Towards Understanding the Cost of Adaptation in Decomposition-Based Optimization Algorithms

Ioannis Giagkiozis, Robin C. Purshouse and Peter J. Fleming
Department of Automatic Control and Systems Engineering
The University of Sheffield
Sheffield, UK
{i.giagkiozis,r.purshouse,p.fleming}@sheffield.ac.uk

***Abstract*—Decomposition-based methods are an increasingly popular choice for a posteriori multi-objective optimization. However the ability of such methods to describe a trade-off surface depends on the choice of weighting vectors defining the set of subproblems to be solved. Recent adaptive approaches have sought to progressively modify the weighting vectors to obtain a desirable distribution of solutions. This paper argues that adaptation imposes a non-negligible cost – in terms of convergence – on decomposition-based algorithms. To test this hypothesis, the process of adaptation is abstracted and then subjected to experimentation on established problems involving between three and 11 conflicting objectives. The results show that adaptive approaches require longer traversals through objective-space than fixed-weight approaches. Since fixed weights cannot, in general, be specified in advance, it is concluded that the new wave of decomposition-based methods offer no immediate panacea to the well-known conflict between convergence and distribution afflicting Pareto-based a posteriori methods.**

***Keywords*—Decision support systems, multi-objective optimization, decomposition, adaptation.**

I. Introduction

Many decision problems have multiple performance criteria, or objectives, that must be considered simultaneously. If these objectives are in conflict, such that an improvement in one objective cannot be achieved without detriment to another objective, then the problem will have more than one solution that can be considered optimal. In order to select a single solution to the problem, it is necessary to resolve the trade-off between competing solutions by introducing subjective preferences that a decision-maker has for varying levels of performance across the objectives.

Expression of preferences is often difficult for a decision-maker, and so the analyst will typically develop a decision support system that attempts to reveal the performance trade-offs prior to eliciting the preference information that can identify a single desired solution. The task of the optimizer in this system is to find a set of *Pareto optimal* solutions that can well describe the trade-offs. This process is known as *a posteriori multi-objective optimization* [1, pp. 63]. The quality of the solution set is measured in terms of (i) the convergence, or proximity, to global optimality; and (ii) the distribution, or diversity, of solutions across the range of available trade-offs.

Population-based optimizers, such as evolutionary algorithms, have proved a popular choice for a posteriori optimization, principally because they offer the potential for a parallel search aimed at both good convergence and good distribution. Despite arguments for the existence of a meta-trade-off between convergence and distribution in a posteriori methods [2], good performance can generally be achieved for decision problems with two or three conflicting criteria. However recent evidence has indicated that the meta-trade-off becomes increasingly difficult to manage as the number of conflicting criteria rises beyond three [3]–[5]. Much of the analysis has focused on *Pareto-based* algorithms, where the convergence process relies on a partial ordering of candidate solutions imposed by a dominance operator. This partial ordering collapses rapidly with increasing numbers of conflicting criteria, leading to either a random search or an illusory search for a 'good' distribution, depending on the mechanics of the algorithm [4], [6].

Decomposition-based methods, which transform a single multi-objective decision problem into multiple single-objective problems to be solved in parallel [7], [8], have recently gained favour as they are believed not to possess the convergence issues associated with the dominance operator. Although this assertion feels intuitive – and is reinforced by empirical findings [5], [6], [8]–[10] – it cannot logically be concluded that decomposition-based methods are not affected by similar issues. For example, the Chebyshev scalarizing function, which is very often employed in decomposition-based methods, is identical to the dominance relation in terms of the probability of finding an improved solution, which has implications for convergence [11].

The seminal decomposition-based methods, e.g. MOEA/D [8], have employed a set of weighting vectors that is unchanged over the course of the optimization. This means that the subproblems to be solved remain fixed. If we are prepared to speculate that, in Pareto-based methods, the dominance relation is equivalent to a Chebyshev scalarizing function under certain conditions, then we can reinterpret a population of candidate solutions as defining a set of subproblems to be solved. Since the population changes during the course of the optimization, this implies that the subproblems are varying also. Intuitively, trying to solve a fixed problem is easier than trying to solve one that varies. Could this be the key reason why decomposition-based methods offer improved performance?

Decomposition-based methods have a requirement to identify a suitable distribution of weighting vectors that can pro-

duce a desired (usually even) distribution of Pareto optimal solutions. Whilst pioneers of decomposition recognised this requirement, they remained somewhat incoherent on the extent to which proposed weighting schemes (e.g. even distribution on the $(k-1)$-simplex [8]), when combined with favoured scalarizing functions (e.g. Chebyshev [8]), would produce the desired outcome. This incoherence provided an early motivation for adaptive approaches that would progressively identify a suitable distribution of weighting vectors during the optimization process. In recent work, [12], we have shown that if the geometry of the problem is known then an optimal set of weighting vectors can, in fact, be readily identified. However, given that the geometry is rarely available in practice for real-world problems, the motivation for adaptation remains.

In one of the earliest attempts at adaptation, Jin et al. [13], [14] employed concepts from control theory, e.g. *bang-bang* control, to *sweep* the entire Pareto front in the hope of obtaining a representative Pareto optimal set. Although these ideas are inspired, there are several questions that are troubling. For instance, what would be an optimal rate of change for the weighting vectors, or, how would the weighting vectors be scanned for problems with more than two objectives? Later, Jaszkiewicz [15] suggested the use of uniformly distributed weighting vectors on the $(k-1)$-simplex. As in the previous case, the method introduced by Jaszkiewicz was not constrained to a particular region of the Pareto front; rather, weighting vectors were allowed to change across the entire $(k-1)$-simplex thus *scanning* the entire front. A similar method to [15] was also employed by Mahfouf et al. [16], coupled with a particle swarm optimization algorithm. In a more elaborate method, Jiang et al. [17], [18] attempted to identify the Pareto front geometry and then use this information to find a set of weighting vectors that will define subproblems that will result in solutions that are well distributed across the Pareto front. However, again, the weighting vectors were adapted on every iteration of the optimization procedure. Later it was shown that the weighting vectors, in combination with the selected scalarizing function, fully determine the location to which solutions tend to converge [12].

In this work, we argue that there is an associated and potentially non-negligible cost – in terms of convergence – to adapting the weighting vectors in decomposition-based algorithms. Varying the weighting vectors inadvertently changes the set of subproblems over the course of the optimization. This can potentially create difficulties for the algorithm which may lead to slower convergence, while there is no guarantee that the promised benefits from this adaptation will materialise. We argue further that this adaptation may lead to an equivalence with Pareto-based methods under certain conditions. Although much more effort is required to clarify this connection, the results presented in this work can be considered as the first step in this direction. So, the questions we explore in this work can be summarised as follows: given the fact that variation of weighting vectors in decomposition-based algorithms is equivalent to variation of the problem for which a solution is sought, does this affect algorithm convergence? If so, to what extent?

The contributions of this work can be summarised as follows:

- We highlight that Pareto-based algorithms essentially attempt to solve a varying problem. This is mainly due to the interplay of clustering, fitness assignment and archiving in Pareto-based methods.
- We test a hypothesis that we introduced in [11], namely, that decomposition-based algorithms that employ the Chebyshev scalarizing function appear to exhibit superior performance to Pareto-based algorithms simply because the former use constant weighting vectors. This amounts to solving a fixed problem as opposed to a varying problem.
- We signal that the current trend to adaptive decomposition-based methods should be reevaluated as it appears that if this path is followed it is highly likely to obtain decomposition-based methods that in essence replicate Pareto-based algorithms and as such inherit their deficiencies.

The rest of this paper is organised as follows. In Section II we present the necessary background information. In Section III we describe our experiments and present the results. In Section IV we discuss our findings and present further hypotheses. Finally we summarise and conclude the present work in Section V.

II. Background

The standard definition of a multi-objective problem (MOP) is:

$$\begin{aligned} \min_{\mathbf{x}} \mathbf{F}(\mathbf{x}) &= \left(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})\right), \\ \text{subject to } \mathbf{x} &\in D \subseteq \mathbb{R}^n, \end{aligned} \tag{1}$$

where k is the number of scalar objective functions $f(\cdot)$ and also the dimension of the problem in objective space. D is the *feasible region*. The vector of variables, $\mathbf{x}$, in this context is often referred to as decision vector while $\mathbf{z} = \mathbf{F}(\mathbf{x})$ is referred to as objective vector. We typically assume that the individual scalar objective functions in (1) are in *conflict* [19], leading to a potentially $(k-1)$-dimensional trade-off required to be represented via a posteriori methods.

Two mainstream methods employed for comparing candidate solutions in an a posteriori approach are Pareto-based and decomposition-based techniques. Both labels disguise a wide range of heterogeneity, however the methods be briefly contrasted as follows:

- Pareto-based methods directly employ a relationship that induces a partial ordering in objective space, and an implicit partial ordering in decision space. A variety of supplementary schemes – often based around regularizing solution density in objective-space – are used to direct the search towards a good distribution.
- Decomposition-based methods use a scalarizing function to *decompose* the mutli-objective problem into a set of single-objective problems. The quality of distribution is implicitly encoded by the subproblems.

Although there are several concepts of optimality in multi-objective optimization, weak Pareto optimality is arguably the most commonly employed [1, pp. 19]. In what follows we

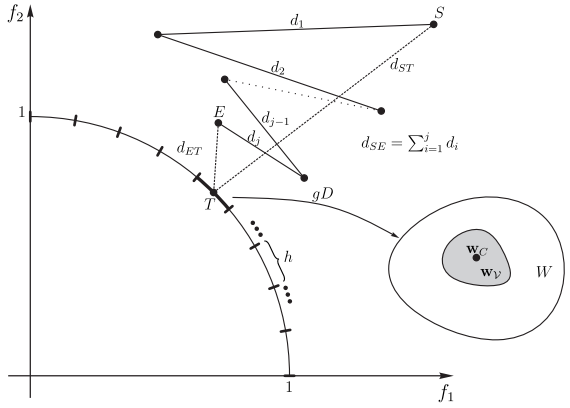

Fig. 1. Experiment setting. Note that scales have been exaggerated for illustrative purposes.

assume, without loss of generality, that all objectives are to be minimized. In brief, a solution, $\mathbf{x}$ weakly dominates a solution $\mathbf{y}$ iff $f_i(\mathbf{x}) < f_i(\mathbf{y})\ \forall i$. This relation allows a partial ordering to be imposed across all $\mathbf{x} \in D$. If there exists no other solution $\mathbf{y} \in D$ such that $f_i(\mathbf{y}) < f_i(\mathbf{x})\ \forall i$ then $\mathbf{x}$ is defined as weakly Pareto optimal.

A widely used family of scalarizing functions, especially in the evolutionary multi-objective optimization community, is the weighted metrics. This function is usually defined as:

$$g_p(\mathbf{x}) = \|\mathbf{w} \circ |\mathbf{F}(\mathbf{x}) - \mathbf{z}^\star|\|_p, \tag{2}$$

where $\circ$ is the Hadamard product and represents element-wise vector or matrix multiplication and $\mathbf{z}^\star$ is the ideal vector [1, pp. 15]. Taking the limit, $p \to \infty$, is a special case of (2) and is commonly referred to as the Chebyshev scalarising function:

$$\begin{aligned} g_\infty(\mathbf{x}) &= \|\mathbf{w} \circ |\mathbf{F}(\mathbf{x}) - \mathbf{z}^\star|\,\|_\infty \\ &= \max\{w_1|f_1(\mathbf{x}) - z_1^\star|, \ldots, w_k|f_k(\mathbf{x}) - z_k^\star|\}. \end{aligned} \tag{3}$$

Although (3) is not differentiable it can guarantee to produce all Pareto optimal solutions for some weighting vector, if the underlying algorithm can converge. Namely, it can *reach* every Pareto optimal point. As mentioned in the introduction, this scalarizing function has a very similar effect to Pareto-dominance relations.

Using (2) for some $p > 1$, and a set of weighting vectors, all of which must be $\mathbf{w} \succeq \mathbf{0}$, a good representation of the Pareto-front can be obtained, see for example [7], [8], [20], [21].

III. Experiment Setting

To test our hypotheses we have devised the following experiment. A generic random search algorithm is employed, namely, a decision vector is selected at random, using a uniform distribution in decision space. Subsequently, after evaluating the resulting decision vector to obtain its corresponding objective vector, the Euclidean distance of the objective vector from the target Pareto optimal point is measured. These two vectors are denoted as S and T in *Fig. 1*, respectively, and their distance d_{ST}. This concludes the initialisation phase of the random search algorithm. Next, the decision space is sampled sequentially $1\,000\,000$ times and the following decision procedure is employed after the generation of a new sample:

- Evaluate the new sample using the objective function, $\mathbf{F}(\mathbf{x})$.
- Evaluate the fitness of the new sample using the Chebyshev scalarizing function (3) the corresponding objective vector and a weighting vector, $\mathbf{w}$.
- If the resulting *fitness* from the Chebyshev function is smaller than the best so far solution then the best solution is updated.
- When a *better* solution is generated, its Euclidean distance from the previous solution is added to a running total, d_{SE} in *Fig. 1*. This sum represents the *distance covered* in objective space, and is the trajectory that solutions follow as they attempt to approach the target Pareto optimal point, T.

The same procedure is applied twice for the same seed in the random number generator, once with a constant weighting vector, $\mathbf{w}_C$, and one more time with a varying weighting vector, $\mathbf{w}_\mathcal{V}$. The neighbourhood of variation for the weighting vector can be seen in *Fig. 1*. This neighbourhood is selected by calculating the mean nearest neighbour distance of algorithms employed for the selected dimensions in objective space - namely 3, 5, 7 and 11 as shown in Table I. In the following section we elaborate on the details in selecting the varying weighting. This experiment was conducted on a subset of the DTLZ [22] and WFG [23] problem sets. The criterion for inclusion in this work was that the Pareto front have a geometry that would allow us to predict the location on the Pareto front to which a subproblem will converge. This for example could be accomplished for the WFG3 test problem, however as the objective space is like an acute cone the number of solutions near the Pareto front, which is a line, would pose significant difficulties for the random search algorithm. Additionally, to remove any bias towards a particular objective, we have normalized these problems so that the Pareto front is either the positive orthant of the unit hypersphere (WFG4-9 and DTLZ2-4) or the $(k-1)$-simplex, which is the case for DTLZ1. Therefore in the case of the DTLZ1 test problem the following condition holds for Pareto optimal solutions: $\sum_{i=1}^{k} f_i(\mathbf{x}) = 1$.

TABLE II. Mean nearest neighbour distance, h, for the different problem instances used in this work. N is the equivalent population size that if distributed evenly across the Pareto front would result in a mean nearest neighbour distance h.

Objectives, k	3	5	7	11
h	0.0524	0.1571	0.2357	0.2828
N	406	715	924	3003

A. Weighting Vectors

In our experiment we specify a desired location on the trade-off surface (which would be a component of some desired trade-off distribution). We simulate an adaptive process as a region of uncertainty around this desired location. In terms of

TABLE I. THE LEFT FOUR COLUMNS ILLUSTRATE THE MEAN DISTANCE AND ITS STANDARD DEVIATION, OF THE BEST SOLUTION FOUND TO THE TARGET SOLUTION T FOR CONSTANT AND VARYING WEIGHTING VECTOR RESPECTIVELY. THE RIGHT FOUR COLUMNS ILLUSTRATE THE MEAN AND STANDARD DEVIATION OF THE $\frac{d_{SE}}{d_{ST}}$ QUANTITY. THIS REPRESENTS THE DISTANCE TRAVERSED IN OBJECTIVE SPACE WITH RESPECT TO THE INITIAL DISTANCE OF THE POINT S FROM T.

		l_2 distance to target point T				Multiple of d_{SE} distance traversed			
		Constant - $\mathbf{w}_C$		Varying - $\mathbf{w}_\mathcal{V}$		Constant - $\mathbf{w}_C$		Varying - $\mathbf{w}_\mathcal{V}$	
		mean	std	mean	std	mean	std	mean	std
3 obj	DTLZ1	144.4935	16.7053	**144.2508**	16.8302	3.2194	0.9754	**3.1512**	1.1481
	DTLZ2	$0.6845^\dagger$	0.0469	$0.6952^\dagger$	0.0556	**4.9076**	1.8303	4.9555	1.9927
	DTLZ3	**190.5301**	31.9463	192.9661	29.6660	**3.7472**	1.1408	3.7814	1.1897
	DTLZ4	$0.7519^\dagger$	0.0448	$0.7482^\dagger$	0.0535	**2.7291**	0.8677	2.7910	0.8825
	WFG4	**0.0794**	0.0142	0.0798	0.0143	**3.7938**	1.1758	4.0920	1.3697
	WFG5	**0.0724**	0.0075	0.0777	0.0087	$3.1386^\dagger$	1.1049	3.6714	1.1690
	WFG6	**0.0790**	0.0153	0.0825	0.0160	3.9585	1.1342	**3.9273**	1.1171
	WFG7	**0.0531**	0.0097	0.0568	0.0093	**3.3947**	0.9296	3.5258	1.0358
	WFG8	**0.0956**	0.0092	0.0985	0.0100	3.1991	1.2054	**3.1846**	0.9957
	WFG9	**0.0326**	0.0078	0.0382	0.0108	$4.1610^\dagger$	1.4493	4.2759	1.6581
5 obj	DTLZ1	**67.8381**	10.4022	68.7036	10.0983	4.7688	1.7869	**4.6265**	1.3961
	DTLZ2	**0.8343**	0.0319	0.8414	0.0292	$2.6253^\dagger$	1.1343	2.8659	1.1989
	DTLZ3	**97.7828**	19.9067	99.9511	19.1352	4.6545	1.7736	**4.5705**	1.6507
	DTLZ4	**0.8792**	0.0290	0.8873	0.0262	**2.3701**	0.6079	2.4859	0.6948
	WFG4	**0.1809**	0.0316	0.2117	0.0427	**4.2611**	1.1664	4.8248	1.4613
	WFG5	**0.1310**	0.0135	0.1527	0.0244	**5.4966**	1.8136	6.6102	2.0830
	WFG6	**0.1301**	0.0189	0.1517	0.0269	**5.7494**	1.5537	6.9240	2.2242
	WFG7	**0.1240**	0.0177	0.1512	0.0221	**4.8372**	1.7918	5.5069	1.6631
	WFG8	$0.2170^\dagger$	0.0207	0.2277	0.0203	**5.9583**	1.8287	6.9868	2.1210
	WFG9	**0.0893**	0.0211	0.1247	0.0357	**5.6870**	1.7024	6.4356	2.2510
7 obj	DTLZ1	**35.2566**	5.9010	36.7003	5.9289	**5.0804**	1.5432	5.3688	1.5851
	DTLZ2	**0.8942**	0.0165	0.8991	0.0201	**1.6928**	0.6557	1.7088	0.6122
	DTLZ3	**58.3713**	10.7935	58.8268	9.7380	**5.6023**	1.9782	5.7535	2.1258
	DTLZ4	**0.9335**	0.0180	0.9363	0.0141	**2.1783**	0.5273	$2.4194^\dagger$	0.7649
	WFG4	**0.3085**	0.0709	0.3886	0.0729	**4.3738**	1.0940	5.1883	1.2384
	WFG5	**0.1723**	0.0319	0.2069	0.0379	**7.1909**	2.2791	7.6020	2.4588
	WFG6	**0.1533**	0.0230	0.2062	0.0426	$7.7440^\dagger$	2.5615	9.1902	3.0116
	WFG7	**0.1867**	0.0187	0.2381	0.0445	**5.7042**	1.8572	5.9257	2.0361
	WFG8	$0.2660^\dagger$	0.0330	0.3142	0.0471	**7.3345**	2.3912	8.0492	2.4481
	WFG9	$0.1269^\dagger$	0.0374	0.1700	0.0425	**7.0071**	1.9931	8.1374	2.5252
11 obj	DTLZ1	**12.0409**	1.7214	13.5004	2.4425	**5.3972**	1.3233	5.6889	1.8210
	DTLZ2	**0.9472**	0.0128	0.9505	0.0096	$0.8766^\dagger$	0.2369	**0.7858**	0.2092
	DTLZ3	20.9175	4.1670	**20.8809**	3.6253	**5.5481**	1.5725	5.7157	1.9170
	DTLZ4	**0.9651**	0.0111	0.9698	0.0091	**1.8171**	0.4847	1.9608	0.5237
	WFG4	**0.5440**	0.0657	0.5824	0.0767	**4.3602**	1.2820	4.8900	1.2743
	WFG5	**0.2777**	0.0364	0.3515	0.0598	**6.9273**	1.9601	7.8535	2.5710
	WFG6	**0.2420**	0.0312	0.3176	0.0600	**7.7897**	1.9793	$9.1467^\dagger$	2.6586
	WFG7	**0.3314**	0.0527	0.4231	0.0675	**5.7006**	1.8121	6.0291	1.5542
	WFG8	$0.3600^\dagger$	0.0454	0.4634	0.0771	**6.8620**	1.9413	7.5327	2.3058
	WFG9	**0.2604**	0.0606	0.3212	0.0823	**8.1595**	2.7932	9.0655	2.6091

implementation, our pair of experiments differ only in the selection of the weighting vector. The constant weighting vector has been set to $\mathbf{w}_C = \frac{1}{k}\mathbf{1}$, where k is the number of objectives and $\mathbf{1}$ is a vector of ones of in $\mathbb{R}^k$. The reason for this selection is based primarily to preserve symmetry in the problem and to avoid potential truncation in the density used for the varying weighting vector, as is explained in what follows. However, an interesting point, which we have left for future studies, is that variation of the weighting vectors - even for symmetrical Pareto front geometries such as these employed in this work - could potentially have a more pronounced effect on the convergence rate of the algorithm. Nevertheless this hypothesis still remains untested.

The selection of the varying weighting vector, $\mathbf{w}_\mathcal{V}$, is more involved. For instance, if we vary directly the weighting vector, the comparison could be biased because the mapping from objective space to weighting vector space is nonlinear [12]. [1] Instead, we make the assumption that the algorithm can in fact produce a set of weighting vectors that cannot precisely target the point, T, but can generate weighting vectors to target points in its neighbourhood which is within a radius $\frac{h}{2}$ from the point T *Fig. 1*. Therefore, for every new sample, a point in the neighbourhood of T is generated with equal probability within the hypersphere defined by $(T, \frac{h}{2})$. This way, even for the varying weighting vector the target point still is T, although in this case this holds true in the mean.

Once a neighbouring point to T, say $\tilde{T}$, is selected, we must identify the weighting vector that would produce a subproblem which will tend to converge to $\tilde{T}$. To accomplish this we use *generalized decomposition*, which we introduced in [12]. Generalized decomposition solves the following problem: Given a point on the Pareto front, what is the weighting vector by which

[1] It should be noted however that methods exist that employ this scheme for perturbing their weighting vectors: see for instance [15], [16]. Nevertheless such methods are not commonly used by state-of-the-art optimization algorithms.

we can define a subproblem using a scalarizing function from the family of the weighted metrics (2), to obtain the given Pareto optimal solution. Although generalized decomposition can be applied to a much wider context, here, we employ it in the following form:

$$\begin{aligned} &\min_{\mathbf{w}} \|\mathbf{w} \circ \mathbf{F}(\mathbf{x})\|_\infty, \\ &\text{subject to } \sum_{i=1}^{k} w_i = 1, \\ &\text{and } w_i \geq 0, \forall\, i \in \{1, \ldots, k\}, \mathbf{F}(\mathbf{x}) \geq 0. \end{aligned} \tag{4}$$

It should be noticed that in (4), the optimization is with respect to $\mathbf{w}$. We also make the assumption that the ideal vector, $\mathbf{z}^\star = \mathbf{0}$, which holds for the selected problem set. For further details on generalized decomposition the reader is referred to [12], [21].

And therefore, the procedure that we employ to generate the varying weighting vector, $\mathbf{w}_\mathcal{V}$ can be summarised as follows:

- Sample a Pareto optimal point within $\frac{h}{2}$ distance of the target point T with equal probability.
- Use generalized decomposition to identify the weighting vector that will produce a subproblem that will converge to the sampled point, $\tilde{T}$.
- Use the obtained weighting vector, $\mathbf{w}_\mathcal{V}$, and the Chebyshev decomposition to assign a fitness to the new solution.

The above procedure is performed for every new decision vector produced by the random search algorithm described in Section III.

The results shown in Table I have been obtained by 50 independent trials. To test whether the obtained data can reasonably be represented by a normal distribution we have used the Lilliefors test for normality [24, pp. 532] with $\alpha = 0.01$ which corresponds to 99% confidence. The results in Table I that failed this test are marked with a dagger†. This means that the data are unlikely to originate from a normal distribution. The reason for this deviation is beyond the scope of this work, and is to be further investigated in a future work.

IV. Discussion

A glance at Table I illuminates the clear correlation between the use of a constant weighting vector and faster convergence. This behaviour is consistent for a number of test problems across a variety of objective-space dimensions. Additionally, although the mean target vector for both the constant weighting vector, $\mathbf{w}_C$, and the varying weighting vector, $\mathbf{w}_\mathcal{V}$, is the same, it can be clearly seen that there is a strong tendency for solutions to *wander* in the objective-space when the weighting vector is varying. These results are intuitive: convergence will be hindered when the decision problem to be solved is varying. Nevertheless, as there is a trend towards extending adaptive weighting vectors in decomposition-based methods, our findings show that care must be taken when considering such adaptive schemes. It is also worth noting that our results are consistent across dimensions of objective-space: meaning that the cost of adaptation is apparent for even three-objective problems.

It should also be noted that in this scenario we have made conservative assumptions that will be difficult to be achieved in a practical situation. For instance, there is no guarantee that the varying weighting vector will in fact converge to the mean of the target vector. Since this target vector, when extended to the entire Pareto front, represents the set of Pareto optimal solutions that are evenly spaced, this means that such adaptation bears the risk of not accomplishing its aim, which is to distribute solutions *well*.

In terms of previous results, for instance the work of Zhang and Li [8], where the authors compare their algorithm to a version of the multi-objective genetic local search algorithm (MOGLS) [15], the reason for its superior performance is clarified. This is because, if a *fair* variation of weighting vectors, as employed in this work has such a profound effect in terms of convergence, then, the selection employed in [15] where weighting vectors are varied across the entire simplex, will result in even worse performance as seen in [8]. This also means that a comparison of decomposition-based methods with fixed and varying weighting vectors is not entirely fair - as the latter attempt to solve a more complicated problem which in turn stalls their convergence rate.

Our current experiment is set up to compare, via simulation, fixed weighting vectors and adaptive weighting vectors in decomposition-based algorithms. An advantage of the framework is that it is readily extendible to simulate the operation of Pareto-based algorithms. In future work we will identify the distribution of solutions over the course of operation of a Pareto-based algorithm and then recast this distribution in terms of varying weighting vectors. We anticipate that these new experiments will further illuminate the source of difficulties in controlling the convergence-distribution meta-trade-off that Pareto-based algorithms face with decision problems featuring more than three conflicting objective functions. Alongside the experiments, we hope to more clearly formulate the theoretical linkages between Pareto-based algorithms and decomposition-based algorithms that employ varying weighting vectors. To see this, consider the aim of a posteriori optimization methods: that is, the relatively even distribution of solutions across the entire trade-off surface. To accomplish this, Pareto-based algorithms employ archiving and clustering methods that in a sense segment the Pareto front in even regions and only a certain number of solutions are allowed inside them. This imposes an implicit constraint on the density of Pareto-optimal solutions and this constraint could be represented by something very similar to the region defined by $(T, \frac{h}{2})$ in Section III-A. However, as Pareto-based methods do not have a constant weighting vector to clearly guide their search, solutions move randomly within the constraints of that region - as did the target solutions in our experiment. This hypothesis is further supported by the results in [11], where it is shown that the Chebyshev decomposition, that is more than often employed in decomposition-based algorithms, is equivalent to the Pareto dominance relationship in the sense that the probability of finding an improved solution is identical for both cases.

V. Conclusion

In this paper we have provided evidence to support the hypothesis that varying the weighting vectors in a decomposition-based algorithm will impose a non-negligible cost in terms of convergence, when compared to methods that employ fixed weighting vectors. We speculate that this cost is probably the distinguishing line between decomposition-based and Pareto-based algorithms. Specifically, the apparent poor performance of Pareto-based methods in optimization problems with more than three objectives is likely to be due to this cost rather the loss of selection pressure as suggested in [4], [6]. Further, our findings suggest that adaptive decomposition-based schemes are likely to replicate the known issues for Pareto-based algorithms.

At this point we feel that we should clarify that we are not advocating against adaptive decomposition-based algorithms. We simply suggest that there are numerous difficulties yet to be overcome for such algorithms to become viable – in the sense that they at least do not significantly hinder convergence to a solution that is already difficult to approach. Therefore we conclude that, for now at least, the new wave of decomposition-based methods offer no panacea to the familiar conflict between convergence and distribution that has proved the bane of Pareto-based methods over the previous decade.

Concluding, we believe that there is a need for a shift in perspective in the multi-objective optimization community. Namely, it is becoming increasingly obvious that there is a need for what we refer to as *optimization problem identification*. That is, the focus should shift from algorithmic frameworks towards broadening the class of problems that we can deal with efficiently and the development of strategies to address specific challenges. This will of course inevitably lead to the introduction of novel procedures, however the setting will be rather different. We envisage that such an approach would greatly enhance the ability of algorithms that are designed for nonconvex optimization problems, in a single or multi-objective context. This can potentially lessen the gap between industry and academia so eloquently exposed by Michaelowicz [25].

References

[1] K. Miettinen, *Nonlinear Multiobjective Optimization*. Springer, 1999, vol. 12.

[2] P. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 174–188, 2003.

[3] R. Purshouse and P. Fleming, "Evolutionary Many-Objective Optimisation: An Exploratory Analysis," in *IEEE Congress on Evolutionary Computation*, vol. 3. IEEE, 2003, pp. 2066–2073.

[4] ——, "On the Evolutionary Optimization of Many Conflicting Objectives," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 770 –784, dec. 2007.

[5] D. Hadka and P. Reed, "Diagnostic assessment of search controls and failure modes in many - objective evolutionary optimization," *Evolutionary Computation*, 2012.

[6] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary Many-Objective Optimization: A Short Review," in *IEEE Congress on Evolutionary Computation*, june 2008, pp. 2419 –2426.

[7] E. J. Hughes, "Evolutionary Many-Objective Optimisation: Many Once or One Many?" in *IEEE Congress on Evolutionary Computation*, vol. 1, 2005, pp. 222–227.

[8] Q. Zhang and H. Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[9] E. Hughes, "MSOPS-II: A General-Purpose Many-Objective Optimiser," in *IEEE Congress on Evolutionary Computation*, sept. 2007, pp. 3944 –3951.

[10] H. Ishibuchi, N. Akedo, H. Ohyanagi, and Y. Nojima, "Behavior of EMO Algorithms on Many-Objective Optimization Problems with Correlated Objectives," in *IEEE Congress on Evolutionary Computation*, june 2011, pp. 1465 –1472.

[11] I. Giagkiozis and P. Fleming, "Methods for many-objective optimization: An analysis," Research Report No. 1030, November 2012.

[12] I. Giagkiozis, R. Purshouse, and P. Fleming, "Generalized decomposition," in *Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, vol. 7811, pp. 428–442.

[13] Y. Jin, T. Okabe, and B. Sendho, "Adapting Weighted Aggregation for Multiobjective Evolution Strategies," in *Evolutionary Multi-Criterion Optimization*. Springer, 2001, pp. 96–110.

[14] Y. Jin, M. Olhofer, and B. Sendhoff, "Dynamic weighted aggregation for evolutionary multi-objective optimization: Why does it work and how?" in *Genetic and Evolutionary Computation Conference, pp.1042-1049, San Francisco*, 2001.

[15] A. Jaszkiewicz, "On the Performance of Multiple-Objective Genetic Local Search on the 0/1 Knapsack Problem - A comparative Experiment," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 4, pp. 402–412, 2002.

[16] M. Mahfouf, M. Chen, and D. Linkens, "Adaptive weighted particle swarm optimisation for multi-objective optimal design of alloy steels," in *Parallel problem solving from nature-ppsn viii*. Springer, 2004, pp. 762–771.

[17] S. Jiang, J. Zhang, and Y. Ong, "Asymmetric Pareto-adaptive Scheme for Multiobjective Optimization," in *Advances in Artificial Intelligence*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, vol. 7106, pp. 351–360.

[18] S. Jiang, Z. Cai, J. Zhang, and Y.-S. Ong, "Multiobjective Optimization by Decomposition with Pareto-Adaptive Weight Vectors," in *International Conference on Natural Computation*, vol. 3, july 2011, pp. 1260 –1264.

[19] R. C. Purshouse and P. J. Fleming, "Conflict, Harmony, and Independence: Relationships in Evolutionary Multi-Criterion Optimisation," in *Conference on Evolutionary Multi-Criterion Optimization*. Berlin: Springer, 2003, pp. 16–30.

[20] E. Hughes, "Multiple Single Objective Pareto Sampling," in *Congress on Evolutionary Computation, 2003*, vol. 4. IEEE, 2003, pp. 2678–2684.

[21] I. Giagkiozis, R. Purshouse, and P. Fleming, "Generalized Decomposition and Cross Entropy Methods for Many-Objective Optimization," Department of Automatic Control and Systems Engineering, The University of Sheffield, Research Report No. 1029, November 2012.

[22] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable Multi-Objective Optimization Test Problems," in *Congress on Evolutionary Computation*, vol. 1, may 2002, pp. 825 –830.

[23] S. Huband, P. Hingston, L. Barone, and L. While, "A Review of Multiobjective Test Problems and A Scalable Test Problem Toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006.

[24] M. Hollander and D. Wolfe, *Nonparametric Statistical Methods*. Wiley-Interscience, 1999.

[25] Z. Michalewicz, "Quo vadis, evolutionary computation?" *Advances in Computational Intelligence*, pp. 98–121, 2012.