2015 IEEE Symposium Series on Computational Intelligence

Co-operative Vector-Evaluated Particle Swarm Optimization for Multi-objective Optimization

Justin Maltese
Department of Computer Science
Brock University
St, Catharines, ON, Canada
jm10lh@brocku.ca

Beatrice M. Ombuki-Berman
Department of Computer Science
Brock University
St Catharines, ON, Canada
bombuki@brocku.ca

Andries. P. Engelbrecht
Department of Computer Science
University of Pretoria
Pretoria, South Africa
engel@cs.up.ac.za

Abstract*—Vector-evaluated particle swarm optimization is a particle swarm optimization variant which employs multiple swarms to solve multi-objective optimization problems. Recently, three variants of particle swarm optimization which utilize co-operative principles were shown to improve performance in single-objective environments. This work proposes co-operative vector-evaluated particle swarm optimization algorithms, which employ co-operative particle swarm optimization variants within vector-evaluated particle swarm optimization swarms. Performance of the proposed algorithms is compared with the standard vector-evaluated particle swarm optimization algorithm using various knowledge transfer strategies. A comparison of the best performing co-operative vector-evaluated particle swarm optimization variants is also made against well-known multi-objective PSO algorithms. Each co-operative vector-evaluated particle swarm optimization variant significantly outperforms standard vector-evaluated particle swarm optimization with respect to the hypervolume metric, with two of three variants also yielding improved solution distribution. The results indicate that co-operation is a powerful tool which enhances hypervolume and solution distribution of the original vector-evaluated particle swarm optimization algorithm, allowing co-operative vector-evaluated particle swarm optimization variants to successfully compete with top multi-objective PSO optimization algorithms.

I. Introduction

Many real-world optimization problems contain multiple (often conflicting) objectives to be dealt with simultaneously. Problems of this nature are commonly referred to as multi-objective problems (MOPs). One such algorithm which tackles MOPs is the vector-evaluated particle swarm optimization (VEPSO) algorithm [1], an extension of the heuristic-based particle swarm optimization (PSO) algorithm [2]. VEPSO utilizes a multi-swarm model, optimizing each objective individually using a dedicated swarm. However, each swarm also simultaneously optimizes the problem as a whole by passing information to other swarms using a knowledge transfer strategy (KTS).

In an attempt to improve performance of the original VEPSO algorithm, several researchers have experimented with modifying different aspects of VEPSO. Specifically, the following modifications have been made in previous work:

- Harrison *et al.* proposed additional KTSs [3] and investigated scalability as the number of objectives increases [4].
- Matthyssen *et al.* [5] addressed the issue of stagnation within VEPSO.
- Helbig and Engelbrecht adapted VEPSO to solve dynamic MOPs [6] and introduced a variant of VEPSO which utilized non-dominated solution guides [7].

One area which exhibits high potential for improvement is the optimization algorithm used by the individual swarms within VEPSO. Traditionally, each swarm uses the original PSO algorithm to perform optimization. Recently the performance of PSO has been improved for single-objective optimization by applying concepts inspired from cooperative systems [8]. These co-operative concepts deal with partitioning the search space into lower dimensional subspaces to avoid performance deterioration as search space dimensionality increases. This paper applies these co-operative principles to each swarm within VEPSO, attempting to enhance the performance of each individual PSO swarm and thus hopefully improving the overall performance of the VEPSO algorithm.

The remainder of this paper is organized as follows: Section II contains background information about multi-objective optimization and algorithms such as PSO, VEPSO and CPSO. Section III provides an explanation of how VEPSO with CPSO swarms is implemented. Section IV describes the experimental setup used in this study. Section V presents the results of all experiments performed, including analysis and discussion of observations. Finally, Section VI concludes the paper and suggests avenues for future research.

II. Background

This section provides an overview of all concepts pertaining to this paper. Topics covered are multi-objective optimization, PSO, CPSO variants, and VEPSO.

A. Multi-objective Optimization

Multi-objective optimization deals with simultaneously optimizing more than one objective. MOPs are especially challenging when objectives conflict with one another, requiring a series of trade-offs during optimization. A typical MOP can be formally expressed as

$$\begin{aligned}&\text{minimize } \vec{f}(\vec{x})\\&\text{subject to } \vec{x} \in [x_{min}, x_{max}]^{n_x}\end{aligned}$$

978-1-4799-7560-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SSCI.2015.185

IEEE computer society

where $\vec{f}(\vec{x}) = f_1(\vec{x}), f_2(\vec{x}), ..., f_{n_c}(\vec{x}))$, $\vec{x} = (x_1, x_2, ..., x_{n_x})$; n_x refers to the dimensionality of the search space and n_c represents the number of objectives. Solutions which cannot improve any objective further without worsening any other objective are extremely desirable for MOPs, referred to as Pareto optimal solutions. The set of Pareto optimal solutions, referred to as the Pareto front, is typically the end goal of a multi-objective optimizer.

B. Particle Swarm Optimization

PSO [2] is a stochastic metaheuristic optimization algorithm which simulates the flocking behavior of birds. A swarm of particles is used for optimization purposes, where each particle represents a candidate solution with respect to the problem being optimized. Particles iteratively improve solutions by moving towards the position of the best candidate solution found by the particle's neighbourhood and the best candidate solution found by the particle itself. Initial particle positions are generated randomly within the valid boundaries of the search space.

The PSO algorithm utilizes two update equations, which are:

$$S.\vec{v}_i(t+1) = \omega S.\vec{v}_i(t) + c_1\vec{r}_1(S.\vec{y}_i(t) - S.\vec{x}_i(t)) + c_2\vec{r}_2(S.\vec{\hat{y}}(t) - S.\vec{x}_i(t)) \quad (1)$$

$$S.\vec{x}_i(t+1) = S.\vec{x}_i(t) + S.\vec{v}_i(t+1) \quad (2)$$

where $S.\vec{v}_i$ corresponds to the velocity of particle i in swarm S, $S.\vec{x}$ references the current position of particle i in swarm S, $S.\vec{y}_i$ is the personal best position of particle i in swarm S, $S.\vec{\hat{y}}$ represents the global best (GBest) position of swarm S, $\vec{r}_1$ and $\vec{r}_2$ are vectors of random numbers sampled from a uniform distribution in the range [0,1], ω is the inertia weight, c_1 is the cognitive weight and c_2 is the social weight.

C. Co-operative Particle Swarm Optimization

The original PSO algorithm, along with many other stochastic optimization algorithms, suffers from a problem called the "curse of dimensionality" [9]. In general terms, the curse of dimensionality implies that algorithmic performance wanes as problem dimensionality increases. In an attempt to overcome the curse of dimensionality problem, Van den Bergh and Engelbrecht [10] proposed a new variant of PSO based on concepts observed in co-operative systems, referred to as the co-operative PSO (CPSO). This variant later became known as co-operative split PSO (CPSO-S).

CPSO-S improves the PSO algorithm by partitioning the search space into lower dimensional subspaces. This partioning is performed by splitting the single swarm which is attempting to optimize a vector of n_x dimensions into n_x subswarms each optimizing a single dimension. In order to evaluate the quality of candidate solutions, CPSO-S maintains a vector whose components consists of the global best positions of each subswarm respectively, referred to as a *context vector* [11].

Although CPSO-S was empirically determined to perform well [12], it was found that the algorithm suffered performance degradation in the case of dependent decision variables due to being erroneously considered in isolation. To reduce the impact of the variable dependence problem, Van den Bergh and Engelbrecht [12] introduced a variant of CPSO-S named CPSO-S_K. CPSO-S_K is similar to CPSO-S, however CPSO-S_K splits the n_x-dimensional search space into k parts arbitrarily. Each swarm therefore optimizes $\frac{n_x}{k}$ dimensions with the hope that related dimensions are optimized together.

CPSO-S and CPSO-S_K both experience a stagnation problem in the presence of deceptive functions, a problem not present in the original PSO algorithm [10]. These algorithms are susceptible to becoming stuck in *pseudo-optima* - locations which are not locally or globally optimal. To address this problem, the CPSO-H_K hybrid algorithm [10] was proposed which executes CPSO-S_K and regular PSO sequentially. Note that CPSO-H_K uses a KTS to exchange information between the CPSO-S_K and PSO algorithm. Knowledge transfer is done by simply injecting the best solution from CPSO-S_K into the PSO swarm and vice-versa.

D. Vector Evaluated Particle Swarm Optimization

Parsopoulos and Vrahatis [1] proposed the vector-evaluated PSO (VEPSO) algorithm as a multi-objective extension to the original PSO algorithm. To solve MOPs, VEPSO employs a multi-swarm model where each swarm is tasked with optimizing a single objective. The problem as a whole is optimized through the use of a KTS, as swarms pass optimization information among themselves. VEPSO KTSs each uniquely modify the choice of global guide for each swarm, where a global guide is a particle whose dimensions are used in equation (3) instead of $\vec{\hat{y}}$. VEPSO KTSs introduced in previous literature are the ring KTS [1], random KTS [13], parent-centric crossover (PCX) archive KTS [3] and the PCX GBest KTS [3].

VEPSO uses the concept of Pareto domination and saves non-dominated solutions within a structure referred to as the *archive*. A size limit is defined for the archive prior to optimization. Previous work [14] has demonstrated that larger archive sizes increase accuracy while smaller archive sizes yield better stability. Particle fitness is evaluated as a vector of sub-objective fitnesses rather than a single scalar value.

III. Co-operative Vector-Evaluated Particle Swarm Optimization

This section proposes the co-operative VEPSO (CVEPSO) algorithm and its variants. Implementation details such as knowledge transfer and archive addition/maintenance are addressed.

A. Co-operation Incorporation

Co-operative principles are incorporated into VEPSO to produce CVEPSO by simply replacing the standard PSO used by each swarm of VEPSO with a CPSO variant. The following algorithms are proposed as variants of CVEPSO:

- **CVEPSO-S**: VEPSO which utilizes CPSO-S subswarms.
- **CVEPSO-S_K**: VEPSO which utilizes CPSO-S_K subswarms.
- **CVEPSO-H_K**: VEPSO which utilizes CPSO-H_K subswarms.

B. Knowledge Transfer

Using a CVEPSO variant does not affect the knowledge transfer method. Knowledge flow between swarms (with respect to the chosen KTS) remains the same irrespective of whether PSO or a CPSO variant is used. The ring, random and PCX GBest KTSs require a "global best" particle to determine the global guides. Because there is no global best position in the CPSO variants, these KTSs use the context vector instead of a global best particle. Figure 1 exemplifies knowledge flow of the random KTS via utilization of the context vector.

C. Archive Addition and Maintenance

Whenever a context vector of a CVEPSO swarm is non-dominated with reference to all current solutions in the archive, that context vector is added into the archive. If a non-dominated context vector is found while the archive is full, the context vector is added to the archive and a removal strategy is executed to return the archive to its size limit. It is desirable to select removal strategies which promote solution diversity within the archive, however, random strategies are also possible.

IV. Experimental Setup

This section details the methods and components used in performing experimentation. Topics covered include performance measures, statistical analysis methods, algorithm parameters and benchmark functions.

A. Performance Measures

The performance measures used in this work aim to provide a fair assessment of each algorithm without assuming a known Pareto front. Each measure used is based solely on the obtained approximation front, as described below.

Hypervolume: The hypervolume indicator [15] is a scalar metric which measures the hypervolume of the objective space that is weakly dominated by an approximation set. To maximize hypervolume, the solution set must solely consist of Pareto-optimal points. Hypervolume calculation has been shown to be an NP-hard problem [16], taking exponential time in the number of objectives.

Solution Distribution: Introduced in [17], the solution distribution metric measures the spacing density of a given set of solutions. Minimizing the solution distribution metric corresponds to a more desirable solution spread.

B. Statistical Methods

The Mann-Whitney-Wilcoxon rank sum test [18] was used in a pairwise fashion to check for statistical significance in all experiments. In the case of a statistically significant difference, the algorithm with the higher mean over 30 independent runs was given a win and the algorithm with the lower mean was given a loss. A confidence level of 95% was used for each test.

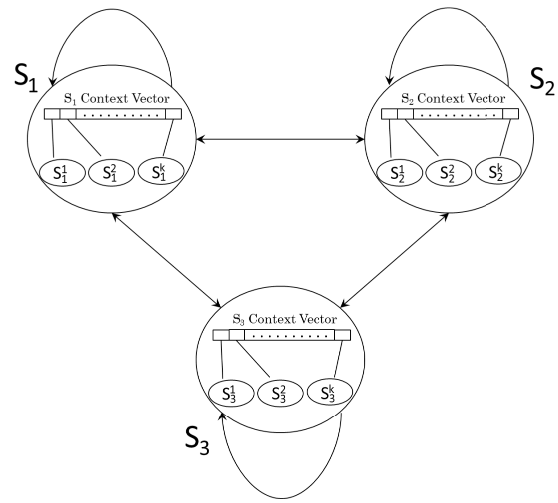

Fig. 1. Visualization of the random KTS knowledge transfer flow using context vectors. The example illustrates CVEPSO for a three objective problem.

C. Particle Swarm Optimization Parameters

Each PSO swarm was created with 50 particles. The initial velocity of these particles was set to zero. To ensure convergent weight values [19], ω was set to 0.729844, c_1 was set to 1.496180 and c_2 was set to 1.496180. Particles were re-initialized randomly within the legal bounds of the search space in the case where a boundary constraint was violated.

D. Co-operative Particle Swarm Optimization Parameters

To ensure an unbiased comparison between PSO and the CPSO-S and CPSO-S_K variants, each CVEPSO swarm was assigned the same number of particles, i.e 50, before performing splitting into sub-swarms. When the sub-swarm split was performed based on the CPSO variant, the particles were divided as evenly as possible across all sub-swarms. This ensured an identical number of fitness function evaluations for each algorithm. The weight values, initial velocity values, boundary clamping techniques and synchronous update methodologies were identical to those of PSO, described in Section IV.D.

The k value of CPSO-S_K was set to 6 for all experiments, motivated by previous work in [12] and validated empirically. The CPSO-S_K swarm within CPSO-H_K also used a k value of 6.

E. Vector Evaluated Particle Swarm Optimization Parameters

The number of swarms used in the VEPSO and CVEPSO algorithms were equal to the number of objectives required for a given optimization problem. The size limit of the archive was set to 250 solutions. When inserting a solution into a full archive, a solution was removed according to a distance-based strategy, promoting archive diversity. Distance-based removal selected the two solutions with the smallest distance between each other and removed one of them randomly.

F. Benchmark Suites

Various benchmark suites exist to test the performance of multi-objective optimizers. One such suite, the Walking Fish Group (WFG) suite [20], provides a set of problems that are defined in terms of a vector of parameters. Various transition and shape functions are applied to the WFG problem set to produce complex Pareto fronts. The WFG functions provide a wide variety of shapes and modalities, incorporating unique difficulties such as deception, degeneracy and disconnection.

Each WFG function in this work uses an identical parameter set, consisting of three objectives and a total of 24 decision parameters. The decision parameters contain 20 position-related parameters and four distance-related parameters. A complete overview of WFG suite problems can be found in [20], [21].

V. Experimental Results and Discussion

This section presents and discusses results obtained from the experiments performed. Section V.A consists of experiments comparing the performance of each CVEPSO variant using four distinct KTSs. In Section V.B, experiments comparing the performance of the two best CVEPSO variants combinations against other well known multi-objective PSO algorithms are presented and discussed.

A. Algorithm Performance

The performance of each algorithm was compared with the objective of establishing which algorithms, if any, performed significantly better over each of the functions.

1) Ring Knowledge Transfer Strategy: The performance of the VEPSO, CVEPSO-S, CVEPSO-S_K and CVEPSO-H_K algorithms was compared over the nine WFG functions using the ring KTS. Table II presents the results of these experiments for both the hypervolume and distribution metric, while Table III displays the minimum, maximum and mean rank over all functions for each algorithm in Table II.

VEPSO yielded extremely poor hypervolume performance in comparison to the other algorithms, because VEPSO recorded one or more losses over each function. Thus, it can be deduced that at least one CVEPSO variant produced a better hypervolume over VEPSO for each WFG function when the ring KTS was used. Concerning the performance of the CVEPSO variants, CVEPSO-S_K outperformed CVEPSO-H_K for the hypervolume metric. However, both algorithms experienced considerably worse hypervolume rankings over the WFG functions on average in comparison to CVEPSO-S.

With regards to the distribution metric, CVEPSO-S_K obtained the lowest mean rank. CVEPSO-H_K distributed non-dominated solutions better than CVEPSO-S_K, but worse than CVEPSO-S and VEPSO on average. Note that CVEPSO distributed non-dominated solutions significantly better than all three CVEPSO variants for WFG1 and WFG3. Thus it cannot be concluded that any one of the CVEPSO variants produced a better non-dominated solution spread than VEPSO for any function.

Overall, notable observations were that CVEPSO-S was undoubtedly the top performing algorithm. CVEPSO-S was ranked first for eight of nine functions with respect to both metrics. VEPSO tended to produce a poor hypervolume with a desirable solution spread, while CVEPSO-S_K was the opposite.

TABLE I. Mann-Whitney Wins and Losses For Ring KTS

Algorithm	Metric	Result	WFG Function 1	2	3	4	5	6	7	8	9
VEPSO	Hypervolume	Wins	0	0	0	0	0	0	0	0	1
		Losses	3	3	3	3	2	3	3	3	1
		Difference	-3	-3	-3	-3	-2	-3	-3	-3	0
		Rank	4	4	4	4	3	4	4	4	2
	Distribution	Wins	3	1	2	1	2	1	0	2	0
		Losses	0	1	0	1	1	1	1	1	1
		Difference	+3	0	+2	0	+1	0	-1	+1	-1
		Rank	1	2	1	2	2	2	2	2	3
CVEPSO-S	Hypervolume	Wins	1	2	3	3	3	3	3	3	3
		Losses	2	0	0	0	0	0	0	0	0
		Difference	-1	+2	+3	+3	+3	+3	+3	+3	+3
		Rank	3	1	1	1	1	1	1	1	1
	Distribution	Wins	0	3	2	3	3	3	3	3	2
		Losses	3	0	0	0	0	0	0	0	0
		Difference	-3	+3	+2	+3	+3	+3	+3	+3	+2
		Rank	4	1	1	1	1	1	1	1	1
CVEPSO-S_K	Hypervolume	Wins	2	2	2	2	2	2	2	2	1
		Losses	0	0	1	1	1	1	1	1	1
		Difference	+2	+2	+1	+1	+1	+1	+1	+1	0
		Rank	1	1	2	2	2	2	2	2	2
	Distribution	Wins	1	1	0	0	1	0	0	0	0
		Losses	2	1	3	3	2	3	1	2	2
		Difference	-1	0	-3	-3	-1	-3	-1	-2	-2
		Rank	3	2	4	4	3	4	2	3	4
CVEPSO-H_K	Hypervolume	Wins	2	1	1	1	0	1	1	1	0
		Losses	0	2	2	2	2	2	2	2	3
		Difference	+2	-1	-1	-1	-2	-1	-1	-1	-3
		Rank	1	3	3	3	3	3	3	3	4
	Distribution	Wins	2	0	1	1	0	1	0	0	1
		Losses	1	3	2	1	3	1	1	2	0
		Difference	+1	-3	-1	0	-3	0	-1	-2	+1
		Rank	2	4	3	2	4	2	2	3	2

TABLE II. Algorithm Rank Summary For Ring KTS

Metric	Measure	Algorithm VEPSO	CVEPSO-S	CVEPSO-S_K	CVEPSO-H_K
Hypervolume	Mean	3.667	1.222	1.778	2.889
	Maximum	4	3	2	4
	Minimum	2	1	1	1
Distribution	Mean	1.889	1.333	3.222	2.667
	Maximum	3	4	4	4
	Minimum	1	1	2	2

2) Random Knowledge Transfer Strategy: Table IV summarizes results obtained using the random KTS. Table V presents a data summary of the rank over all functions for each algorithm.

Table IV shows that VEPSO again produced subpar performance in comparison to the CVEPSO variants with respect to hypervolume. The CVEPSO-S_K algorithm ranked better than both CVEPSO-H_K and VEPSO on average, ranking first overall for three out of nine functions and second for the remaining WFG functions. CVEPSO-S produced significantly better hypervolumes over all other algorithms and yielded the best mean rank as seen in Table V.

Concerning the distribution metric, VEPSO performed notably consistently, as it was ranked second for seven of nine functions. This observation suggests that VEPSO produced a better non-dominated solution spread than CVEPSO-S_K and CVEPSO-H_K, yet worse than CVEPSO-S. CVEPSO-S_K did not distribute non-dominated solutions well, performing worse than all other algorithms for WFG4 and WFG6.

Another observation present in Table IV is that the CVEPSO-S algorithm again ranked better for every function except WFG1, where performance degradation was severe. It is observed that WFG1 is the only function with a strictly convex shape, suggesting that CVEPSO-S has severe difficulty when

TABLE III. MANN-WHITNEY WINS AND LOSSES FOR RANDOM KTS

Algorithm	Metric	Result	WFG Function 1	2	3	4	5	6	7	8	9
VEPSO	Hypervolume	Wins	1	0	0	0	0	0	0	0	1
		Losses	2	3	3	3	3	3	3	2	2
		Difference	-1	-3	-3	-3	-3	-3	-3	-2	-1
		Rank	3	4	4	4	4	4	4	3	3
	Distribution	Wins	1	1	2	1	1	1	0	2	0
		Losses	1	2	1	1	1	1	1	1	0
		Difference	0	-1	+1	0	0	0	-1	+1	0
		Rank	3	3	2	2	2	2	2	2	2
CVEPSO-S	Hypervolume	Wins	0	2	3	3	3	3	2	3	3
		Losses	3	0	0	0	0	0	0	0	0
		Difference	-3	+2	+3	+3	+3	+3	+2	+3	+3
		Rank	4	1	1	1	1	1	1	1	1
	Distribution	Wins	0	3	3	3	3	3	3	3	2
		Losses	3	0	0	0	0	0	0	0	0
		Difference	-3	+3	+3	+3	+3	+3	+3	+3	+2
		Rank	4	1	1	1	1	1	1	1	1
CVEPSO-S_K	Hypervolume	Wins	3	2	2	2	2	2	2	2	2
		Losses	0	0	1	1	1	1	0	1	1
		Difference	+3	+2	+1	+1	+1	+1	+2	+1	+1
		Rank	1	1	2	2	2	2	1	2	2
	Distribution	Wins	1	2	0	0	1	0	0	0	0
		Losses	0	1	2	3	1	3	1	2	1
		Difference	+1	+1	-2	-3	0	-3	-1	-2	-1
		Rank	2	2	3	4	2	4	2	3	3
CVEPSO-H_K	Hypervolume	Wins	2	1	1	1	1	1	1	0	0
		Losses	1	2	2	2	2	2	2	2	3
		Difference	+1	-1	-1	-1	-1	-1	-1	-2	-3
		Rank	2	3	3	3	3	3	3	3	4
	Distribution	Wins	2	0	0	1	0	1	0	0	0
		Losses	0	3	2	1	3	1	1	2	1
		Difference	+2	-3	-2	0	-3	0	-1	-2	-1
		Rank	1	4	3	2	4	2	2	3	3

TABLE IV. ALGORITHM RANK SUMMARY FOR RANDOM KTS

Metric	Measure	Algorithm VEPSO	CVEPSO-S	CVEPSO-S_K	CVEPSO-H_K
Hypervolume	Mean	3.667	1.333	1.667	3.000
	Maximum	4	4	2	4
	Minimum	3	1	1	2
Distribution	Mean	2.222	1.333	2.778	2.667
	Maximum	3	4	4	4
	Minimum	2	1	2	1

TABLE V. MANN-WHITNEY WINS AND LOSSES FOR PCX GBEST KTS

Algorithm	Metric	Result	WFG Function 1	2	3	4	5	6	7	8	9
VEPSO	Hypervolume	Wins	0	0	0	0	0	0	0	0	1
		Losses	2	3	3	3	3	3	3	3	2
		Difference	-2	-3	-3	-3	-3	-3	-3	-3	-1
		Rank	3	4	4	4	4	4	4	4	3
	Distribution	Wins	2	1	2	0	2	0	1	2	0
		Losses	0	2	1	1	1	1	1	1	0
		Difference	+2	-1	+1	-1	+1	-1	0	+1	0
		Rank	1	3	2	3	2	2	2	2	2
CVEPSO-S	Hypervolume	Wins	0	2	3	3	3	3	2	3	3
		Losses	2	1	0	0	0	0	0	0	0
		Difference	-2	+1	+3	+3	+3	+3	+2	+3	+3
		Rank	3	2	1	1	1	1	1	1	1
	Distribution	Wins	0	3	3	3	3	3	3	3	1
		Losses	3	0	0	0	0	0	0	0	0
		Difference	-3	+3	+3	+3	+3	+3	+3	+3	+1
		Rank	4	1	1	1	1	1	1	1	1
CVEPSO-S_K	Hypervolume	Wins	3	3	2	2	2	2	2	2	2
		Losses	0	0	1	1	1	1	0	1	1
		Difference	+3	+3	+1	+1	+1	+1	+2	+1	+1
		Rank	1	1	2	2	2	2	1	2	2
	Distribution	Wins	1	2	0	0	1	0	0	0	0
		Losses	2	1	2	2	2	1	1	3	1
		Difference	-1	+1	-2	-2	-1	-1	-1	-3	-1
		Rank	3	2	3	4	3	2	3	4	4
CVEPSO-H_K	Hypervolume	Wins	2	1	1	1	1	1	1	1	0
		Losses	1	2	2	2	2	2	2	2	3
		Difference	+1	-1	-1	-1	-1	-1	-1	-1	-3
		Rank	2	3	3	3	3	3	3	3	4
	Distribution	Wins	2	0	0	1	0	0	0	1	0
		Losses	0	3	2	1	3	1	2	2	0
		Difference	+2	-3	-2	0	-3	-1	-2	-1	0
		Rank	1	4	3	2	4	2	4	3	2

TABLE VI. ALGORITHM RANK SUMMARY FOR PCX GBEST KTS

Metric	Measure	Algorithm VEPSO	CVEPSO-S	CVEPSO-S_K	CVEPSO-H_K
Hypervolume	Mean	3.778	1.333	1.667	3.000
	Maximum	4	3	2	4
	Minimum	3	1	1	2
Distribution	Mean	2.111	1.333	3.111	2.778
	Maximum	3	4	4	4
	Minimum	1	1	2	1

presented with this type of landscape. Note that CVEPSO-H_K performed worse than the other co-operative algorithms for the deceptive WFG5 and WFG9 functions with respect to both metrics. This is an interesting observation, since the CPSO-H_K algorithm was designed primarily to improve performance of CPSO-S and CPSO-S_K in the presence of deception yet does not seem to do so for the WFG functions.

3) PCX GBest Knowledge Transfer Strategy: Table VI shows results for the algorithms using the PCX GBest KTS, with Table VII displaying the mean, maximum and minimum algorithm ranks over all functions in Table VI. Observations drawn from the results of these experiments are nearly identical to those of the ring and random KTSs.

In terms of hypervolume, CVEPSO-H_K had an exact mean rank of three, performing better than CVEPSO but worse than both CVEPSO-S and CVEPSO-S_K. It is observed from Table VII that CVEPSO-S_K had a worse mean rank than CVEPSO-S. However, it should be noted that CVEPSO-S_K was able to outperform CVEPSO-S for WFG1, WFG2 and WFG7.

Table VII also presents the non-dominated solution distribution results of each algorithm. Similar to previous observations for the ring and random KTSs, VEPSO distributed its non-dominated solutions better than CVEPSO-S_K and CVEPSO-H_K. CVEPSO-S again obtained a better distribution of non-dominated solutions than all other algorithms on average over the WFG functions.

With regards to overall performance, CVEPSO-S was dominant over all other algorithms. The algorithm consistently ranked first, struggling only with WFG1. CVEPSO-H_K earned its best rankings on strictly convex functions such as WFG1 for both metrics, outperforming both CVEPSO-S and CVEPSO-S_K.

4) PCX Archive Knowledge Transfer Strategy: Performance while using the PCX Archive KTS was tested for each algorithm, detailed in Table VIII. Table IX presents the mean, maximum and minimum rank over all WFG functions for each algorithm.

Table VIII reveals that VEPSO again experienced subpar hypervolume performance in comparison to the other algorithms. The algorithm had a negative difference score for all functions, consistently yielding more losses than wins. Since this observation was present for all other KTSs thus far, it is concluded that CPSO and its variants yield greater hypervol-

TABLE VII. MANN-WHITNEY WINS AND LOSSES FOR PCX ARCHIVE KTS

Algorithm	Metric	Result	WFG Function 1	2	3	4	5	6	7	8	9
VEPSO	Hypervolume	Wins	1	0	0	0	0	0	0	0	0
		Losses	2	3	2	3	2	2	3	2	2
		Difference	-1	-3	-2	-3	-2	-2	-3	-2	-2
		Rank	3	4	3	4	3	3	4	3	3
	Distribution	Wins	1	2	1	0	0	1	0	0	1
		Losses	0	0	1	1	1	0	1	1	0
		Difference	+1	+2	0	-1	-1	+1	-1	-1	+1
		Rank	2	1	2	2	2	1	2	2	1
CVEPSO-S	Hypervolume	Wins	0	2	3	3	3	3	3	3	3
		Losses	3	0	0	0	0	0	0	0	0
		Difference	-3	+2	+3	+3	+3	+3	+3	+3	+3
		Rank	4	1	1	1	1	1	1	1	1
	Distribution	Wins	0	0	3	3	3	1	3	3	1
		Losses	3	0	0	0	0	0	0	0	1
		Difference	-3	0	+3	+3	+3	+1	+3	+3	0
		Rank	4	2	1	1	1	1	1	1	3
CVEPSO-S_K	Hypervolume	Wins	3	2	2	2	2	2	2	2	2
		Losses	0	0	1	1	1	1	1	1	1
		Difference	+3	+2	+1	+1	+1	+1	+1	+1	+1
		Rank	1	1	2	2	2	2	2	2	2
	Distribution	Wins	1	0	0	0	0	0	0	0	0
		Losses	1	2	3	1	1	3	1	1	2
		Difference	0	-2	-3	-1	-1	-3	-1	-1	-2
		Rank	3	4	4	2	2	4	2	2	4
CVEPSO-H_K	Hypervolume	Wins	2	1	0	1	0	0	1	0	0
		Losses	1	2	2	2	2	2	2	2	2
		Difference	+1	-1	-2	-1	-2	-2	-1	-2	-2
		Rank	2	3	3	3	3	3	3	3	3
	Distribution	Wins	2	1	1	0	0	1	0	0	1
		Losses	0	1	1	1	1	0	1	1	0
		Difference	+2	0	0	-1	-1	+1	-1	-1	+1
		Rank	1	2	2	2	2	1	2	2	1

TABLE VIII. ALGORITHM RANK SUMMARY FOR PCX ARCHIVE KTS

Metric	Measure	Algorithm VEPSO	CVEPSO-S	CVEPSO-S_K	CVEPSO-H_K
Hypervolume	Mean	3.333	1.333	1.778	2.889
	Maximum	4	4	2	3
	Minimum	3	1	1	2
Distribution	Mean	1.667	1.667	3.000	1.667
	Maximum	2	4	4	2
	Minimum	1	1	2	1

ume performance over VEPSO in general, independent of the KTS used.

One noticeable difference from previous results is present with respect to the non-dominated solution distribution performances of CVEPSO-H_K. The algorithm yielded a very desirable non-dominated solution distribution for all functions, ranking first and second consistently. From this, it can be stated that CVEPSO-H_K works well with the PCX Archive KTS. Another observation is that CVEPSO-S did not dominate the distribution metric for the PCX Archive KTS as it did for the other KTSs. CVEPSO-S achieved the same mean non-dominated solution distribution rank as VEPSO and CVEPSO-H_K.

Overall, CVEPSO-S ranked best with respect to both non-dominated solution distribution and hypervolume for all KTSs. VEPSO typically experienced significantly lower hypervolume for all KTSs in comparison to CVEPSO-S_K and CVEPSO-H_K. However, because CVEPSO-S_K and CVEPSO-H_K produced a worse non-dominated solution spread in comparison to VEPSO, the utilization of these algorithms is seen as more of a trade-off than a strict improvement. This does not apply for CVEPSO-S, as its consistent domination of both metrics over VEPSO allows one to conclude that it is a definite improvement over VEPSO.

B. Comparison to other Algorithms

While it has been concluded that the CVEPSO variants improve VEPSO performance, it is necessary to compare the performance of the CVEPSO variants with that of other well-known multi-objective PSO optimization algorithms. The two best CVEPSO variants found in Section V.A, CVEPSO-S and CVEPSO-S_K, were compared with optimized multi-objective PSO (oMOPSO) [22] and speed constrained multi-objective PSO (SMPSO) [23]. The PCX GBest KTS is used for knowledge transfer purposes as it was empirically found to work best for the chosen CVEPSO variants. Experiments were performed over all of the WFG functions for both three and five objectives. An equal number of function iterations were used across all algorithms to avoid any bias.

1) Performance in 3-D Objective Space: Table XVI presents the results for each algorithm for the three-objective instances of the benchmark functions. Table XVII presents the mean, maximum and minimum algorithm ranks for the experiments performed in Table XVI.

Both oMOPSO and SMPSO exhibited better hypervolume performance on average over the WFG functions compared to CVEPSO-S and CVEPSO-S_K. CVEPSO-S had a slightly worse mean algorithm rank in comparison to SMPSO, observable in Table XVI. However, CVEPSO-S performed best for WFG4, WFG5 and WFG6, each of which are functions with a simple concave shape. Together, these functions contain all modalities present within the WFG suite, which are unimodal, multimodal and deceptive. Each of these modalities are contained within other WFG functions which CVEPSO-S performed poorly on, suggesting that CVEPSO-S is fairly insensitive to function modality. CVEPSO-S_K experienced the worst hypervolume performance overall, as it was never the best performing algorithm for any of the functions.

The distribution performance of CVEPSO-S_K was especially bad, as the algorithm ranked worst for every function other than WFG1. This is not surprising, as CVEPSO-S_K was selected as a top CVEPSO-KTS combination for its superb hypervolume performance rather than its distribution rankings in Section V.A. oMOPSO distributed very well, yielding the best rankings on average over the WFG functions followed by SMPSO. However, both oMOPSO and SMPSO were outperformed by CVEPSO-S for WFG4, WFG5, WFG6, WFG8 and WFG9 with reference to the solution distribution metric.

These observations lead to the conclusion that the CVEPSO-S algorithm exhibits the potential to compete with both oMOPSO and SMPSO in 3-D objective space, especially with regards to the distribution metric in which it outperformed both algorithms on five of nine functions. This conclusion is not valid for CVEPSO-S_K, as it was completely dominated by oMOPSO and SMPSO when three objectives were present.

2) Performance in 5-D Objective Space: Table XVIII summarizes the performance of each algorithm with regards to the five-objective instances of the benchmark functions. Table XIX presents the mean, maximum and minimum algorithm rank over the nine WFG functions in 5-D objective space.

TABLE IX. MANN-WHITNEY WINS AND LOSSES IN 3-D OBJECTIVE SPACE

Algorithm	Metric	Result	WFG Function 1	2	3	4	5	6	7	8	9
CVEPSO-S	Hypervolume	Wins	0	0	1	3	3	2	1	1	1
		Losses	3	3	2	0	0	0	1	2	2
		Difference	-3	-3	-1	+3	+3	+2	0	-1	-1
		Rank	4	4	3	1	1	1	3	3	3
	Distribution	Wins	0	1	1	3	3	2	3	3	1
		Losses	3	2	2	0	0	0	0	0	2
		Difference	-3	-1	-1	+3	+3	+2	+3	+3	-1
		Rank	4	3	3	1	1	1	1	1	3
CVEPSO-S_K	Hypervolume	Wins	1	1	0	1	0	0	1	0	0
		Losses	2	2	3	1	3	3	0	3	3
		Difference	-1	-1	-3	0	-3	-3	+1	-3	-3
		Rank	3	3	4	2	4	4	2	4	4
	Distribution	Wins	1	0	0	0	0	0	0	0	0
		Losses	2	3	3	3	3	3	3	3	3
		Difference	-1	-3	-3	-3	-3	-3	-3	-3	-3
		Rank	3	4	4	4	4	4	4	4	4
oMOPSO	Hypervolume	Wins	2	3	3	1	1	2	2	2	3
		Losses	0	0	0	1	1	0	0	1	0
		Difference	+2	+3	+3	0	0	+2	+2	+1	+3
		Rank	1	1	1	2	2	1	1	2	1
	Distribution	Wins	3	2	2	2	1	2	1	1	2
		Losses	0	0	0	1	1	0	1	1	0
		Difference	+3	+2	+2	+1	0	+2	0	0	+2
		Rank	1	1	1	2	2	1	2	2	1
SMPSO	Hypervolume	Wins	2	2	2	0	1	1	0	3	2
		Losses	0	1	1	3	1	2	3	0	1
		Difference	+2	+1	+1	-3	0	-1	-3	+3	+1
		Rank	1	2	2	4	2	3	4	1	2
	Distribution	Wins	2	2	2	1	1	1	1	1	2
		Losses	1	0	0	2	1	2	1	1	0
		Difference	+1	+2	+2	-1	0	-1	0	0	+2
		Rank	2	1	1	3	2	3	2	2	1

TABLE X. ALGORITHM RANK SUMMARY FOR 3-D OBJECTIVE SPACE

Metric	Measure	Algorithm CVEPSO-S	CVEPSO-S_K	oMOPSO	SMPSO
Hypervolume	Mean	2.556	3.333	1.333	2.333
	Maximum	4	4	2	4
	Minimum	1	2	1	1
Distribution	Mean	2.000	3.889	1.444	1.889
	Maximum	4	4	2	3
	Minimum	1	3	1	1

TABLE XI. MANN-WHITNEY WINS AND LOSSES IN 5-D OBJECTIVE SPACE

Algorithm	Metric	Result	WFG Function 1	2	3	4	5	6	7	8	9
CVEPSO-S	Hypervolume	Wins	0	0	0	2	2	2	2	2	3
		Losses	3	3	3	1	1	0	1	1	0
		Difference	-3	-3	-3	+1	+1	+2	+1	+1	+3
		Rank	4	4	4	2	2	1	2	2	1
	Distribution	Wins	0	1	2	3	3	3	1	3	3
		Losses	3	2	1	0	0	0	0	0	0
		Difference	-3	-1	+1	+3	+3	+3	+1	+3	+3
		Rank	4	3	2	1	1	1	2	1	1
CVEPSO-S_K	Hypervolume	Wins	1	3	1	3	3	2	3	3	0
		Losses	2	0	2	0	0	0	0	0	2
		Difference	-1	+3	-1	+3	+3	+2	+3	+3	0
		Rank	3	1	3	1	1	1	1	1	2
	Distribution	Wins	1	0	3	0	0	0	0	0	0
		Losses	2	3	0	3	3	3	3	3	3
		Difference	-1	-3	+3	-3	-3	-3	-3	-3	-3
		Rank	3	4	1	4	4	4	4	4	4
oMOPSO	Hypervolume	Wins	2	1	2	0	1	0	0	1	1
		Losses	0	1	0	3	2	3	3	2	2
		Difference	+2	0	+2	-3	-1	-3	-3	-1	-1
		Rank	1	2	1	4	3	4	4	3	3
	Distribution	Wins	3	2	0	1	1	2	2	1	1
		Losses	0	0	2	1	2	1	0	1	1
		Difference	+3	+2	-2	0	-1	+1	+2	0	0
		Rank	1	1	3	2	3	2	1	2	2
SMPSO	Hypervolume	Wins	2	1	2	1	0	1	1	0	1
		Losses	0	1	0	2	3	2	2	3	1
		Difference	+2	0	+2	-1	-3	-1	-1	-3	0
		Rank	1	2	1	3	4	3	3	4	2
	Distribution	Wins	2	2	0	1	2	1	1	1	1
		Losses	1	0	2	1	1	2	1	1	1
		Difference	+1	+2	-2	0	+1	-1	0	0	0
		Rank	2	1	3	2	2	3	3	2	2

TABLE XII. ALGORITHM RANK SUMMARY FOR 5-D OBJECTIVE SPACE

Metric	Measure	Algorithm CVEPSO-S	CVEPSO-S_K	oMOPSO	SMPSO
Hypervolume	Mean	2.444	1.556	2.778	2.556
	Maximum	4	3	4	4
	Minimum	1	1	1	1
Distribution	Mean	1.778	3.556	1.889	2.222
	Maximum	4	4	3	3
	Minimum	1	1	1	1

The hypervolume performance of CVEPSO-S_K was greatly improved in comparison to when only three objectives were used. This suggests that CVEPSO-S_K scales very well with respect to the hypervolume metric. Both CVEPSO-S_K and CVEPSO-S yielded better hypervolume scalability than SMPSO and oMOPSO. oMOPSO scaled worst, as its mean hypervolume rank more than doubled in comparison to its 3-D objective space performance. Note that SMPSO scaled much better than oMOPSO with respect to hypervolume.

The dominant hypervolume performance of CVEPSO-S_K was offset by its poor non-dominated solution distribution performance. CVEPSO-S_K ranked worst for six out of nine functions, yielding a worse non-dominated solution distribution than CVEPSO-S, oMOPSO and SMPSO for nearly every function. CVEPSO-S yielded the best distribution rankings on average over the WFG functions, seen in Table XIX. CVEPSO-S earned more wins than losses on seven out of nine functions, scaling well with regards to the distribution metric. oMOPSO and SMPSO both had worse mean distribution rankings in comparison to CVEPSO-S.

Overall, CVEPSO-S scaled very well in comparison to oMOPSO and SMPSO. It was only outperformed by oMOPSO and SMPSO in the presence of complex function shapes such as those of WFG1, WFG2 and WFG3. When presented with simpler concave shapes such as WFG4, WFG5, WFG6, WFG7, WFG8 and WFG9, CVEPSO-S outperformed oMOPSO and SMPSO every time. CVEPSO-S_K exhibits great potential due to its dominant hypervolume performance, however the utilization of CVEPSO-S_K is seen as a tradeoff due to its generally poor non-dominated solution distribution. Situations in which one prioritizes hypervolume over non-dominated solution distribution would be the most efficient use of CVEPSO-S_K.

VI. CONCLUSION

This work analyzed the performance of different forms of co-operation within VEPSO. Variants of VEPSO which utilized co-operative PSO algorithms were proposed, formally referred to as CVEPSO-S, CVEPSO-S_K and CVEPSO-H_K. These algorithms were analyzed and compared to VEPSO. The best performing CVEPSO variants were then compared to well-known multi-objective PSO optimization algorithms.

Results indicated that co-operation is a powerful addition to VEPSO. The standard VEPSO was outperformed in terms of hypervolume by CVEPSO-S, CVEPSO-S_K and CVEPSO-H_K for all the KTSs with respect to the hypervolume metric. VEPSO obtained equal or better distribution in comparison to CVEPSO-S_K and CVEPSO-H_K. CVEPSO-S was dominant over VEPSO in both metrics, as it consistently earned higher rankings. Concerning CVEPSO variant selection, it is clear that CVEPSO-S and CVEPSO-S_K performed better than CVEPSO-H_K. It was concluded that CVEPSO-H_K did not improve performance over CVEPSO-S or CVEPSO-S_K in deceptive environments. Performance was repeatedly observed to be worse than CVEPSO-S and CVEPSO-S_K.

The two best performing CVEPSO variants, CVEPSO-S and CVEPSO-S_K, were compared to two well known multi-objective PSO optimization algorithms. CVEPSO-S_K scaled exceptionally well in terms of hypervolume but poorly in terms of solution distribution. It was concluded that CVEPSO-S_K is desirable when one desires higher hypervolume with little regard for solution distribution. CVEPSO-S exhibited a balance between hypervolume and solution distribution, performing equally as well or better than oMOPSO and SMPSO for all functions except the few with complex shapes. Both algorithms were found to be very competitive with the well known oMOPSO and SMPSO algorithms, as they often performed equally or better. Previous work [4] has indicated that VEPSO stagnates and does not scale well with these algorithms, thus incorporating co-operation improves performance considerably and allows the algorithm to contend with several top multi-objective PSO algorithms.

There are many opportunities for future work in this area. Further research into the scalability of CVEPSO-S_K and CVEPSO-S is encouraged, specifically when the number of objectives becomes very large. Another potential area of interest is the sensitivity to dimensionality of VEPSO. Performance of VEPSO while using co-operative variants over a varying number of decision dimensions can be observed with the intent of determining the corresponding sensitivity to decision-space dimensionality in comparison to VEPSO.

REFERENCES

[1] K. E. Parsopoulos and M. N. Vrahatis, "Particle swarm optimization method in multiobjective problems," in *Proceedings of the 2002 ACM Symposium on Applied Computing*, ser. SAC '02. New York, NY, USA: ACM, 2002, pp. 603–607. [Online]. Available: http://doi.acm.org/10.1145/508791.508907

[2] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, Nov 1995, pp. 1942–1948.

[3] K. R. Harrison, B. Ombuki-Berman, and A. P. Engelbrecht, "Knowledge transfer strategies for vector evaluated particle swarm optimization," in *Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science, R. Purshouse, P. Fleming, C. Fonseca, S. Greco, and J. Shaw, Eds. Springer Berlin Heidelberg, 2013, vol. 7811, pp. 171–184. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-37140-0_16

[4] K. Harrison, A. Engelbrecht, and B. Ombuki-Berman, "A scalability study of multi-objective particle swarm optimizers," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*, June 2013, pp. 189–197.

[5] W. Matthysen, A. Engelbrecht, and K. Malan, "Analysis of stagnation behavior of vector evaluated particle swarm optimization," in *Swarm Intelligence (SIS), 2013 IEEE Symposium on*, April 2013, pp. 155–163.

[6] M. Helbig and A. Engelbrecht, "Dynamic multi-objective optimization using pso," in *Metaheuristics for Dynamic Optimization*, ser. Studies in Computational Intelligence, E. Alba, A. Nakib, and P. Siarry, Eds. Springer Berlin Heidelberg, 2013, vol. 433, pp. 147–188. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-30665-5_8

[7] M. Helbig, "Solving dynamic multi-objective optimisation problems using vector evaluated particle swarm optimisation," Ph.D. dissertation, Pretoria, South Africa, 2012.

[8] M. El-Abd, "Cooperative models of particle swarm optimizers," Ph.D. dissertation, Waterloo, Canada, 2008, aAINR43264.

[9] T. Hendtlass, "Particle swarm optimisation and high dimensional problem spaces," in *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*, May 2009, pp. 1988–1994.

[10] F. van den Bergh and A. Engelbrecht, "Cooperative Learning in Neural Networks using Particle Swarm Optimizers," pp. 84–90, 2000. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.27.4265

[11] N. Unger, B. Ombuki-Berman, and A. Engelbrecht, "Cooperative particle swarm optimization in dynamic environments," in *Swarm Intelligence (SIS), 2013 IEEE Symposium on*, April 2013, pp. 172–179.

[12] F. Van den Bergh and A. Engelbrecht, "A cooperative approach to particle swarm optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 8, no. 3, pp. 225–239, June 2004.

[13] J. Grobler, A. P. Engelbrecht, and V. S. S. Yadavalli, "Multi-objective de and pso strategies for production scheduling," in *2008 IEEE World Congress on Computational Intelligence*, J. Wang, Ed., IEEE Computational Intelligence Society. Hong Kong: IEEE Press, 1-6 June 2008, pp. –.

[14] M. Helbig and A. Engelbrecht, "Influence of the archive size on the performance of the dynamic vector evaluated particle swarm optimisation algorithm solving dynamic multi-objective optimisation problems," in *Evolutionary Computation (CEC), 2015 IEEE Congress on*, 2015.

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

[16] K. Bringmann and T. Friedrich, "Approximating the volume of unions and intersections of high-dimensional geometric objects," in *Algorithms and Computation*, ser. Lecture Notes in Computer Science, S.-H. Hong, H. Nagamochi, and T. Fukunaga, Eds. Springer Berlin Heidelberg, 2008, vol. 5369, pp. 436–447. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-92182-0_40

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

[18] 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

[19] F. Van Den Bergh, "An analysis of particle swarm optimizers," Ph.D. dissertation, Pretoria, South Africa, 2002, aAI0804353.

[20] S. Huband, L. Barone, L. While, and P. Hingston, "A scalable multi-objective test problem toolkit," in *Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science, C. Coello Coello, A. Hernndez Aguirre, and E. Zitzler, Eds. Springer Berlin Heidelberg, 2005, vol. 3410, pp. 280–295. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-31880-4_20

[21] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *Evolutionary Computation, IEEE Transactions on*, vol. 10, no. 5, pp. 477–506, Oct 2006.

[22] C. A. Coello Coello and M. Lechuga, "Mopso: a proposal for multiple objective particle swarm optimization," in *Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on*, vol. 2, 2002, pp. 1051–1056.

[23] 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 *Computational intelligence in multi-criteria decision-making, 2009. mcdm '09. IEEE symposium on*, March 2009, pp. 66–73.