done on the ranks (a non-parametric test).

type of a function but not only did it solve the problem well but it also outperformed NSGA-II.

4.2.4 F8 Comparison Results

For F8 the combined Pareto front consists 3/5 from DWA and 2/5 from NSGA-II. As can be seen from the confidence intervals for Coverage measure it is inconclusive which algorithm is better. The T-test in Table 6 confirms this. As a result we look at Diversity of the two methods where DWA outperforms NSGA-II. So, based on these measurements we conclude that DWAGA performs better than NSGA-II on F8.

Table 6. T Test[4] for looking in more detail if there is an advantage in coverage for DWAGA. It can be seen that it cannot be determined that DWA has better coverage than NSGA-II.

T Test on NSGA Coverage – DWA Coverage for f8			
α	0.01	Diff(f8)	0.179
No. of Ind. tests	48	pooled std	0.0551
α / 48(see footnote[5])	0.00021	conf. interval	-0.053
N	30		0.412
T	4.22	t-score	3.2578
		p-value	0.0019
		p-value * 48 (see footnote[3])	0.0902

4.3 Results When Run on 3 Objective Problems

The Coverage_1 measurements in Table 7 indicate that for functions F12 and F13, the combined Pareto front consists almost entirely of the solutions found by DWAGA algorithm while the NSGA-II had found a smaller part of the Pareto front. The Coverage_2 results indicate that both algorithms find same quality of solutions because almost all solutions found by each algorithm are used in the combined Pareto front. The Coverage_3 results indicate that the DWAGA has identified a large number of solutions that the NSGA-II was unable to find. The DWAGA managed to find almost all the solutions that NSGA-II identified plus many more. As a result the DWAGA provided a better and more detailed representation of the Pareto front and outperformed the NSGA-II.

As can be seen in Table 8, the DWAGA is executing much faster than NSGA-II, which is a big benefit with the huge search spaces that are associated with multi-objective problems.

This shows a possible deficiency in the Pareto front style approach. When a search space gets large, the NSGA seems to have trouble finding many solutions and is negatively impacted in its performance time. For example by switching from 2 objectives to 3, the Pareto front has changed from a line to a plane. As the number of objectives increases, the size of the Pareto front increase geometrically in the size.

[4] The regular T test was used as the results were found to be normally distributed when using a normality plot.

Table 7. The Coverage_1, Coverage_2, and Coverage_3 measurements for the NSGA-II and DWAGA algorithms on two tri-objective functions

	Coverage_1(DWA)				Coverage_1(NSGA)			
	avg	std	95% Confidence Interval		avg	std	95% Confidence Interval	
F12	0.9729	0.0029	0.9719	0.9739	0.5203	0.00539	0.5182	0.5223
F13	0.9007	0.0821	0.8713	0.9300	0.7108	0.02205	0.7029	0.7187

	Coverage_2(DWA)				Coverage_2(NSGA)			
	avg	std	95% Confidence Interval		avg	std	95% Confidence Interval	
F12	0.9997	0.0005	0.9995	0.9999	0.9791	0.0075	0.9765	0.9818
F13	0.9859	0.0110	0.9820	0.9898	0.9889	0.0044	0.9873	0.9905

	Coverage_3(DWA)				Coverage_3(NSGA)			
	avg	std	95% Confidence Interval		avg	std	95% Confidence Interval	
F12	0.4797	0.0054	0.4777	0.4816	0.0271	0.0029	0.0261	0.0281
F13	0.2892	0.0220	0.2813	0.2971	0.0993	0.0821	0.0700	0.1287

Table 8. The algorithm run-time measurements for NSGA-II and DWAGA

	Time (DWA)				Time (NSGA)			
	avg	std	95% Confidence Interval		avg	std	95% Confidence Interval	
F12	333009	151033	278962	387055	612279	5148	610437	614122
F13	442595	92429	409520	475670	4257111	88272	4225523	4288699

Consequently, to find this Pareto front, an algorithm must find a proportionately greater number of solutions. Since the NSGA-II stores the Pareto front solutions in its population it requires an geometrically larger population size because once a population member finds an optimal solution it will keep that solution to the end, especially with elitism. This causes more and more of the population members to be used for storing solutions instead of exploring. Eventually near the end of the run only few population members will remain free to explore. In order to have the NSGA-II be able to explore a large search space and be able to store solutions that represent it well, it will require the possession of a very large population. This will cause the algorithm to run slowly, due to the fact that it has to perform fitness calculations as well as the time taken sorting this huge population.

This problem does not apply to the DWAGA, which has an archive to store all the best solutions. It can have a smaller population, which can be used only for the searching of new solutions and not have to try to maintain all the best solutions. This allows the algorithm to identify a very large solution space with a relatively small population. This seems to allow the DWAGA to scale better than NSGA-II for problems with higher number of objectives.

5 Conclusion

In this paper, we compared two EMOO methods against each other: the Non-dominated Sorting Genetic Algorithm (NSGA-II) and the Dynamic Weighted Aggregation (DWA) system. To make the comparison fair and to remove an extra factor from the analysis, the DWA has been layered on top of a GA instead of and ES algorithm that it was originated for (since the DWA can be easily applied to any EC system). Using various traditional EMOO measures, such as Coverage, Spacing and Diversity, we determined that the DWA could handle concave problems as advertised. Furthermore, while most of the bi-objective functions we tried were better handled by the NSGA-II, when tri-objective problems were used, the DWAGA outperforms the NSGA-II and runs much faster. We believe that the cause of the DWAGA's success at higher number of objectives is due to its use of an archive, alleviating the need of the storage of the Pareto front (which can grow exponentially with the number of objectives) within the population itself.

Acknowledgements

The Authors would like to acknowledge the Natural Science and Engineering Research Council of Canada for support of this research.

References

[1] Coello Coello, C.A.: A Short Tutorial on Evolutionary Multiobjective Optimization. In Zitzler, E., Deb, K., et. al. (eds.): *The First International Conference on Evolutionary Multi-Criterion Optimization* (EMO 2001). Springer, Berlin (2001) 21-40

[2] Jin, Y., Okabe, T., Sendhoff., B.: Adapting Weighted Aggregation for Multiobjective Evolution Strategies. In Zitzler, E., Deb, K., et. al. (eds.): *The First International Conference on Evolutionary Multi-Criterion Optimization* (EMO 2001). Springer, Berlin (2001) 96-110

[3] Jin, Y., Okabe, T., Sendhoff., B.: Dynamic Weighted Aggregation for Evolutionary Multi-Ojbective Optimization: Why Does It Work and How? In Spector L., et. al. (eds.): *GECCO 2001 - Proceedings of the Genetic and Evolutionary Computation Conference*. Morgan Kaufmann, San Francisco (2001) 1042-1049

[4] Srinivas, N. and Deb, K.: Multi-Objective function optimization using non-dominated sorting genetic algorithms. *Evolutionary Computation*, 2(3):221–248 (1995)

[5] Deb, K., Goel, T.: Controlled Elitist Non-dominated Sorting Genetic Algorithms. In Zitzler, E., Deb, K., et. al. (eds.): *The First International Conference on Evolutionary Multi-Criterion Optimization* (EMO 2001). Springer, Berlin (2001) 67-81

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

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

[8] Ang, K.H., Chong, G., Li, Y.: Preliminary Statement on the Current Progress of Multi-Objective Evolutionary Algorithm Performance Measurement. In Eberhart R., Fogel, D.B. (eds.): *Proceedings of the 2002 Congress on Evolutionary Computation* (CEC'02). IEEE Press (2002) 1139-1144

[9] Yaochu Jin, Tatsuya Okabe and Bernhard Sendhoff. Solving Three-objective Optimization Problems Using Evolutionary Dynamic Weighted Aggregation: Results and Analysis. In: *Proceedings of Genetic and Evolutionary Computation Conference*. pp.636, Chicago, 2003

[10] Czyzak, P., and Jaszkiewicz, A.: Pareto simulated annealing - a metaheuristic technique for multiple-objective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis* vol. 7, pp. 34-47 (1998).

[11] Hansen, P.H.: Tabu Search for Multiobjective Optimization: MOTS. *Proceedings of the 13th International Conference on Multiple Criteria Decision Making*, 1997.

[12] Serafini P.: Simulated annealing for multi objective optimization problems. In Multiple Criteria Decision Making: Expand and Enrich the Domains of Thinking and Application, G.H. Tzeng, ed., Springer (1993).

[13] Ulungu, E., Teghem, J., Fortemps, P., and Tuyytens, D.: MOSA Method: A Tool for Solving Multiobjective Combinatorial Optimization Problems. *Journal of Multi-Criteria Decision Analysis*, vol. 8/4, pp. 221-236 (1999).

[14] Joshua Knowles and David Corne: Memetic Algorithms for Multiobjective Optimization: Issues, Methods and Prospects. 2003