A Scalability Study of Many-Objective Optimization Algorithms

Justin Maltese, *Member, IEEE*, Beatrice M. Ombuki-Berman, and Andries P. Engelbrecht, *Senior Member, IEEE*

Abstract—Over the past few decades, a plethora of computational intelligence algorithms designed to solve multi-objective problems have been proposed in the literature. Unfortunately, it has been shown that a large majority of these optimizers experience performance degradation when tasked with solving problems possessing more than three objectives, referred to as many-objective problems. The downfall of these optimizers is that simultaneously maintaining a uniformly-spread set of solutions along with appropriate selection pressure to converge towards the Pareto-optimal front becomes significantly difficult as the number of objectives increases. This difficulty is further compounded for large-scale many-objective problems, i.e., many-objective problems with a large number of decision variables. In this work, insight is given into the current state of many-objective research by investigating scalability of state-of-the-art algorithms using three to 15 objectives and 30 to 1000 decision variables. Results indicate that evolutionary optimizers are generally the best performers when the number of decision variables is low, but are outperformed by the swarm intelligence optimizers in several large-scale many-objective problem instances. However, a recently proposed evolutionary algorithm which combines dominance and subregion-based decomposition is shown to be promising for handling the immense search spaces encountered in large-scale many-objective problems.

Index Terms—many-objective optimization, large-scale optimization, computational intelligence, scalability, Pareto optimality.

I. INTRODUCTION

Many practical optimization problems contain multiple (often conflicting) goals to be optimized concurrently, commonly referred to as multi-objective problems (MOPs). A MOP can be formally defined as

$$\text{minimize } \mathbf{F}(\vec{x})$$
$$\text{subject to } \vec{x} \in \Omega$$

where $\vec{x} = (x_1, x_2, ..., x_{n_x})$ is a candidate solution, Ω is the decision variable space, $\mathbf{F}(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), ..., f_{n_c}(\vec{x}))$, and n_c is the number of objective functions. Due to the prevalence of conflicting objectives it is rare for a single solution to optimize all objectives simultaneously, thus one must accept a set of trade-offs when solving MOPs. The goal of a multi-objective optimizer (MOO) is to find the Pareto optimal front, formally given as

$$PF^* = \{\vec{y}^* \in \mathbb{R}^{n_c} | \nexists \vec{y} \in R^m : \vec{y}^* \prec \vec{y}\} \qquad (1)$$

J. Maltese and B. M. Ombuki-Berman are with the Department of Computer Science at Brock University, St. Catharines, Canada.
A. P Engelbrecht is with the Department of Computer Science at the University of Pretoria, South Africa.

where $\mathbb{R}^{n_c}$ represents the objective space and $\prec$ is a strict dominance relation such that $\vec{y}^* \prec \vec{y}$ iff $\vec{y}_i^* \leq \vec{y}_i$ for all i and $\vec{y}_i < \vec{y}_i^*$ for some i, where $i \in \{1, ..., n_c\}$.

MOPs are commonly encountered in fields such as engineering [1], business [2] , mathematics [3] and physics [4]. The abundance of practical MOPs presents a real need for effective MOOs, thus unsurprisingly a large amount of effort has been devoted to proposing novel computational intelligence MOOs (CIMOOs). A small sample of evolutionary multi-objective optimization (EMO) can be viewed in [5], [6], [7], [8] and [9]. Several examples of swarm intelligence MOOs can be seen in [10], [11], [12], [13] and [14].

A large majority of previously proposed CIMOOs evaluate their performance on MOPs possessing two or three objectives, often producing a set of well converged and spread solutions. Unfortunately, a large majority of the classic CIMOOs have been shown to degrade when the number of objectives are increased, especially those which perform optimization using the Pareto-dominance relation [15], [16]. Problems possessing more than three objectives, commonly referred to as many-objective problems (MaOPs), present a serious challenge for CIMOOs. Possessing the ability to efficiently solve MaOPs is highly desirable for CIMOOs, since many real-world applications such as industrial scheduling [17], [18], automotive engine calibration problems [19] and hybrid car controller optimization [20] have more than three objectives.

The poor scalability of CIMOOs is due to a variety of challenges unique to MaOPs. The first and arguably most significant issue experienced when attempting to solve MaOPs is the difficulty in balancing convergence and diversity [16]. Since the end goal of any MOO is to obtain an approximation of the Pareto-optimal front, selection pressure to converge, along with a focus on solution spread, must be concurrently maintained throughout optimization. Unfortunately, prioritizing both convergence and diversity becomes increasingly more difficult as the number of objectives grows, primarily as a result of the large search space sizes encountered in MaOPs. Consequently, most CIMOOs end up sacrificing convergence for solution spread, or vice-versa.

The end result of many-objective optimization is thus either a solution set which is nearly (or exactly) uniformly spread but is undesirably far from Pareto-optimality, or a solution set which has converged to a small subregion near or on the Pareto-optimal front. In the case of traditional Pareto-based optimizers, the former is often the case. The main reason for this phenomenon is a loss of selection pressure towards the Pareto-optimal front. As the number of objectives grows, the Pareto-dominance relation essentially loses the ability to

distinguish desirable solutions, since nearly all population members are non-dominated at an early stage of the search [21]. In fact, over 90% of a randomly generated initial solution set is non-dominated when the number of objectives are eight or more [22]. Therefore, utilizing only the Pareto-dominance relation as selection criterion is nearly a random search, likely guiding a CIMOO into sub-optimal areas of the search space.

Maintenance of a uniformly spread solution set becomes increasingly difficult for MaOPs due to several reasons. Traditionally, Pareto-based algorithms employ the fast, computationally efficient crowding distance [5] operator, which yields satisfactory performance on two and three objective MOPs. Unfortunately, the effectiveness of the crowding distance operator degrades considerably for MaOPs [23], largely due to the overprioritization of dominance-resistant solutions [15], i.e., solutions with exceptional performance in one objective and extremely poor performance in many others.

Since dominance-resistant solutions have a high chance of remaining non-dominated by definition, CIMOOs are essentially mislead by them. One should note that while dominance-resistant solutions worsen convergence, they should not be entirely eliminated as preserving a reasonable number of them may benefit population diversity [15]. Therefore, utilizing a diversity-preservation mechanism that maintains some dominance-resistant solutions, but does not overprioritize them, is key in maintaining a uniformly spread set of solutions for MaOPs. Another non-trivial issue regarding solution spread maintenance for MaOPs is the high computational complexity cost of accurately gauging the crowdedness of solutions. While this can be made faster by estimation methods, spread of the final approximated front may be impacted considerably.

The second major difficulty encountered in MaOPs is the extremely large objective space resulting from an increase in objectives. Consequently, it is highly probable that many solutions found throughout the search are likely to be distant from each other [24]. This has consequences for evolutionary multi-objective optimization (EMO), since when two distant parents are mated the offspring is likely to be far from both parents. Therefore, it is necessary to employ some form of modification to recombination operators which ensures that offspring are reasonably close to parents. Without any modification, evolutionary algorithms (EAs) may experience reduced effectiveness of recombination operators, leading to difficulties during the search process.

Another issue for MaOPs is the high level of computational complexity required to measure algorithm performance. For many performance metrics, evaluating an approximation front produced by a CIMOO often requires a large number of high-dimensional points. Consequently, some metrics become practically infeasible to calculate over a certain number of objectives, such as the hypervolume metric [7]. Since calculating exact hypervolume is exponential with respect to the number of objectives [25], hypervolume values are often estimated using a Monte Carlo sampling technique [26] for MaOPs possessing more than 10 objectives. This is problematic since estimated values produce some degree of inaccuracy when comparing CIMOOs.

The final difficulty encountered for MaOPs relates to selection of a final solution from an approximated front produced by a CIMOO. Visualization of the high-dimensional trade-off surfaces is challenging. This issue is also compounded with the fact that an exponentially increasing number of points are required to accurately represent higher-dimensional trade-off surfaces. CIMOOs thus require larger numbers of candidate solutions for MaOPs, increasing the level of difficulty for a decision maker regarding the assessment and selection of a final solution from an approximated front. Although this difficulty is not related to optimization directly, it presents significant obstacles for practical many-objective optimization.

The aforementioned difficulties of MaOPs have presented a need for CIMOOs specifically designed to scale well when tasked with an increasing number of objectives. Unsurprisingly, recent years have seen a dramatic rise in the number of proposed many-objective CI algorithms. A comprehensive survey on evolutionary many-objective optimization is presented in [27]. The large majority of proposed many-objective optimizers focus exclusively on objective scalability, ignoring decision variable scalability almost entirely by benchmarking on MaOPs possessing a relatively low number of decision variables, often less than 40. However, since employing more decision variables will further increase the already large search space of MaOPs, it is plausible that some many-objective algorithms will scale poorly to large-scale MaOPs, i.e., MaOPs possessing large numbers of decision variables. Previous work by Durillo et. al [28] has investigated the performance of CIMOOs on large-scale MOPs. However, comparisons were limited to two and three objective problems.

In this work, the scalability of many-objective optimizers is explored by analyzing performance of state-of-the-art CIMOOs using an increasing number of decision variables and objectives. To the best of the authors' knowledge, no existing literature compares many-objective algorithms on large-scale problems, especially using algorithms from both CI subfields of swarm intelligence and evolutionary computation. Our intent is to help fill this literature gap, providing insight into the performance of CIMOOs when tasked with large-scale many-objective optimization.

The remainder of this paper is organized as follows: Section II provides a review of previous literature on many-objective optimization. Section III gives implementation-level details of each selected CIMOO used for testing in this work. Section IV describes the experimental setup used in this work is described. Section V presents the results of all experiments performed, including an analysis and a discussion of the observations. Finally, Section VI concludes the paper and suggests avenues for future research.

II. PREVIOUS LITERATURE

This section provides a review of previous literature regarding many-objective optimization. Five distinct categories of techniques used to solve MaOPs are given, along with previously proposed algorithms which fall under each category.

For MaOPs containing redundant objectives, dimensionality reduction techniques can be employed to shrink the number

of objectives. Problems for which the number of objectives are reducible to three or less can be solved using classic CIMOOs such as [5], [6], whose efficiency has already been proven on two and three objective MOPs [29]. An example of this approach can be seen in [30], where principal component analysis is employed to remove redundant objectives.

MaOPs without a sufficient number of redundant objectives present a considerably more difficult challenge. A number of many-objective techniques have been proposed in previous literature, which can largely be separated into five unique categories. The first category of methods is to simply modify the dominance relation in a way that alleviates the lost selection pressure to converge to the Pareto-optimal front. Concepts which fall under this category are ϵ-dominance [31], controlling dominance area of solutions [32], [33], fuzzy-based Pareto optimality [22], k-optimality [34] and preference ordering [35].

Another possible approach is to employ an additional convergence-related metric as secondary selection criteria alongside the traditional Pareto dominance relation. In this category, the original dominance relation is typically left intact and some form of convergence criterion is used to provide selection pressure on the non-dominated solutions, determining which of them are converging best towards the Pareto-optimal front. Special care must be taken to preserve solution spread, thus diversity is either maintained indirectly by the convergence metric or through a separate tertiary operator which gauges the crowdedness of solutions. Examples of previously proposed convergence metrics are knee points [36], grid-based fitness [37] and sub-objective dominance count [38].

The third category of methods is to integrate performance metrics during optimization, helping to distinguish desirable solutions. Here, a performance indicator which ideally measures both spread and convergence is chosen to guide the search, selecting individuals which contribute best to more desirable values for the indicator. Since the computational cost of most useful performance indicators (e.g., hypervolume) grows with the number of objectives, many indicator-based optimizers possess a considerably high level of computational complexity for MaOPs. While recent work [26], [39], [40] has successfully reduced the computational complexity of some indicators, the issue is still far from remedied. The indicator-based evolutionary algorithm [41] and hypervolume estimation algorithm (HypE) [26] are two such examples of indicator-based approaches.

The fourth category of methods to solve MaOPs is the decomposition method. Decomposition-based optimizers essentially extract a set of subproblems from a given MOP, optimizing each of them simultaneously. A comprehensive survey on decomposition-based EAs is presented in [42]. The main representative of this category is the multi-objective evolutionary algorithm using decomposition (MOEA/D) [6], which utilizes a user-defined aggregation function to perform MOP decomposition. MOEA/D has shown decomposition to be a promising avenue for solving MaOPs, as it is a reasonably fast method which performs quite well on MaOPs [24]. However, Isibuchi et. al [43] have shown that the performance of decomposition-based optimizers is highly sensitive to Pareto front shapes. Recently, three promising methods combining dominance and decomposition have been proposed, namely MOEA/DD [44], θ-DEA [45] and non-dominated sorting genetic algorithm III (NSGA-III) [24].

Abandoning the Pareto-dominance relation entirely is another viable possibility for scaling CIMOOs to MaOPs, forming the final category of methods. New relations would be used to determine the desirability of each solution, ideally incorporating a blend of convergence and diversity. If the relation favors convergence too much, diversity may be lost and the search may converge to a small subregion near or on the Pareto-optimal front. Conversely, if the relation overvalues diversity it will encounter the same convergence issues of traditional Pareto dominance, losing selection pressure to converge to the Pareto-optimal front. Conformance to the original Pareto-dominance relation is also a desirable property, i.e., the new relation always considers non-dominated solutions as more desirable than non-dominated solutions. Approaches which fall under this category are F_{sum} [46], F_{min} [46], normalized F_{sum} [47], [48], sum-of-ratios [49] and L-dominance [50].

III. SELECTED ALGORITHMS

For the purposes of this research, state-of-the-art algorithms are selected from each many-objective technique category described in Section II. Each category along with a description of the algorithm chosen to represent it is given within this section. A summary of all selected algorithms is given in Table I.

A. Dominance relation modification

The speed-constrained multi-objective particle swarm optimization (SMPSO) algorithm, proposed by Nebro *et. al* [10], is a multi-objective particle swarm optimization (PSO) variant with two defining features. The first defining feature is an external archive which stores non-dominated solutions, where the solution with the worst crowding distance [5] is removed when the maximum size has been exceeded. The second defining feature is a unique velocity constriction mechanism, which is performed by multiplying particle velocity using a small factor in the event where a particle has exited the feasible region. SMPSO also applies polynomial mutation [51] to 15% of the particles in the swarm, selected randomly.

A many-objective extension of the original SMPSO algorithm exists which modifies the traditional dominance relation using the controlling dominance area of solutions (CDAS) technique, referred to as CDAS-SMPSO [52]. The CDAS method alters the area dominated by a solution through adjustment of the angles that meet the axis bounding this area. CDAS employs a vector $\vec{S}$ to modify the dominance area, where S_i corresponds to the dominance area control factor of objective i. Note that CDAS shrinks the dominance area of objective i if $S_i < 0.5$, grows the area if $S_i > 0.5$, and is equivalent to the traditional Pareto-dominance relation if $S_i = 0.5$. The objective vector $\bar{\mathbf{F}}$ of a solution $\vec{x}$ is transformed into $\mathbf{F}'$ by CDAS using the equation:

TABLE I
OVERVIEW OF SELECTED ALGORITHMS

Algorithm	Category	Reference	Year
CDAS-SMPSO	C1	de Carvalho and Pozo [52]	2012
KnEA	C2	Zhang *et. al* [36]	2014
KnPSO	C2	Maltese *et. al* [53]	2016
HypE	C3	Bader and Zitzler [26]	2011
MOEA/D	C4	Zhang and Li [6]	2007
MOEA/DD	C4	Li *et. al* [44]	2015
NSGA-III	C4	Deb and Jain [24]	2014
dMOPSO	C4	Martínez and Coello [54]	2011
SrEA	C5	Kukkonen and Lampinen [46]	2007

C1: Dominance relation modification
C2: Secondary convergence metric
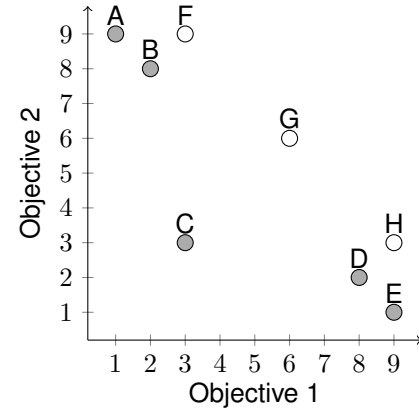C3: Indicator-based, C4: Decomposition-based, C5: Non-Pareto relation

Fig. 1. Eight solutions are shown for a bi-objective MOP. Each non-dominated solution is displayed with a gray fill. Here, solution C is a knee point due to its large marginal rates of return in comparison to the other non-dominated solutions.

$$\mathbf{F'_i}(\vec{x}) = \frac{r(\vec{x}) \cdot sin(\lambda_i(\vec{x}) + S_i \pi)}{sin(S_i \pi)} \qquad (2)$$

$$\text{where } r(\vec{x}) = ||f(\vec{x})|| \qquad (3)$$

$$\text{and } \lambda_i = arccos(\frac{\mathbf{F_i}(\vec{x})}{r(\vec{x})}) \qquad (4)$$

B. Secondary convergence metric

Knee points are defined as Pareto front solutions for which an improvement in any objective will severely degrade one or more objectives. Knee points correspond to the naturally preferrable points within the Pareto optimal front when no problem-specific knowledge is available due to their maximal marginal rates of return. In the absence of a decision maker each objective is treated as equally important, thus minimally improving an objective while immensely degrading others is undesirable. Figure 1 exemplifies a knee point for a MOP possessing two objectives. Within Figure 1, Solutions A to E form the non-dominated front, each displayed with a gray fill. Note that solution C is a knee point, as it exhibits larger marginal rates of return than the other non-dominated solutions.

Two many-objective optimizers incorporating knee points are selected for comparison in this work, namely the knee-driven evolutionary algorithm (KnEA) [36] and knee-driven PSO (KnPSO) algorithm [53]. KnEA and KnPSO employ the NSGA-II [5] and SMPSO [10] algorithm frameworks, respectively, utilizing the concept of knee points as an additional convergence metric alongside the traditional dominance relation.

Converging knee points within the current front are given archival priority in an elitist fashion, while non-crowded knee points are given preference when selecting solutions to guide the search. It is important to realize that solutions identified as knee points during earlier stages of the search are often not true knee points of the Pareto-optimal front, since the current front is simply an approximation. Rather, knee points of the current front represent solutions that are converging best within their immediate neighbourhood and are therefore useful

for increasing the selection pressure to converge towards the Pareto-optimal front. Note that prioritizing knee points also biases the search towards a higher hypervolume metric value [36].

Both KnEA and KnPSO utilize an adaptive strategy to determine knee points, initially proposed by Zhang *et. al* in [36]. This strategy is as follows: First, construct a hyperplane H which passes through the extremal points of each objective. For each candidate solution s, s is deemed a knee point if, and only if, s possesses the maximum objective space distance to H within its neighbourhood. The neighbourhood of a solution is a hypercube with n_c sides, where each side is calculated as

$$R_t^j = (max_t^j - min_t^j) \cdot r_t \qquad (5)$$

where max_t^j and min_t^j denote the maximal and minimal values of the j-th objective at time t respectively and r_t corresponds to the ratio of the neighbourhood size to the range spanned by objective j at time t. r_t is updated using

$$r_t = r_{t-1} \cdot e^{-\frac{1-(\alpha_{t-1}/T)}{n_c}} \qquad (6)$$

where α_{t-1} denotes the ratio of knee points to non-dominated solutions at time $t-1$ and T is a user-defined parameter which represents the desired ratio of knee points to non-dominated solutions, with $0 < T < 1$. Note that initially $\alpha_t = 0$ and $r_t = 1$. The strategy above essentially shrinks and grows the neighbourhood size adaptively until the ratio of knee points to non-dominated solutions in the solution set converges to T.

C. Performance indicator

The HypE algorithm [26] performs optimization based on the popular hypervolume indicator, which measures both convergence to the Pareto-optimal front and solution spread in a single scalar value. However, since calculation of hypervolume is extremely expensive for MaOPs, approximate hypervolume values are generated when the number of objectives $n_c > 3$. The hypervolume-based fitness function f^H central to HypE is defined as

$$f^H(s) = \begin{cases} actual\_hypervolume(s), & \text{if } n_c \leq 3. \\ estimate\_hypervolume(s), & \text{otherwise} \end{cases} \quad (7)$$

where $estimate\_hypervolume$ estimates the hypervolume contribution of solution s using Monte Carlo sampling [55] and $actual\_hypervolume$ calculates the exact hypervolume contribution of s using a given hypervolume calculation method, e.g., [39].

D. Decomposition

Since recent studies have shown decomposition-based optimizers to be very promising for MaOPs [44], [45], [24], several algorithms are selected from this category. Each decomposition approach utilizes a user-defined aggregation function to decompose a MOP into a set of subproblems, since intuitively, a Pareto-optimal solution to a MOP is an optimal solution of a scalar optimization problem corresponding to an aggregation of all objectives. All decomposition-based algorithms in this work employ the penalty-based boundary intersection (PBI) [6] aggregation method due to its promising performance for many-objective optimization [24], defined formally as

$$\text{minimize } g^{pbi}(\vec{x}|\vec{w}, \vec{z}^*) = d_1 + \theta d_2 \quad (8)$$

$$\text{where } d_1 = \frac{\left\| (\mathbf{F}(\vec{x}) - \vec{z}^*)^T \vec{w} \right\|}{\|\vec{w}\|} \quad (9)$$

$$d_2 = \left\| \mathbf{F}(\vec{x}) - (\vec{z}^* + d_1 \frac{\vec{w}}{\|\vec{w}\|}) \right\| \quad (10)$$

where $\vec{x} \in \Omega$, $\vec{w}$ is a weight vector, $\vec{z}^* = (z_1^*, ..., z_{n_c}^*)^T$ corresponds to the ideal point such that $z_i^* = min\{f_i(\vec{x})\}$ for all $\vec{x}$. By modifying the choice of $\vec{w}$, different Pareto optimal points can be found. Note that the recently proposed localized-weighted sum (LWS) method [56] is also a viable aggregation function for many-objective optimization, as it has exhibited promising performance on a variety of MaOPs. The LWS technique selects the best solution from among neighbouring solutions for each search direction, where a neighbourhood is defined using a hypercone, allowing efficient handling of non-convex problems.

The first decomposition-based optimizer selected is the multi-objective evolutionary algorithm using decomposition (MOEA/D) algorithm [6]. To maintain a diverse set of K solutions, MOEA/D maintains uniformly spread weight vectors $\{\vec{w}_0, ..., \vec{w}_K\}$, where solution i solves the subproblem corresponding to $\vec{w}_i$, $i \in \{1, ..., K\}$. For each $\vec{w}_i$, mating selection is performed from the T closest weight vectors to $\vec{w}_i$, since a central idea in MOEA/D is that information about subproblem i should be helpful for solving subproblem j if $\vec{w}_i$ and $\vec{w}_j$ are similar. Local replacement is also employed, where a generated offspring p replaces solutions among its neighbouring weight vectors if p possesses a better aggregation function value. Note that population diversity benefits from the inclusion of a small chance to conduct mating and replacement from the entire population [57]. To preserve non-dominated solutions, MOEA/D employs an external archiving structure.

The next decomposition approach is the decomposition-based multi-objective PSO (dMOPSO) [54] algorithm. dMOPSO is similar to MOEA/D in that a set of weights are required, where each individual optimizes the subproblem corresponding to its given weight. The personal best $\vec{y}_i$ of particle i is the best position ever position experienced by particle i with respect to the subproblem corresponding to $\vec{w}_i$. To select neighbourhood bests, an archive N_{bests} is maintained, where $N_{bests}(i)$ is the best ever position found regarding the subproblem of $\vec{w}_i$. The neighbourhood best update phase begins by shuffling the N_{bests} archive, since dMOPSO assumes that all solutions in N_{bests} are equally good. Thereafter, the neighbourhood best $\vec{\hat{y}}_i$ for particle i is simply set as $N_{bests}(i)$.

A unique characteristic of dMOPSO is that each particle maintains an age value, incremented whenever a particle does not update its personal best. In the case that a particle ages past a predefined age threshold parameter T_a, its position is re-initialized using a parametric probability density function.

The third algorithm chosen from the decomposition category is the recently proposed non-dominated sorting genetic algorithm III (NSGA-III) algorithm [24]. The basic flow of NSGA-III is identical to its predecessor, NSGA-II [5], however, diversity preservation is significantly different. At each generation, both NSGA-II and NSGA-III perform non-dominated sorting to partition the combined parent and offspring population into a set of fronts $F_0, F_1, ..., F_K$. Each front is then selected in ascending order, starting from F_0, to form the new population P_{t+1}.

In most cases the final selected front, denoted F_L, will only fit partially into P_{t+1}. In this situation, some form of diversity score is calculated for each solution in F_L to determine which solutions should be inserted into P_{t+1}. It is here that NSGA-III differs from NSGA-II, as NSGA-III computes the diversity of each solution using a set of reference points rather than the traditional crowding distance operator of NSGA-II, which has been shown to perform poorly for MaOPs [23]. Reference points can either be supplied by a user or generated in a structured manner. The approach in [24] utilizes Das and Dennis's boundary intersection approach [58], which places points on a normalized hyperplane possessing an intercept of one on each axis.

The final selected algorithm from the decomposition category is the multi-objective evolutionary algorithm using dominance and decomposition (MOEA/DD), proposed by Li *et. al* in [44]. MOEA/DD combines the strengths of the well-known NSGA-II [5] and MOEA/D [6] algorithms, which have been shown to excel at different problems [29], to form a promising steady-state evolutionary many-objective optimizer. A set of weight vectors $\{\vec{w}_1, ..., \vec{w}_K\}$ which each define a subproblem are maintained as in MOEA/D. However, each weight also corresponds to a subregion which can be used for region density estimation purposes. Specifically, subregion ψ^i is defined as

$$\psi^i = \{\vec{v} \in \mathbb{R}^{n_c} | \forall j \in \{1, ..., K\} : \langle \vec{v}, \vec{w}_i \rangle \leq \langle \vec{v}, \vec{w}_j \rangle \} \quad (11)$$

where $i \in \{1, .., K\}$, $x \in \Omega$ and $\langle \vec{v}, \vec{w}_i \rangle$ corresponds to the acute angle between $\vec{v}$ and $\vec{w}_i$. Note that to generate weight vectors, MOEA/DD employs Das and Dennis's boundary intersection method [58] to sample a layer of $\binom{D+n_c-1}{n_c-1}$ uniformly spread points from a unit simplex, where D is the desired number of divisions. In the case where $n_c \geq 7$, both a boundary layer and inner layer are employed to reduce the total number of generated reference points. After initializing the population randomly and performing weight vector generation, MOEA/DD performs the following steps for each weight vector $\vec{w}_i$ until some stopping criterion is satisfied:

Step 1) Mating Selection: Choose a set of parents Q for mating purposes randomly from the closest T subregions to $\vec{w}_i$. Note that to benefit population diversity, a small probability (1-δ) to select from the entire population is included. In the case where no solutions exist in the closest T subregions, Q is selected randomly from the entire population by default.

Step 2) Variation: Apply recombination operators to Q, producing a set of offspring R. The original work in [44] employs the simulated binary crossover [59] and polynomial mutation [60] operators.

Step 3) Population Update: For each solution $\vec{x} \in R$, add $\vec{x}$ to P and update the non-domination structure of P according to [61]. If after inserting $\vec{x}$ all solutions in P are non-dominated, simply remove the worst solution according to Algorithm 1. Else, if some solutions in P are dominated, P can be divided into fronts $F_0, ..., F_L$ and the following distinct cases arise:

1) *F_L has one solution $\vec{x}_j$ residing in subregion ψ^j and multiple solutions exist in ψ^j*: Here, simply remove $\vec{x}_j$ from P.

2) *F_L has one solution $\vec{x}_j$ residing in subregion ψ^j and $\vec{x}_j$ is the only solution in ψ^j*: In this case, ψ^j is an isolated subregion and thus $\vec{x}_j$ should be preserved to maintain solution spread. Remove from P the worst solution according to Algorithm 1.

3) *F_L contains multiple solutions*: In this case, identify the subregion ψ^k which contains the largest number of solutions in F_L. If $|\psi^k| = 1$, then ψ^k is an isolated subregion and diversity is preserved by simply removing from P the worst solution according to Algorithm 1. Otherwise, $|\psi^k| > 1$ and thus ψ^k is not an isolated subregion. In this case, remove from P the solution in ψ^k possessing the worst PBI value with respect to subproblem k.

After removing a solution from P, the non-domination structure is updated according to the efficient method in [61].

E. Non-Pareto Relation

A simple alternative to the traditional Pareto-dominance relation is the F_{sum} method [46], which sums together the rank of each individual objective of $\vec{x}$. The F_{sum} method is formally defined as

$$F_{sum}(\vec{x}) = \sum_{i=1}^{m} rank(i, f_i(\vec{x})) \tag{12}$$

Algorithm 1 Retrieving the Worst Solution

1: Get the most crowded subregion ψ_k in P. If two subregions are equally crowded, take the subregion with the largest sum of PBI values.
2: $S = \emptyset$
3: **for each** $\vec{u} \in \psi^k$ **do**
4:     **if** $\vec{u} \in F_L$ **then**
5:         $S = S \cup \vec{u}$
6:     **end if**
7: **end for**
8: Identify $\vec{y}$ as the solution in S possessing the largest PBI value with respect to the subproblem corresponding to $\vec{w}_k$.
    return $\vec{y}$

---

where $\vec{x}$ is a feasible decision vector and $rank$ is a function which takes an objective index i and objective value $f_i(\vec{x})$ as input, returning the rank of $f_i(\vec{x})$ when compared to all other objective vectors in the current front using objective i. The process used to calculate F_{sum} is illustrated in Figure 2.

Since F_{sum} prioritizes convergence, diversity must be maintained using an external metric. Unfortunately, this is non-trivial, since solutions are already ordered according to the chosen aggregation method. One method of preserving diversity as proposed in [46] is to gradually increase in a linear fashion the number of solutions selected according to a diversity metric. Kukkonen and Lampinen [46] incorporated this method into the NSGA-II algorithm [5], but did not give a formal name for the resultant algorithm. This work refers to this method as the sum-of-ranks evolutionary algorithm (SrEA), which linearly increases the number of solutions selected according to diversity such that:

- At generation 0, selection from F_L for P_{t+1} and selection of the parent population Q is done according to the ordering imposed by the F_{sum} method.
- Let G_{max} denote the maximum number of generations. At generation $G_{max}/2$, half of the remaining solutions for P_t+1 are selected from F_L based on the F_{sum} ordering and half are selected from F_L according to a diversity-preservation operator. Selection of Q is also performed in this manner.
- At generation G_{max}, selection from F_L for P_{t+1} and selection for Q is performed using only a diversity-preservation operator.

The motivation behind this method is to promote convergence early on and to prioritize diversity in later stages of the search when the population has already converged. Within this work, the diversity-preservation metric employed in SrEA is the weighted distance metric [36].

IV. EXPERIMENTAL SETUP

This section describes the experimental setup employed within this study. Topics covered include benchmark functions, performance indicators, statistical analysis methodology and algorithm parameters.
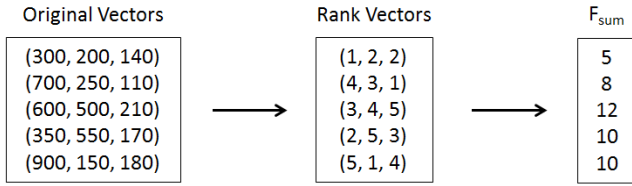
Original Vectors	Rank Vectors	F_{sum}
(300, 200, 140)	(1, 2, 2)	5
(700, 250, 110)	(4, 3, 1)	8
(600, 500, 210)	(3, 4, 5)	12
(350, 550, 170)	(2, 5, 3)	10
(900, 150, 180)	(5, 1, 4)	10

Fig. 2. Calculation of the F_{sum} aggregation method is demonstrated for a front containing five solutions. The MOP in question possesses three objectives.

A. Benchmark Functions

Four problems from the Deb-Thiele-Laumanns-Zitzler (DTLZ) suite [62] and nine problems from the Walking Fish Group (WFG) suite [63] are used in this study. Collectively, the set of chosen functions present a variety of shapes and modalities which simulate practical environments and present a challenging task for optimizers. Note that both the number of objectives and decision variables are modifiable for each selected function, useful for observing the scalability of each selected CIMOO. Characteristics of all selected benchmark functions are summarized in Table II.

B. Statistical Methodology

Pairwise Mann-Whitney-Wilcoxon rank sum tests [64] at a confidence level of 95% were used to test for significant performance differences between algorithms. For each pairwise test, if a statistically significant difference existed, the algorithm with the higher mean over 20 independent runs was given a win and the algorithm with the lower mean was given a loss. The term *difference* is used to distinguish the performance of algorithms, which is simply the difference between pairwise wins and losses for a given algorithm. Additionally, the *rank* of each algorithm denotes the ranking in comparison to all other algorithms with respect to difference values for a given benchmark function.

C. Algorithm Parameters

All algorithms used an identical number of candidate solutions and function evaluations to ensure unbiased comparisons as recommended in [65]. The number of candidate solutions used is dependent on the number of reference points employed for the decomposition-based algorithms. To generate uniformly spaced reference points, a unit simplex is sampled from using Das and Dennis's boundary intersection method [58]. For problems where $n_c > 5$, the method in [24] is used to reduce the number of generated reference points using parameters $D1$ and $D2$, which are the number of divisions of the boundary layer of reference points and inner layer of reference points, respectively. The number of generated reference points for varying n_c values are listed in Table III. The number of candidate solutions used for all algorithms is identical to the number of reference points generated. Note that the relatively small number of reference points (135) for $n_c = 15$ is a consequence of the extremely high computational complexity required for large-scale MaOPs. Ideally a larger number of reference points and candidate solutions would

be used to measure performance. However, when $D_2 > 1$ computational complexity is drastically increased such that experimental runs are no longer feasible to complete in a reasonable amount of time.

The number of function evaluations used for each algorithm was dependent on the problem complexity and number of objectives, and are displayed in Table IV. All algorithms requiring non-dominated sorting used the recently proposed efficient non-dominated sorting (ENS) [66] algorithm. Each decomposition-based approach used the PBI function (see Section III.D) with $\theta = 5.0$ as recommended in [6]. Since NSGA-III utilizes a normalization procedure [24], the other decomposition-based algorithms, i.e., MOEA/D, MOEA/DD, dMOPSO, are also normalized using an identical procedure to ensure that the range of each objective is identical. Concerning the T parameter of the KnPSO and KnEA algorithms, many values were tested for each experiment and the setting which produced the best IGD values on average over 10 runs was chosen. The T values used are given in Table V.

1) Genetic Algorithms: All evolutionary algorithms within this work used the simulated binary crossover [59] with a distribution index of 30 and polynomial mutation [60] with a distribution index of 20. Crossover probability was set to 1.0 and mutation probability was set to $1/n_x$. The weighted distance metric employed by SrGA and KnEA used $k = 5$ nearest neighbours. A value of 20 was used for the neighbourhood size T of the MOEA/D and MOEA/DD algorithms. Both MOEA/D and MOEA/DD used a neighbourhood selection probability of $\delta = 0.9$, identical to the value recommended in [44], [57]. The number of points used for Monte Carlo sampling in the HypE algorithm was set to 10,000 as recommended in [26].

2) PSO Algorithms: CDAS-SMPSO and KnPSO each utilized a maximum archive size identical to the number of reference points generated (see Table IV). Polynomial mutation [60] was used with a probability of $1/n_x$, identical to the evolutionary algorithms. The age threshold T_a of dMOPSO was set to 2, as recommended in the original study [54]. For the CDAS-SMPSO and KnPSO algorithms, at each iteration c_1 and c_2 varied randomly in the range [1.5,2.5] while ω varied randomly in the interval [0,0.8]. To handle boundary constraints, personal and neighbourhood bests were only updated if their positions remained within the legal bounds of the search space. Initial particle velocity was set to zero to comply with the recommendations given in [67].

With regards to the sensitive S value of CDAS-SMPSO, many values were tested for each experiment and the setting which produced the best IGD value on average over 10 runs was chosen. The exact S values used are shown in Table VI.

D. Performance Indicators

Hypervolume: The hypervolume metric, introduced in [7], indicates both the convergence and spread of a MOO in a single scalar value by measuring the amount of space covered by a given solution set. A desirable property of the hypervolume metric is that it is maximized if and only if the solution set consists of equidistant Pareto-optimal points [68]. Additionally, the hypervolume metric can be used to show

TABLE II
OVERVIEW OF BENCHMARK FUNCTIONS USED

Function	Separability	Bias	Shape	Modality
DTLZ1	Separable	No	Linear	Multi
DTLZ2	Separable	No	Concave, Disconnected	Uni
DTLZ3	Separable	No	Concave	Multi
DTLZ4	Separable	Yes	Concave	Uni
WFG1	Separable	Yes	Convex	Uni
WFG2	Non-Separable	No	Convex, Disconnected	Multi
WFG3	Non-Separable	No	Linear	Uni
WFG4	Separable	No	Concave	Multi
WFG5	Separable	No	Concave	Deceptive
WFG6	Non-Separable	No	Concave	Uni
WFG7	Separable	Yes	Concave	Uni
WFG8	Non-Separable	Yes	Concave	Uni
WFG9	Non-Separable	Yes	Concave	Multi, Deceptive

TABLE III
NUMBER OF REFERENCE POINTS EMPLOYED. NOTE THAT THE NUMBER
OF CANDIDATE SOLUTIONS IS IDENTICAL TO THE NUMBER OF REFERENCE
POINTS.

n_c	D_1	D_2	# of reference points
3	12	0	91
5	6	0	210
8	3	2	156
10	3	2	275
15	2	1	135

TABLE IV
NUMBER OF FUNCTION EVALUATIONS USED

Benchmark Function	$n_c = 3$	$n_c = 5$	$n_c = 8$	$n_c = 10$	$n_c = 15$
DTLZ1	30,000	50,000	60,000	85,000	100,000
DTLZ2	25,000	45,000	55,000	80,000	100,000
DTLZ3	75,000	75,000	100,000	100,000	150,000
DTLZ4	40,000	70,000	80,000	100,000	150,000
WFG1-2	40,000	60,000	70,000	90,000	95,000
WFG3-9	35,000	55,000	70,000	85,000	90,000

TABLE V
T VALUES OF KNEA AND KNPSO

| Function | Number of Objectives | | | | |
	3	5	8	10	15
DTLZ1	0.6	0.6	0.1	0.2	0.1
DTLZ3	0.6	0.4	0.1	0.1	0.1
WFG4	0.6	0.5	0.2	0.3	0.3
WFG9	0.6	0.4	0.3	0.3	0.3
others	0.6	0.5	0.5	0.4	0.4

TABLE VI
S VALUES OF CDAS-SMPSO

| Function | Number of Objectives | | | | |
	3	5	8	10	15
DTLZ1	0.30	0.25	0.35	0.40	0.45
DTLZ2	0.25	0.25	0.35	0.45	0.40
DTLZ3	0.45	0.45	0.40	0.40	0.45
DTLZ4	0.35	0.25	0.25	0.45	0.40
WFG1	0.35	0.30	0.35	0.45	0.45
WFG2	0.25	0.35	0.35	0.45	0.35
WFG3	0.25	0.45	0.40	0.40	0.45
WFG4	0.35	0.35	0.25	0.45	0.45
WFG5	0.45	0.40	0.40	0.40	0.40
WFG6	0.35	0.30	0.30	0.40	0.45
WFG7	0.40	0.30	0.35	0.35	0.40
WFG8	0.35	0.25	0.25	0.45	0.45
WFG9	0.30	0.25	0.35	0.45	0.40

Inverted Generation Distance: The IGD metric [72] measures both solution spread and convergence of the approximated front POF^* towards the Pareto-optimal front. Let POF' be a set of uniformly spread Pareto-optimal points. IGD is calculated using

$$IGD = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{|POF'|} \qquad (13)$$

where $|POF'|$ denotes the cardinality of POF' and d_i is the Euclidean distance in the objective space between solution i of POF' and the closest member of POF^*. Note that $POF^* = POF'$ when $IGD = 0$.

In this work, the approach introduced by Deb and Jain [24] is employed to obtain POF'. This method utilizes the analytical forms of the DTLZ functions in combination with the evenly spread weight vectors required by the decomposition-based approaches. Specifically, for each evenly spread weight vector $\vec{w}$, the intersection of the Pareto-optimal surface is calculated to obtain a Pareto-optimal point $\vec{x}^*$ as

$$\text{DTLZ1: } f_i(\vec{x}^*) = 0.5 \times \frac{w_i}{\sum_{j=1}^{n_c} w_j} \qquad (14)$$

$$\text{DTLZ2-4: } f_i(\vec{x}^*) = \frac{w_i}{||\vec{w}||} \qquad (15)$$

where $i \in \{1, ..., n_c\}$ and $||\vec{w}||$ denotes the norm of $\vec{w}$. Note that as intersection points are calculated using the weight vectors of the decomposition-based approaches, the number of IGD points for each n_c is identical to the values in Table III.

that a solution set is not worse than some other solution set for all pairs of solutions [69]. Note that calculation of the hypervolume metric does not require the Pareto-optimal front.

In this work, exact hypervolume is calculated using the WFG algorithm [39] for problems with ten or less objectives. When more than ten objectives are present, hypervolume is estimated using a Monte Carlo sampling technique [26] with 10,000,000 sampling points. In all experiments, each objective value is normalized to the range [0,1] before calculating the hypervolume, where 0 and 1 correspond to the best and worst objective values in the approximated front, respectively. The reference point is chosen to be a vector with values slightly larger than the nadir point, which is the unit vector of length n_c. The motivation behind this choice of reference point is to include the hypervolume contribution of each extreme Pareto optimal solution [70]. Specifically, the reference point in all experiments is set to $(1.1, 1.1, ..., 1.1)$, as this vector has been shown to appropriately emphasize the convergence and diversity of the solution set [71].

V. EXPERIMENTAL RESULTS AND DISCUSSION

This section presents and discusses the results of all experiments performed. Here, the intent is to investigate the overall scalability of the selected many-objective optimizers. This is accomplished by investigating objective scalability and decision variable scalabity in separate subsections. Algorithms are evaluated using the IGD metric for the DTLZ functions, since a set of evenly spread Pareto-optimal points can be easily calculated via the technique in Section IV.D using the analytical forms of each DTLZ function. Since not all of the WFG functions have simple analytical forms for the many-objective cases, hypervolume is used to measure algorithm performance on the WFG suite. This approach is common in many-objective literature, e.g. see [44], [36].

A. Objective Scalability Experiments

First, the scalability of each selected CIMOO is evaluated over an increasing number of objectives. To conform to the recommendations in [63], $n = k + l$ decision variables are employed for each WFG test instance, where $k = 2(n_c - 1)$ is the number of decision variables relating to position and $l = 20$ is the number of decision variables relating to distance. For DTLZ test instances, the number of decision variables is set to $n = n_c + r - 1$ in accordance with the recommendations given in [62], where $r = 5$ for DTLZ1 and $r = 10$ for DTLZ2 to DTLZ4.

Table VII presents IGD metric results of the objective scalability experiments on DTLZ1 to DTLZ4. Hypervolume metric results are given for WFG1 to WFG5 and WFG6 to WFG9 in Tables VIII and IX, respectively. For each tested benchmark function, the algorithm(s) which achieved the best rank are highlighted in gray and written in boldface. Several observations for each algorithm are present in these tables. NSGA-III showed satisfactory scalability in several cases. For the WFG7 problem, NSGA-III performed worst overall for the three-objective instance, but scaled to third best when the number of objectives were increased to 15. Here, the reference point-based diversity preservation mechanism of NSGA-III seemed to aid in maintaining overall convergence and diversity on the concave, separable WFG7 function. Similar scalability was observed for the biased DTLZ4 function and non-separable WFG9 function. Note that NSGA-III performed especially poorly for the WFG1 function, which possesses a flat bias and mixed geometries, as seven of the other eight algorithms outperformed NSGA-III regardless of the number of objectives used.

MOEA/D with PBI aggregation exhibited competitive performance on the DTLZ functions, as it performed best overall for the 3- and 10-objective instances of DTLZ2. DTLZ2 is a relatively simply MOP with a spherical Pareto-optimal front, seemingly solved easily by MOEA/D. It is worth noting that MOEA/D scaled very well for the 10-objective instance of DTLZ4, which is surprising considering that DTLZ possesses the non-uniform property, known to be a weakness of decomposition-based optimizers such as MOEA/D. This weakness is a result of the reference point-based approach, since a uniformly spread set of reference points does not correspond to a uniformly spread set of Pareto-optimal solutions for non-uniform MOPs. Objective scalability on the WFG functions in general was less satisfactory for MOEA/D, especially for WFG1, WFG4, WFG7 and WFG9. MOEA/D scaled poorly in comparison to the other tested algorithm for each of these functions, as a large increase in rank was observed when comparing the 15-objective instances to the 3-objective instances.

MOEA/DD performed very well on DTLZ1 to DTLZ4, ranking best overall for the 5-, 8- and 10- objective instances of DTLZ2, the 10- and 15-objective instance of DTLZ3, and all instances of DTLZ4. In each of these cases, the combination of dominance and decomposition in MOEA/DD produced a better approximated front and a better spread set of solutions than the other tested CIMOOs. Li *et. al* [44] attribute this phenomenon to MOEA/DD's advanced technique for balancing convergence and diversity. MOEA/DD also scaled well in general on the WFG functions, where it performed best for the 15-objective instances of WFG2, WFG4, WFG5, WFG7 and WFG8. The only cases where MOEA/DD performed poorly were the 5-, 10- and 15-objective instances of the WFG3 function. WFG3, whose Pareto-optimal front is the connected version of the disconnected WFG2 function, is characterized by its non-separability and linear Pareto-optimal front shape.

The dMOPSO algorithm obtained promising results for DTLZ1 and DTLZ3, where it was the second best algorithm in all instances except when the number of objectives were set to 15 for DTLZ3. The worst performance observed on the DTLZ functions is seen for DTLZ4, where dMOPSO performed fifth overall for the 5-,8-,10- and 15-objective test instances. For the deceptive WFG5 function, dMOPSO scaled very poorly as it degraded to worst overall when the number of objectives were increased to 15. The scalability of dMOPSO was also subpar on the WFG9 function, as its rank increased from 2 to 8 when the number of objectives were increased from 3 to 15. Note that WFG9 also possesses a deceptive modality, which suggests that dMOPSO experiences performance degradation in the face of deceptive MaOPs.

IGD metric ranking of SrEA was generally poor in comparison to the other CIMOOs on the DTLZ functions. For DTLZ1 to DTLZ4, SrEA was the worst performer overall on the 15-objective instances, indicative of poor overall scalability with respect to the number of objectives. Here, the gradual shift of emphasis from convergence to diversity in SrEA seemed unable to produce a satisfactory set of solutions that were simultaneously both well-converged and uniformly spread. However, when tasked with solving the biased WFG1 function, which possesses both convex and concave geometries, SrEA obtained the second best rank for the 3- and 5-objective instances, and the best overall rank on the 8-, 10- and 15-objective instances. SrEA also exhibited competitive performance on the 3-, 5- and 8-objective test instances of WFG3 and the 3-objective instance of WFG4 and WFG7.

The performance ranks of the KnPSO algorithm was generally poor on DTLZ1-4, as the best rank observed over all instances was only five. However, for DTLZ1, the rank of KnPSO improved from ninth best for the 5-objective test instance to sixth best for the 15-objective instance. A similar

TABLE VII
OBJECTIVE SCALABILITY OF EACH ALGORITHM IS DISPLAYED WITH RESPECT TO THE IGD METRIC USING MANN-WHITNEY DIFFERENCES AND RANKS ON DTLZ1-4

Function	Objectives	Result	NSGA-III	MOEA/D	MOEA/DD	dMOPSO	SrEA	KnPSO	HypE	KnEA	CDAS-SMPSO
DTLZ1	3	Difference	-4	+1	+1	+6	-5	-2	0	-5	**+8**
		Rank	7	3	3	2	8	6	5	8	**1**
	5	Difference	-1	+2	+4	+6	-6	-6	-1	-6	**+8**
		Rank	5	4	3	2	7	7	5	7	**1**
	8	Difference	-4	+2	+4	+6	-8	-4	-4	0	**+8**
		Rank	6	4	3	2	9	6	6	5	**1**
	10	Difference	-5	+3	+3	+6	-8	-5	-2	0	**+8**
		Rank	7	3	3	2	9	7	6	5	**1**
	15	Difference	-6	+2	+4	+6	-8	-2	-2	-2	**+8**
		Rank	8	4	3	2	9	5	5	5	**1**
DTLZ2	3	Difference	+4	**+8**	+6	+2	-4	-2	0	-6	-8
		Rank	3	**1**	2	4	7	6	5	8	9
	5	Difference	+4	+6	**+8**	+2	-2	-7	-7	0	-4
		Rank	3	2	**1**	4	6	8	8	5	7
	8	Difference	+4	+6	**+8**	+2	-2	-8	-5	0	-5
		Rank	3	2	**1**	4	6	9	7	5	7
	10	Difference	+1	**+8**	+6	+3	-7	-7	-1	+1	-4
		Rank	4	**1**	2	3	8	8	6	4	7
	15	Difference	-2	+6	**+8**	+4	-8	-6	-1	+2	-3
		Rank	6	2	**1**	3	9	8	5	4	7
DTLZ3	3	Difference	-2	-2	+2	+6	-6	-2	+2	-6	**+8**
		Rank	5	5	3	2	8	5	3	8	**1**
	5	Difference	-3	+1	+4	+6	-6	-8	+1	-3	**+8**
		Rank	6	4	3	2	8	9	4	6	**1**
	8	Difference	-4	0	+4	+6	-7	-7	-1	+1	**+8**
		Rank	7	5	3	2	8	8	6	4	**1**
	10	Difference	-4	+3	**+8**	+6	-8	-6	0	-2	+3
		Rank	7	3	**1**	2	9	8	5	6	3
	15	Difference	-4	0	**+8**	+4	-8	-4	+2	-4	+6
		Rank	6	5	**1**	3	9	6	4	6	2
DTLZ4	3	Difference	-2	-4	**+8**	+2	-5	+2	-4	+5	-2
		Rank	5	7	**1**	3	9	3	7	2	5
	5	Difference	+3	+1	**+8**	-1	-1	-8	-6	+5	-1
		Rank	3	4	**1**	5	5	9	8	2	5
	8	Difference	+6	-2	**+8**	-1	-3	-5	-8	+4	+1
		Rank	2	6	**1**	5	7	8	9	3	4
	10	Difference	+6	+2	**+8**	0	-3	-3	-6	-7	+3
		Rank	2	4	**1**	5	6	6	8	9	3
	15	Difference	+4	+4	**+8**	0	-8	-4	-6	-1	+3
		Rank	2	2	**1**	5	9	7	8	6	4

observation was seen for DTLZ1, where KnPSO ranked fifth best when the number of objectives were increased to 15. Here, KnPSO begins to demonstrate the ability to scale reasonably well, thus further research into the scalability of KnPSO on the DTLZ test suite is encouraged, specifically for instances where the number of objectives are increased past 15. Results on the WFG functions demonstrated the knee-point driven technique of KnPSO to perform well when the number of objectives were low, as it ranked second or better for the 3-objective instances of WFG3, WFG6, WFG7, and WFG9. However, when the number of objectives were increased, KnPSO decreased its rank on each of these functions except for WFG6. KnPSO handled the non-separability and reduction of WFG6 well, obtaining the highest hypervolume rank out of all tested CIMOOs.

HypE shows promise when the number of objectives are low, as it ranked competitively on the 3-objective instances of DTLZ2, DTLZ3, WFG4 and WFG5. However, in nearly all cases where the number of objectives were increased past three, HypE generally did not rank very well for both the WFG and DTLZ function suites. There are a few exceptions

to this, as HypE performed best overall for the 15-objective instance of the linear WFG3 function and also performed reasonably well for all instances of WFG5. Since HypE estimates hypervolume contributions when $n_c \geq 3$, its poor objective scalability can be attributed to inaccurate fitness value assignments. An obvious solution to this issue is to simply increase the number of Monte Carlo sampling points employed during fitness calculation. However, the runtime of HypE has already been shown to be considerably higher than many other CIMOOs when 10,000 sampling points are employed [36], thus any further increases may quickly cause the computational complexity of HypE to become unacceptably large for MaOPs.

KnEA scaled favorably when the number of objectives were increased for all DTLZ functions except for DTLZ4, even though it did not actually perform best in any of these instances. DTLZ4 challenges MOOs by investigating their ability to maintain solution diversity in the objective space, which KnEA had difficulty with. For the WFG function set, KnEA performed very well, often outperforming all other algorithms. Specifically, KnEA yielded the best hypervolume

TABLE VIII
OBJECTIVE SCALABILITY OF EACH ALGORITHM IS DISPLAYED WITH RESPECT TO THE HYPERVOLUME METRIC USING MANN-WHITNEY
DIFFERENCES AND RANKS ON WFG1-5

Function	Objectives	Result	Algorithm								
			NSGA-III	MOEA/D	MOEA/DD	dMOPSO	SrEA	KnPSO	HypE	KnEA	CDAS-SMPSO
WFG1	3	Difference	-6	-4	+2	+1	+4	+2	-8	**+8**	+1
		Rank	8	7	3	5	2	3	9	**1**	5
	5	Difference	-6	+5	-4	0	+5	0	-8	**+8**	0
		Rank	8	2	7	4	2	4	9	**1**	4
	8	Difference	-7	**+7**	-1	-1	**+7**	-1	-7	+4	-1
		Rank	8	**1**	4	4	**1**	4	8	3	4
	10	Difference	-7	+2	+5	-4	**+8**	-1	-7	+5	-1
		Rank	8	4	2	7	**1**	5	8	2	5
	15	Difference	-7	+2	+4	-4	**+8**	-1	-7	+6	-1
		Rank	8	4	3	7	**1**	5	8	2	5
WFG2	3	Difference	+3	-8	-3	-3	+3	0	-3	+3	**+8**
		Rank	2	9	6	6	2	5	6	2	**1**
	5	Difference	+6	-7	0	0	+5	0	-5	**+7**	-6
		Rank	2	9	4	4	3	4	7	**1**	8
	8	Difference	+2	-8	**+8**	+2	+2	+2	-6	+2	-4
		Rank	2	9	**1**	2	2	2	8	2	7
	10	Difference	+3	-7	**+6**	0	+1	0	-3	**+6**	-6
		Rank	3	9	**1**	5	4	5	7	**1**	8
	15	Difference	**+6**	-6	**+6**	-2	+2	-4	0	**+6**	-8
		Rank	**1**	8	**1**	6	4	7	5	**1**	9
WFG3	3	Difference	+4	-7	+4	-3	+4	**+5**	-3	+3	-7
		Rank	2	8	2	6	2	**1**	6	5	8
	5	Difference	+5	-6	-8	0	+6	-3	+2	**+6**	-2
		Rank	3	8	9	5	1	7	4	**1**	6
	8	Difference	-3	-8	**+5**	-3	**+5**	-3	-3	**+5**	**+5**
		Rank	5	9	**1**	5	**1**	5	5	**1**	**1**
	10	Difference	**+6**	-8	-4	-6	+5	-1	-1	**+6**	+3
		Rank	**1**	9	7	8	3	5	5	**1**	4
	15	Difference	+5	-8	-3	-6	-3	+3	**+8**	+3	+1
		Rank	2	9	6	8	6	3	**1**	3	5
WFG4	3	Difference	0	+2	**+4**	-1	**+4**	-5	+2	+2	-8
		Rank	6	3	**1**	7	**1**	8	3	3	9
	5	Difference	0	+4	0	+6	-5	0	-8	**+8**	-5
		Rank	4	3	4	2	7	4	9	**1**	7
	8	Difference	0	0	**+7**	-1	-3	+4	-8	**+7**	-6
		Rank	4	4	**1**	6	7	3	9	**1**	8
	10	Difference	+3	-3	+6	-7	-3	+3	-7	**+8**	0
		Rank	3	6	2	8	6	3	8	**1**	5
	15	Difference	0	-8	**+7**	-5	+2	+4	-5	**+7**	-2
		Rank	5	9	**1**	7	4	3	7	**1**	6
WFG5	3	Difference	**+1**	0	**+1**	0	0	0	0	**+1**	-4
		Rank	**1**	6	**1**	6	6	6	6	**1**	9
	5	Difference	+5	+2	+4	-5	+1	-2	-5	**+8**	-8
		Rank	2	4	3	7	5	6	7	**1**	9
	8	Difference	+4	-1	+6	-8	+2	-1	-4	**+8**	-6
		Rank	3	5	2	9	4	5	7	**1**	8
	10	Difference	+4	-4	**+8**	-8	+2	0	-2	+6	-6
		Rank	3	7	**1**	9	4	5	6	2	8
	15	Difference	+5	-4	**+8**	-8	+2	-1	-1	+5	-6
		Rank	2	7	**1**	9	4	5	5	2	8

rank for the 5-, 8-, 10- and 15- objective instances of WFG4, WFG6 and WFG9. KnEA also performed notably well for the deceptive WFG5 function. However, it was outperformed in the 10- and 15-objective instances by MOEA/DD. The prioritization of knee points during mating selection and environmental selection seemed to obtain a well-converged, well-distributed set of solutions on the majority of functions. Note that knee point prioritization has been shown to produce higher hypervolume values [36], which may explain the highly competitive hypervolume performance of KnEA.

CDAS-SMPSO obtained its best performance on the linear DTLZ1, where it yielded the best IGD rank in all instances. CDAS-SMPSO seemed to be mislead less than the other optimizers by the $11^5 - 1$ local optima of DTLZ1, indicating that CDAS-SMPSO performed quite well in the face of multi-modality. This observation is further supported by the favorable performance of CDAS-SMPSO on DTLZ3, which is another function possessing the multi-modal property, as CDAS-SMPSO had the best performance for the 3-, 5- and 8-objective test instances. On the WFG functions, CDAS-SMPSO performed well in some cases when the number of objectives were low, outperforming all other algorithms for the 3-objective instances on WFG2, WFG8 and WFG9. Here, the dominance area shrinkage technique of CDAS-SMPSO seemed to be successful on the multi-objective problems, as limiting the dominance region of each solution

TABLE IX
OBJECTIVE SCALABILITY OF EACH ALGORITHM IS DISPLAYED WITH RESPECT TO THE HYPERVOLUME METRIC USING MANN-WHITNEY
DIFFERENCES AND RANKS ON WFG6-9

Function	Objectives	Result	Algorithm								
			NSGA-III	MOEA/D	MOEA/DD	dMOPSO	SrEA	KnPSO	HypE	KnEA	CDAS-SMPSO
WFG6	3	Difference	+2	-8	+7	0	+1	+7	-5	+2	-5
		Rank	5	9	1	6	4	1	7	3	7
	5	Difference	+1	-4	+4	+4	-4	+3	-8	+8	-4
		Rank	5	6	2	2	6	4	9	1	6
	8	Difference	+3	-7	+6	-3	-3	+3	-7	+8	0
		Rank	3	8	2	6	6	3	8	1	5
	10	Difference	+2	-8	+5	-5	-2	+5	-5	+8	0
		Rank	4	9	2	7	6	2	7	1	5
	15	Difference	+2	-8	+4	-6	0	+7	-3	+7	-3
		Rank	4	9	3	8	5	1	6	1	6
WFG7	3	Difference	-6	+1	+8	-2	+1	+4	-4	0	-2
		Rank	9	3	1	6	3	2	8	5	6
	5	Difference	+1	+5	+8	-4	-3	+1	-6	+5	-7
		Rank	4	2	1	7	6	4	8	2	9
	8	Difference	0	+3	+8	-8	-2	+2	-3	+6	-6
		Rank	5	3	1	9	6	4	7	2	8
	10	Difference	+4	-2	+6	-8	-1	+2	-3	+8	-6
		Rank	3	6	2	9	5	4	7	1	8
	15	Difference	+4	-4	+8	-7	+2	0	-2	+6	-7
		Rank	3	7	1	8	4	5	6	2	8
WFG8	3	Difference	-1	-1	+1	-4	+1	-1	-4	+1	+8
		Rank	5	5	2	8	2	5	8	2	1
	5	Difference	+1	+6	+2	-1	-5	+2	-5	+8	-8
		Rank	5	2	3	6	7	3	7	1	9
	8	Difference	+4	-4	+8	-8	-4	+2	0	+6	-4
		Rank	3	6	1	9	6	4	5	2	6
	10	Difference	+4	-7	+7	-7	-3	+1	+1	+7	-3
		Rank	3	8	1	8	6	4	4	1	6
	15	Difference	+6	-8	+7	-6	+2	-1	-1	+5	-4
		Rank	2	9	1	8	4	5	5	3	7
WFG9	3	Difference	-1	+1	+1	+2	+1	+2	-7	-7	+8
		Rank	7	4	4	2	4	2	8	8	1
	5	Difference	+5	+3	-2	+1	-3	+2	-8	+8	-6
		Rank	2	3	6	5	7	4	9	1	8
	8	Difference	+5	-3	+5	-7	0	+2	-7	+8	-3
		Rank	2	6	2	8	5	4	8	1	6
	10	Difference	+6	-6	+2	-8	+2	+2	-2	+8	-4
		Rank	2	8	3	9	3	3	6	1	7
	15	Difference	+6	-7	+3	-7	+1	+2	-3	+8	-3
		Rank	2	8	3	8	5	4	6	1	6

led to an improved convergence and diversity. When the number of objectives were increased, scalability of CDAS-SMPSO seemed to be highly variable, dependent on the function at hand. Favorable scalability is observed for WFG3-4, whereas poor scalability is exhibited for WFG2, WFG8 and WFG9.

B. Decision Variable Scalability Experiments

The scalability of each CIMOO is also evaluated with respect to an increasing number of decision variables. Table X presents IGD metric results for each algorithm using 30, 100, 500 and 1000 decision variables on DTLZ1-4. In Tables XI and XII, hypervolume metric results are given for WFG1-5 and WFG6-9, respectively. In each of these tables, the number of objectives n_c was set to the largest number of objectives for which the 1000-variable instances were computationally feasible to complete for each algorithm, determined by the authors' to be 10 objectives. Similar to Section V.A, each algorithm is discussed individually in this subsection, noting relevant observations where applicable. NSGA-III possessed poor scalability in general for DTLZ1 to DTLZ4. The largest

performance decrease was observed on DTLZ4, which contains a parametric variable mapping to DTLZ2, where the IGD rank of NSGA-III increased from 2 to 7 when the number of decision variables grew to 100. However, when tested on the WFG functions, NSGA-III seemed to scale better than the other tested EAs in many instances, despite being often outperformed by the PSO algorithms. For WFG2, WFG4, WFG5 and WFG7 it was the third best evolutionary optimizer on the 500-variable instance. On the 1000-variable instance of WFG5 and WFG8, NSGA-III was the second best EA. Note that in all of these instances, NSGA-III was outperformed by the three tested PSO algorithms. Regardless, NSGA-III scaled reasonably well when tasked with an increasing number of decision variables. Using a reference point-based selection technique seemed to provide adequate convergence and diversity in many cases.

MOEA/D with PBI aggregation achieved better scalability than the majority of the other tested EAs on DTLZ1 to DTLZ4, but was outperformed by CDAS-SMPSO, dMOPSO and KnPSO for 11 out of 16 test instances. Here, MOEA/D was often unable to produce a better convergence and diversity

TABLE X
DECISION VARIABLE SCALABILITY OF EACH ALGORITHM IS DISPLAYED WITH RESPECT TO THE IGD METRIC USING MANN-WHITNEY DIFFERENCES AND RANKS ON DTLZ1-4 WITH 10 OBJECTIVES

Function	Variables	Result	Algorithm								
			NSGA-III	MOEA/D	MOEA/DD	dMOPSO	SrEA	KnPSO	HypE	KnEA	CDAS-SMPSO
DTLZ1	30	Difference	-6	+2	+4	+6	-8	-2	-2	-2	+8
		Rank	8	4	3	2	9	5	5	5	1
	100	Difference	-6	-4	+4	+6	-8	0	+2	-2	+8
		Rank	8	7	3	2	9	5	4	6	1
	500	Difference	-7	-1	+2	+7	-5	+4	-6	-1	+7
		Rank	9	5	4	1	7	3	8	5	1
	1000	Difference	-6	-1	+2	+6	-4	+4	-5	-4	+8
		Rank	9	5	4	2	6	3	8	6	1
DTLZ2	30	Difference	-2	+8	+6	+4	-8	-6	-1	+2	-3
		Rank	6	1	2	3	9	8	5	4	7
	100	Difference	-6	+6	+7	+5	-8	-1	-4	-1	+2
		Rank	8	2	1	3	9	5	7	5	4
	500	Difference	-6	-2	0	+8	-8	+4	-4	+2	+6
		Rank	8	6	5	1	9	3	7	4	2
	1000	Difference	-6	-2	-2	+7	-8	+4	-2	+2	+7
		Rank	8	5	5	1	9	3	5	4	1
DTLZ3	30	Difference	-4	0	+4	+6	-8	-4	+2	-4	+8
		Rank	6	5	3	2	9	6	4	6	1
	100	Difference	-4	-1	+3	+6	-8	-1	-6	+3	+8
		Rank	7	5	3	2	9	5	8	3	1
	500	Difference	-4	-2	0	+7	-7	+3	-7	+3	+7
		Rank	7	6	5	1	8	3	8	3	1
	1000	Difference	-4	-2	0	+6	-7	+4	-7	+2	+8
		Rank	7	6	5	2	8	3	8	4	1
DTLZ4	30	Difference	+4	+4	+8	0	-8	-4	-6	-1	+3
		Rank	2	2	1	5	9	7	8	6	4
	100	Difference	-4	+3	+7	+1	-6	-1	-8	+1	+7
		Rank	7	3	1	4	8	6	9	4	1
	500	Difference	-4	0	+8	+6	-7	+4	-7	-2	+2
		Rank	7	5	1	2	8	3	8	6	4
	1000	Difference	-4	0	+8	+6	-7	+4	-7	-2	+2
		Rank	7	5	1	2	8	3	8	6	4

than the PSO approaches, despite yielding better performance than the other EAs. Note that MOEA/D performed very well for the 100-variable instances of DTLZ2 and DTLZ4, indicating that MOEA/D may yield good results even when 100 decision variables are employed. Performance of MOEA/D was not nearly as satisfactory on the WFG functions, except for WFG1 and WFG2, where MOEA/D scaled to third and fourth best performer, respectively, for the 1000-variable cases. It is notable that MOEA/D, MOEA/DD and dMOPSO, which each utilize the PBI aggregation approach, scaled exceptionally well on the WFG1 function for the 100-, 500- and 1000-variable cases. The aggregative decomposition technique seemed to scale better than all other approaches on the flat bias and mixed Pareto-optimal front geometries experienced on WFG1 when the number of decision variables were increased up to 1000.

MOEA/DD exhibited very promising performance on the tested large-scale MaOPs, performing well on most of the DTLZ1 to DTLZ4 instances. For DTLZ4 in particular, MOEA/DD demonstrated an exceptional ability to find a converged and well-spread set of solutions, outperforming all other algorithms for 31/32 instances, where no statistically significant difference was found between itself and CDAS-SMPSO in the single case where MOEA/DD did not show better performance. Competitive scalability was also shown for MOEA/DD on the WFG1, WFG2, WFG7 and WFG9 functions. Note that the non-separable and reduced WFG6 problem seemed to present the most difficulty for

MOEA/DD on the large-scale instances, as it obtained a rank of 9 and 7 when 500 and 1000 decision variables were employed, respectively. Aside from this, MOEA/DD was the overall best-performing evolutionary optimizer in many cases for the tested large-scale instances. It is hypothesized that the unique subregion-based mating restriction scheme of MOEA/DD improves the effectiveness of the crossover operator for the immense search space encountered in large-scale MaOPs. Since parents are nearly always selected from neighbouring subregions, mated solutions should be kept reasonably close in the objective space, helping to avoid the production of distant children in many instances. Thus, the degradation experienced by the crossover operator in very large search spaces, described in [24], may be partially alleviated by MOEA/DD.

dMOPSO scaled very well on all of the DTLZ functions, performing best overall for the 500-variable instances of DTLZ1 to DTLZ3 and the 1000-variable instance of DTLZ2. Notable scalability was achieved on the DTLZ4 function, where the IGD rank of dMOPSO decreased from 5 to 2 when the number of decision variables were increased from 30 to 1000. For the WFG function suite, dMOPSO scaled well for all problems except WFG4, WFG5 and WFG7, where it possessed poor performance in all cases. Note that degradation was severe for each of these functions, as dMOPSO was outperformed by all other PSO optimizers and EAs, indicating that these functions in particular may present significant

TABLE XI
DECISION VARIABLE SCALABILITY OF EACH ALGORITHM IS DISPLAYED WITH RESPECT TO THE HYPERVOLUME METRIC USING MANN-WHITNEY DIFFERENCES AND RANKS ON WFG1-5 WITH 10 OBJECTIVES

Function	Variables	Result	Algorithm								
			NSGA-III	MOEA/D	MOEA/DD	dMOPSO	SrEA	KnPSO	HypE	KnEA	CDAS-SMPSO
WFG1	30	Difference	-7	+4	+4	-4	+8	-1	-7	+2	-1
		Rank	8	2	2	7	1	5	8	4	5
	100	Difference	-8	+6	+8	+2	-4	+2	-6	-2	+2
		Rank	9	2	1	3	7	3	8	6	3
	500	Difference	-6	+6	+8	+4	-4	+1	-8	-2	+1
		Rank	8	2	1	3	7	4	9	6	4
	1000	Difference	-6	+2	+8	+6	-4	+2	-8	-2	+2
		Rank	8	3	1	2	7	3	9	6	3
WFG2	30	Difference	+3	-7	+6	0	+1	0	-3	+6	-6
		Rank	3	9	1	5	4	5	7	1	8
	100	Difference	+5	-7	+6	-5	+7	-3	-5	+1	+1
		Rank	3	9	2	7	1	6	7	4	4
	500	Difference	-1	+2	+8	-8	+2	+6	-5	-3	-1
		Rank	5	3	1	9	3	2	8	7	5
	1000	Difference	-2	+2	+6	-8	0	+8	-6	-4	+4
		Rank	6	4	2	9	5	1	8	7	3
WFG3	30	Difference	+6	-8	-4	-6	+5	-1	-1	+6	+3
		Rank	1	9	7	8	3	5	5	1	4
	100	Difference	+4	-8	-5	-4	+5	-1	+5	-2	+6
		Rank	4	9	8	7	2	5	2	6	1
	500	Difference	0	-8	-5	+5	+2	+5	-5	-2	+8
		Rank	5	9	7	2	4	2	7	6	1
	1000	Difference	0	-8	-4	+6	+2	+4	-6	-2	+8
		Rank	5	9	7	2	4	3	8	6	1
WFG4	30	Difference	+3	-3	+6	-7	-3	+3	-7	+8	0
		Rank	3	6	2	8	6	3	8	1	5
	100	Difference	0	-6	+8	-8	+3	+6	-3	-3	+3
		Rank	5	8	1	9	3	2	6	6	3
	500	Difference	0	-2	+8	-8	+3	+3	-4	-6	+6
		Rank	5	6	1	9	3	3	7	8	2
	1000	Difference	0	-2	+7	-8	+3	+4	-4	-6	+7
		Rank	5	6	1	9	4	3	7	8	1
WFG5	30	Difference	+4	-4	+6	-8	+2	0	-2	+8	-6
		Rank	3	7	2	9	4	5	6	1	8
	100	Difference	-2	-2	+8	-8	+5	+2	-6	+5	-2
		Rank	5	5	1	9	2	4	8	2	5
	500	Difference	0	+4	-3	-8	+2	+8	-3	-6	+6
		Rank	5	3	6	9	4	1	6	8	2
	1000	Difference	0	-4	0	-8	+4	+8	0	-6	+6
		Rank	4	7	4	9	3	1	4	8	2

difficulties for dMOPSO regardless of the number of decision variables employed.

IGD metric performance of the SrEA algorithm was subpar on DTLZ1 to DTLZ4, where SrEA was the worst overall performing algorithm for 7/16 instances. Note that on DTLZ1, SrEA began to scale well as the number of decision variables increased, as it obtained a rank of 7 and 6 for the 500- and 1000-variable cases, respectively. Further testing of SrEA on DTLZ1 is encouraged, as it may scale well when more than 1000 decision variables are employed if it continues to follow the exhibited trend. The scalability of SrEA was competitive for WFG4, as SrEA outperformed all of the other optimizers for the 500- and 1000-variable instances of this function. Here, the linearly increasing shift from convergence to spread employed by SrEA was adept at producing excellent hypervolume values. Another notable observation is that SrEA scaled worse than the PSO optimizers on several problems, namely WFG1 to WFG3, WFG7 and WFG9.

Scalability of the KnPSO algorithm with respect to the number of decision variables was in general much more satisfactory than its objective scalability observed in Section V.A. KnPSO is the medium-high performer in most large-scale cases, as it performed third best or better for 23/26 instances where the number of decision variables were set to 500 or greater. KnPSO performed best overall on the 500- and 1000-variable instances of WFG8 and the 1000-variable instance of the challenging WFG2 problem, which consists of several disconnected convex segments and has non-separable variables. Here, the focus on selecting and preserving knee points in the approximated front allowed the swarm to discover solutions that were converging well in non-crowded regions, leading to a good blend of convergence and diversity.

HypE encountered considerable difficulty for large-scale MaOPs on the DTLZ function set. HypE possessed the overall worst scalability for DTLZ1, DTLZ2 and DTLZ4, where for the 500- and 1000-decision variable instances it achieved a rank of 8 on all functions. On the WFG1 function, HypE experienced significant difficulties regardless of the number of decision variables employed, performing eight worst overall for the 30- and 100-variable cases and ninth worst overall for the 500- and 1000-variable instances. Performance also declined significantly for the higher-dimensional cases of WFG2 and

TABLE XII
DECISION VARIABLE SCALABILITY OF EACH ALGORITHM IS DISPLAYED WITH RESPECT TO THE HYPERVOLUME METRIC USING MANN-WHITNEY DIFFERENCES AND RANKS ON WFG6-9 WITH 10 OBJECTIVES

Function	Variables	Result	Algorithm								
			NSGA-III	MOEA/D	MOEA/DD	dMOPSO	SrEA	KnPSO	HypE	KnEA	CDAS-SMPSO
WFG6	30	Difference	0	-8	+5	-5	-2	+5	-5	+8	+2
		Rank	5	9	2	7	6	2	7	1	4
	100	Difference	-6	-1	+6	-1	-4	+4	-8	+2	+8
		Rank	8	5	2	5	7	3	9	4	1
	500	Difference	-4	+1	-8	+5	+1	+5	-3	-5	+8
		Rank	7	4	9	2	4	2	6	8	1
	1000	Difference	-3	-5	-5	+7	+2	+4	0	-7	+7
		Rank	6	7	7	1	4	3	5	9	1
WFG7	30	Difference	+4	-2	+6	-8	-1	+2	-3	+8	-6
		Rank	3	6	2	9	5	4	7	1	8
	100	Difference	0	+4	+8	-8	+6	+2	-5	-4	-3
		Rank	5	3	1	9	2	4	8	7	6
	500	Difference	0	-6	+8	-8	-3	+3	+3	-3	+6
		Rank	5	8	1	9	6	3	3	6	2
	1000	Difference	0	-5	+6	-8	-5	+4	+2	-2	+8
		Rank	5	7	2	9	7	3	4	6	1
WFG8	30	Difference	+4	-7	+7	-7	-3	+1	+1	+7	-3
		Rank	3	8	1	8	6	4	4	1	6
	100	Difference	+4	-8	+8	-6	+5	+2	-2	-3	0
		Rank	3	9	1	8	2	4	6	7	5
	500	Difference	+2	-2	+4	-6	0	+8	-4	-8	+6
		Rank	4	6	3	8	5	1	7	9	2
	1000	Difference	+1	-6	+4	-3	+1	+8	-2	-8	+6
		Rank	4	8	3	7	4	1	6	9	2
WFG9	30	Difference	+6	-6	+2	-8	+2	+2	-2	+8	-4
		Rank	2	8	3	9	3	3	6	1	7
	100	Difference	+8	-7	+4	-7	+6	0	-1	-1	-2
		Rank	1	8	3	8	2	4	5	5	7
	500	Difference	-4	-8	-2	+2	-4	+6	+4	-3	+8
		Rank	8	9	5	4	8	2	3	6	1
	1000	Difference	-4	-6	+2	+4	-8	+6	-1	-1	+8
		Rank	7	8	4	3	9	2	5	5	1

WFG3. The hypervolume-driven fitness assignment of HypE generally seemed to be ineffective at identifying desirable solutions on large-scale MaOPs. However, one should note that there were several instances where HypE exhibited promise on large-scale MaOPs, namely for the 500- and 1000-variable instances of WFG7 and WFG9.

The performance of KnEA was very good when low amounts of decision variables were employed, but generally less satisfactory in high-dimensional decision space. For WFG2 to WFG9, KnEA was the best performer overall for the 30-variable cases, yet dropped to sixth best or worse when 1000 decision variables were employed. Note that the drop in performance on large-scale instances was not as drastic for the DTLZ functions, and in the case of DTLZ3 KnEA scaled well. The drop in performance on large-scale MaOPs seen for KnEA (and several other EAs) is potentially a result of the crossover operator difficulties encountered for large search spaces [24]. Based on the competitive performance of MOEA/DD in this section, it may be useful to employ a similar mating restriction scheme for KnEA when solving large-scale MaOPs.

CDAS-SMPSO was generally very competitive on the large-scale instances, as it scaled well to the increased number of decision variables. Notable performances were observed on DTLZ1 and DTLZ3, as CDAS-SMPSO was the best performing algorithm overall for all test instances on these functions. It is notable that CDAS-SMPSO decreased its rank as the number of decision variables grew for all WFG functions, indicative of excellent scalability overall. The dominance area control technique of CDAS-SMPSO seemed to provide excellent convergence and diversity, as the stricter dominance criterion led to the selection and preservation of better solutions. Given the satisfactory performance of CDAS-SMPSO and the other tested PSO algorithms on the large-scale MaOPs in this section, it is clear that the swarm intelligence approach in general scaled quite well to the increasing number of decision variables.

VI. CONCLUSION

This paper investigated the scalability of many-objective CIMOOs with respect to both the number of decision variables and objectives. To accomplish this, various many-objective algorithms were compared over 13 challenging MOPs using three to 15 objectives and 30 to 1000 decision variables. Results demonstrated the evolutionary approaches to generally scale well with respect to the number of objectives for low numbers of decision variables. The majority of evolutionary optimizers were able to maintain a good blend of convergence and diversity despite the increased size of the objective space. Techniques which performed especially well were the knee-driven approach of KnEA and the subregion-based decomposition approach of MOEA/DD. The dominance area shrinkage technique also performed competitively when the number of objectives grew, as CDAS-SMPSO was the best swarm intelligence optimizer for most of the MaOP instances.

When the number of decision variables were increased, performance of several evolutionary optimizers waned in many instances, while the PSO approaches scaled well generally. It was hypothesized that this phenomenon was a result of reduced crossover operator effectiveness due to the immense search space and higher levels of epistasis encountered in large-scale MaOPs. However, it was noted that the combination of dominance and subregion-based decomposition for MOEA/DD was very promising on large-scale MaOPs, as the algorithm found a well-spread, well-converged solution set in many instances despite the increase in decision variables.

While this work aims to serve as the starting point for research on large-scale many-objective optimization, a broad set of recommendations can be given for each algorithm category in regards to improving performance on large-scale MaOPs. For the decomposition-based optimizers, experimental results in this work suggest the use of subregion-based mating restriction schemes to ensure that parents are kept reasonably close to each other in the objective space, which may help to partially avoid the issue mentioned in [24]. Although not investigated in this work, applying a subregion-based mating restriction scheme in the decision space may also aid in handling this issue.

A similar idea can be applied to knee-driven algorithms by considering the objective- or decision-space distance of knee points during the mating selection phase, where closer knee point pairs are given a higher priority to mate together. For algorithms which fall under the indicator-based, dominance relation modification or non-Pareto categories, experimental results in this work highlighted the issue but did not produce a solution for improving performance. However, a general recommendation for these algorithm types is to integrate co-operative co-evolution principles [73] into their design, which have been shown to improve performance in multi-objective environments [74]. While the recommendations given here are sufficient to improve performance on large-scale many-objective problems, further research is encouraged to produce a more specific set of recommendations for each algorithm type.

A variety of interesting opportunities for future work within this area exist. It is within the realm of possibility that some EAs may simply take longer to converge on large-scale MaOPs, which would explain some of the performance disparities observed in this study. Thus, a future study will involve further analysis regarding the scalability of EAs on large-scale MaOPs, specifically using a larger number of function evaluations to allow more convergence time. Another direction is to propose creative many-objective optimizers designed with decision variable scalability in mind, as the many-objective issue in general is still far from being solved satisfactorily for large-scale instances. Novel mating restriction schemes can also be incorporated into a variety of evolutionary many-objective optimizers, evaluating the subsequent performance gains on large-scale MaOPs.

REFERENCES

[1] W.-z. Zhao, C.-y. Wang, L.-y. Yu, and T. Chen, "Performance optimization of electric power steering based on multi-objective genetic algorithm," *Journal of Central South University*, vol. 20, no. 1, pp. 98–104, 2013. [Online]. Available: http://dx.doi.org/10.1007/s11771-013-1464-4

[2] R. Chelouah and C. Baron, "Ant colony algorithm hybridized with tabu and greedy searches as applied to multi-objective optimization in project management," *Journal of Heuristics*, vol. 13, no. 6, pp. 640–640, 2007. [Online]. Available: http://dx.doi.org/10.1007/s10732-007-9043-1

[3] S. Kitayama, J. Srirat, M. Arakawa, and K. Yamazaki, "Sequential approximate multi-objective optimization using radial basis function network," *Structural and Multidisciplinary Optimization*, vol. 48, no. 3, pp. 501–515, 2013. [Online]. Available: http://dx.doi.org/10.1007/s00158-013-0911-z

[4] F. Djeffal and N. Lakhdar, "An improved analog electrical performance of submicron dual-material gate (dm) gaas-mesfets using multi-objective computation," *Journal of Computational Electronics*, vol. 12, no. 1, pp. 29–35, 2013. [Online]. Available: http://dx.doi.org/10.1007/s10825-012-0430-y

[5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, Apr 2002.

[6] 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, Dec 2007.

[7] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, Nov 1999.

[8] S. Kukkonen and J. Lampinen, "Gde3: the third evolution step of generalized differential evolution," in *IEEE Congress on Evolutionary Computation*, vol. 1, Sept 2005, pp. 443–450 Vol.1.

[9] N. Beume, B. Naujoks, and M. Emmerich, "Sms-emoa: Multiobjective selection based on dominated hypervolume," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653 – 1669, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221706005443

[10] A. Nebro, J. Durillo, J. Garcia-Nieto, C. Coello Coello, F. Luna, and E. Alba, "Smpso: A new pso-based metaheuristic for multi-objective optimization," in *IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making*, March 2009, pp. 66–73.

[11] K. E. Parsopoulos, D. K. Tasoulis, M. N. Vrahatis, and K. Words, "Multiobjective optimization using parallel vector evaluated particle swarm optimization," in *Proceedings of the IASTED International Conference on Artificial Intelligence and Applications, year = 2004, pages = 823–828, publisher = ACTA Press*.

[12] M. R. Sierra and C. A. C. Coello, "Improving pso-based multi-objective optimization using crowding, mutation and -dominance," in *In EMO2005, pages 505519. LNCS 3410*. Springer-Verlag, 2005, pp. 505–519.

[13] X. Li, *Genetic and Evolutionary Computation — GECCO 2003: Genetic and Evolutionary Computation Conference Chicago, IL, USA, July 12–16, 2003 Proceedings, Part I*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, ch. A Non-dominated Sorting Particle Swarm Optimizer for Multiobjective Optimization, pp. 37–48. [Online]. Available: http://dx.doi.org/10.1007/3-540-45105-6_4

[14] C. kin Chow and H.-T. Tsui, "Autonomous agent response learning by a multi-species particle swarm optimization," in *Congress on Evolutionary Computation*, vol. 1, June 2004, pp. 778–785 Vol.1.

[15] K. Ikeda, H. Kita, and S. Kobayashi, "Failure of pareto-based moeas: does non-dominated really mean near to optimal?" in *Proceedings of the 2001 Congress on Evolutionary Computation*, vol. 2, 2001, pp. 957–962 vol. 2.

[16] O. Schutze, A. Lara, and C. A. Coello Coello, "On the influence of the number of objectives on the hardness of a multiobjective optimization problem," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 4, pp. 444–455, Aug 2011.

[17] A. Sülflow, N. Drechsler, and R. Drechsler, *Evolutionary Multi-Criterion Optimization: 4th International Conference, EMO 2007, Matsushima, Japan, March 5-8, 2007. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, ch. Robust Multi-Objective Optimization in High Dimensional Spaces, pp. 715–726. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-70928-2_54

[18] Y. Yuan and H. Xu, "Multiobjective flexible job shop scheduling using memetic algorithms," *Automation Science and Engineering, IEEE Transactions on*, vol. 12, no. 1, pp. 336–353, Jan 2015.

[19] R. Lygoe, M. Cary, and P. Fleming, "A real-world application of a many-objective optimisation complexity reduction process," in *Evolutionary Multi-Criterion Optimization*, R. Purshouse, P. Fleming, C. Fonseca, S. Greco, and J. Shaw, Eds. Springer Berlin

Heidelberg, 2013, vol. 7811, pp. 641–655. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-37140-0_48

[20] K. Narukawa and T. Rodemann, "Examining the performance of evolutionary many-objective optimization algorithms on a real-world application," in *Sixth International Conference on Genetic and Evolutionary Computing*, Aug 2012, pp. 316–319.

[21] 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.

[22] Z. He, G. Yen, and J. Zhang, "Fuzzy-based pareto optimality for many-objective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 269–285, April 2014.

[23] S. Kukkonen and K. Deb, "Improved pruning of non-dominated solutions based on crowding distance for bi-objective optimization problems," in *IEEE Congress on Evolutionary Computation*, 2006, pp. 1179–1186.

[24] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, Aug 2014.

[25] L. While, P. Hingston, L. Barone, and S. Huband, "A faster algorithm for calculating hypervolume," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 29–38, Feb 2006.

[26] J. Bader and E. Zitzler, "Hype: An algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, Mar. 2011. [Online]. Available: http://dx.doi.org/10.1162/EVCO_a_00009

[27] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Computing Surveys (CSUR)*, vol. 48, no. 1, p. 13, 2015.

[28] J. J. Durillo, A. J. Nebro, C. A. C. Coello, J. Garcia-Nieto, F. Luna, and E. Alba, "A study of multiobjective metaheuristics when solving parameter scalable problems," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 4, pp. 618–635, Aug 2010.

[29] G. Yen and Z. He, "Performance metric ensemble for multiobjective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 131–144, Feb 2014.

[30] D. K. Saxena, J. A. Duro, A. Tiwari, K. Deb, and Q. Zhang, "Objective reduction in many-objective optimization: Linear and nonlinear algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 1, pp. 77–99, Feb. 2013. [Online]. Available: http://dx.doi.org/10.1109/TEVC.2012.2185847

[31] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining convergence and diversity in evolutionary multiobjective optimization," *Evolutionary Computation*, vol. 10, no. 3, pp. 263–282, Sep. 2002. [Online]. Available: http://dx.doi.org/10.1162/106365602760234108

[32] H. Sato, H. E. Aguirre, and K. Tanaka, "Controlling dominance area of solutions and its impact on the performance of moeas," in *Proceedings of the 4th International Conference on Evolutionary Multi-criterion Optimization*, ser. EMO'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 5–20. [Online]. Available: http://dl.acm.org/citation.cfm?id=1762545.1762552

[33] H. Sato, H. Aguirre, and K. Tanaka, "Self-controlling dominance area of solutions in evolutionary many-objective optimization," in *Simulated Evolution and Learning*, ser. Lecture Notes in Computer Science, K. Deb, A. Bhattacharya, N. Chakraborti, P. Chakroborty, S. Das, J. Dutta, S. Gupta, A. Jain, V. Aggarwal, J. Branke, S. Louis, and K. Tan, Eds. Springer Berlin Heidelberg, 2010, vol. 6457, pp. 455–465. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-17298-4_49

[34] M. Farina and P. Amato, "A fuzzy definition of optimality for many-criteria optimization problems," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 34, no. 3, pp. 315–326, May 2004.

[35] F. di Pierro, S.-T. Khu, and D. Savic, "An investigation on preference order ranking scheme for multiobjective evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 1, pp. 17–45, Feb 2007.

[36] X. Zhang, Y. Tian, and Y. Jin, "A knee point driven evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. PP, no. 99, pp. 1–1, 2014.

[37] M. Li, J. Zheng, R. Shen, K. Li, and Q. Yuan, "A grid-based fitness strategy for evolutionary many-objective optimization," in *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '10. New York, NY, USA: ACM, 2010, pp. 463–470. [Online]. Available: http://doi.acm.org/10.1145/1830483.1830570

[38] M. Koppen and K. Yoshida, "Substitute distance assignments in nsga-ii for handling many-objective optimization problems," in *Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science, S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, Eds. Springer Berlin Heidelberg, 2007, vol. 4403, pp. 727–741. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-70928-2_55

[39] L. While, L. Bradstreet, and L. Barone, "A fast way of calculating exact hypervolumes," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 1, pp. 86–95, Feb 2012.

[40] K. Bringmann and T. Friedrich, "An efficient algorithm for computing hypervolume contributions," *Evolutionary Computation*, p. 2010.

[41] E. Zitzler and S. Knzli, "Indicator-based selection in multiobjective search," in *Parallel Problem Solving from Nature - PPSN VIII*, ser. Lecture Notes in Computer Science, X. Yao, E. Burke, J. Lozano, J. Smith, J. Merelo-Guervs, J. Bullinaria, J. Rowe, P. Tio, A. Kabn, and H.-P. Schwefel, Eds. Springer Berlin Heidelberg, 2004, vol. 3242, pp. 832–842. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-30217-9_84

[42] A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh, "A survey of multiobjective evolutionary algorithms based on decomposition," *IEEE Transactions on Evolutionary Computation*, 2016.

[43] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, "Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes," *IEEE Transactions on Evolutionary Computation*, 2016.

[44] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 694–716, Oct 2015.

[45] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A new dominance relation-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 16–37, Feb 2016.

[46] S. Kukkonen and J. Lampinen, "Ranking-dominance and many-objective optimization," in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, Sept 2007, pp. 3983–3990.

[47] W. Barry and B. J. Ross, "Virtual photography using multi-objective particle swarm optimization," in *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '14. New York, NY, USA: ACM, 2014, pp. 285–292. [Online]. Available: http://doi.acm.org/10.1145/2576768.2598210

[48] B. Ross, "Evolution of stochastic bio-networks using summed rank strategies," in *IEEE Congress on Evolutionary Computation*, June 2011, pp. 773–780.

[49] P. Bentley and J. Wakefield, "Finding acceptable solutions in the pareto-optimal range using multiobjective genetic algorithms," in *Soft Computing in Engineering Design and Manufacturing*, P. Chawdhry, R. Roy, and R. Pant, Eds. Springer London, 1998, pp. 231–240. [Online]. Available: http://dx.doi.org/10.1007/978-1-4471-0427-8_25

[50] X. Zou, Y. Chen, M. Liu, and L. Kang, "A new evolutionary algorithm for solving many-objective optimization problems," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 38, no. 5, pp. 1402–1412, Oct 2008.

[51] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: John Wiley & Sons, Inc., 2001.

[52] A. B. de Carvalho and A. Pozo, "Measuring the convergence and diversity of {CDAS} multi-objective particle swarm optimization algorithms: A study of many-objective problems," *Neurocomputing*, vol. 75, no. 1, pp. 43 – 51, 2012, brazilian Symposium on Neural Networks (SBRN 2010)International Conference on Hybrid Artificial Intelligence Systems (HAIS 2010). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S092523121100405X

[53] J. Maltese, B. Ombuki-Berman, and A. P. Engelbrecht, "Pareto-based many-objective optimization using knee points," 2016, submitted to IEEE CEC.

[54] S. Zapotecas Martínez and C. A. Coello Coello, "A multi-objective particle swarm optimizer based on decomposition," in *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '11. New York, NY, USA: ACM, 2011, pp. 69–76. [Online]. Available: http://doi.acm.org/10.1145/2001576.2001587

[55] J. Bader, K. Deb, and E. Zitzler, "Faster hypervolume-based search using monte carlo sampling," in *Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems*, ser. Lecture Notes in Economics and Mathematical Systems, M. Ehrgott, B. Naujoks, T. J. Stewart, and J. Wallenius, Eds. Springer Berlin Heidelberg, 2010, vol. 634, pp. 313–326. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-04045-0_27

[56] R. Wang, Z. Zhou, H. Ishibuchi, T. Liao, and T. Zhang, "Localized weighted sum method for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, 2016.

[57] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, April 2009.

[58] I. Das and J. E. Dennis, "Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems," *SIAM Journal on Optimization*, vol. 8, no. 3, pp. 631–657, 1998. [Online]. Available: http://dx.doi.org/10.1137/S1052623496307510

[59] K. Deb, R. B. Agrawal, and R. B. Agrawal, "Simulated Binary Crossover for Continuous Search Space," 1995.

[60] K. Deb and M. Goyal, "A combined genetic adaptive search (geneas) for engineering design," *Computer Science and Informatics*, vol. 26, pp. 30–45, 1996.

[61] K. Li, K. Deb, Q. Zhang, and S. Kwong, "Efficient non-domination level update approach for steady-state evolutionary multiobjective optimization," *Department of Electrical and Computer Engineering, Michigan State University, East Lansing, USA, Tech. Rep. COIN Report*, no. 2014014, 2014.

[62] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*. London: Springer, 2005, ch. Scalable Test Problems for Evolutionary Multiobjective Optimization, pp. 105–145. [Online]. Available: http://dx.doi.org/10.1007/1-84628-137-7_6

[63] 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, Oct 2006.

[64] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *The Annals of Mathematical Statistics*, vol. 18, no. 1, pp. 50–60, 03 1947. [Online]. Available: http://dx.doi.org/10.1214/aoms/1177730491

[65] A. Engelbrecht, "Fitness function evaluations: A fair stopping condition?" in *IEEE Symposium on Swarm Intelligence*, Dec 2014, pp. 1–8.

[66] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "An efficient approach to nondominated sorting for evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 2, pp. 201–213, April 2015.

[67] A. Engelbrecht, "Particle swarm optimization: Velocity initialization," in *IEEE Congress on Evolutionary Computation*, 2012, pp. 1–8.

[68] M. Fleischer, "The measure of pareto optima applications to multi-objective metaheuristics," in *Proceedings of the 2nd International Conference on Evolutionary Multi-criterion Optimization*. Berlin, Heidelberg: Springer-Verlag, 2003, pp. 519–533. [Online]. Available: http://dl.acm.org/citation.cfm?id=1760102.1760146

[69] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, and V. da Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, April 2003.

[70] H. Ishibuchi, Y. Hitotsuyanagi, N. Tsukamoto, and Y. Nojima, *Many-Objective Test Problems to Visually Examine the Behavior of Multiobjective Evolution in a Decision Space*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 91–100. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-15871-1_10

[71] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler, "Theory of the hypervolume indicator: Optimal μ-distributions and the choice of the reference point," in *Proceedings of the Tenth ACM SIGEVO Workshop on Foundations of Genetic Algorithms*, ser. FOGA '09. New York, NY, USA: ACM, 2009, pp. 87–102. [Online]. Available: http://doi.acm.org/10.1145/1527125.1527138

[72] C. Goh and K. Tan, "An investigation on noisy environments in evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 3, pp. 354–381, June 2007.

[73] M. A. Potter and K. A. D. Jong, "A cooperative coevolutionary approach to function optimization," in *Proceedings of the International Conference on Evolutionary Computation: The Third Conference on Parallel Problem Solving from Nature*, ser. PPSN III. London, UK, UK: Springer-Verlag, 1994, pp. 249–257. [Online]. Available: http://dl.acm.org/citation.cfm?id=645822.670374

[74] J. Maltese, B. Ombuki-Berman, and A. P. Engelbrecht, "Co-operative vector-evaluated particle swarm optimization for multi-objective optimization," in *2015 IEEE Symposium Series on Computational Intelligence*, Dec 2015, pp. 1294–1301.
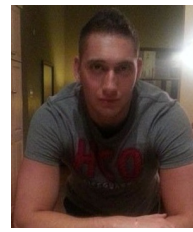
Justin Maltese Justin Maltese received the BSc and MSc degrees in Computer Science from Brock University, Ontario, Canada. He is currently running his own technology startup business. He is also applying computational intelligence paradigms to practical applications. He is a member of the Bio-Inspired Computational Intelligence Research Group at Brock University. His interests are evolutionary computation, swarm intelligence and neural networks with applications in challenging optimization problems, games and classification.

Beatrice M. Ombuki-Berman Dr. Beatrice Ombuki-Berman is a Professor in Computer Science at Brock University in Ontario, Canada. She received her Masters and PhD degrees in Information Engineering from University of the Ryukyus, Okinawa, Japan in 1998 and 2001, respectively. Her BSc degree in Double Mathematics and Computer Science (Minor) is from Jomo Kenyatta University of Agriculture and Technology, Nairobi, Kenya. She is the Co-founder/director of the Bio-Inspired Computational Intelligence Research Group at Brock University. Her research interests include computational intelligence paradigms, especially evolutionary computation and swarm intelligence, and applications to various classes of challenging optimization problems including real-world problems in logistics and complex networks.

Andries P. Engelbrecht Andries Engelbrecht received the Masters and PhD degrees in Computer Science from the University of Stellenbosch, South Africa, in 1994 and 1999 respectively. He is a Professor in Computer Science at the University of Pretoria, and serves as Head of the department. He also holds the position of South African Research Chair in Artificial Intelligence, and leads the Computational Intelligence Research Group at the University of Pretoria, consisting of 40 Masters and PhD students. His research interests include swarm intelligence, evolutionary computation, artificial neural networks, artificial immune systems, and the application of these Computational Intelligence paradigms to data mining, games, bioinformatics, finance, and difficult optimization problems. He has published over 220 papers in these fields in journals and international conference proceedings, and is the author of two books, Computational Intelligence: An Introduction and Fundamentals of Computational Swarm Intelligence.