A Fast Hypervolume Driven Selection Mechanism for Many-Objective Optimisation Problems

Shahin Rostami[a,*], Ferrante Neri[b]

[a]Department of Computing and Informatics, Bournemouth University, Bournemouth, BH12 5BB, United Kingdom.
[b]Centre for Computational Intelligence, De Montfort University, Leicester, LE1 9BH, United Kingdom.

Abstract

Solutions to real-world problems often require the simultaneous optimisation of multiple conflicting objectives. In the presence of four or more objectives, the problem is referred to as a "many-objective optimisation problem". A problem of this category introduces many challenges, one of which is the effective and efficient selection of optimal solutions.

The hypervolume indicator (or s-metric), i.e. the size of dominated objective space, is an effective selection criterion for many-objective optimisation. The indicator is used to measure the quality of a non-dominated set, and can be used to sort solutions for selection as part of the contributing hypervolume indicator. However, hypervolume based selection methods can have a very high, if not infeasible, computational cost.

The present study proposes a novel hypervolume driven selection mechanism for many-objective problems, whilst maintaining a feasible computational cost. This approach, named the Hypervolume Adaptive Grid Algorithm (HAGA), uses two-phases (narrow and broad) to prevent population-wide calculation of the contributing hypervolume indicator. Instead, HAGA only calculates the contributing hypervolume indicator for grid populations, i.e. for a few solutions, which are close in proximity (in the objective space) to a candidate solution when in competition for survival. The result is a trade-off between complete accuracy in selecting the fittest individuals in regards to hypervolume quality, and a feasible computational time in many-objective space. The real-world efficiency of the proposed selection mechanism is demonstrated within the optimisation of a classifier for concealed weapon detection.

Keywords: Multi-Objective Optimisation, Many-Objective Optimisation, Hypervolume Indicator, Selection Mechanism, Evolutionary Optimisation

[*]Corresponding author
Email addresses: srostami@bournemouth.ac.uk (Shahin Rostami), fneri@dmu.ac.uk (Ferrante Neri)

1. Introduction

Optimisation metaheuristics are composed of two phases: search to generate a new candidate solution and selection to choose the solutions to retain for the following iteration, e.g. see [25] and [12]. In multi-objective optimisation, the most critical operation is the selection since the fitness based comparisons must take into account the fact that a solution can be better performing than another in terms of one objective and not another. Candidate solutions in this situation are said to not dominate each other. The theoretical set of solutions which are not dominated by any other solution is referred to as Pareto-optimal (or simply Pareto) set [20, 65]. Metaheuristics designed to solve multi-objective problems aim to detect an approximation of the Pareto set (approximation set) [102]. The term approximation set is used to refer to "the set of all non-dominated points found during the run" [51], that is, the population at each iteration/generation of a multi-objective optimisation algorithm [52].

Many applications, such as engineering design, require that one solution (or in some cases a few alternatives) rather than a large set is ultimately selected. The process of performing this selection is named Decision Making while the criterion or algorithm that leads to the decision making is said to be the Decision Maker (DM). In other words, the DM implicitly classifies "interesting and uninteresting" solutions. The area of the objective space where the interesting solutions fall within is named the Region Of Interest (ROI). It must be noted that the multi-objective optimisation algorithm that detects the set of non-dominated solutions and the DM are related entities that perform different phases of the same task.

A good representation of a Pareto-optimal set, in terms of DM action, is characterised in three key areas, see [72]. These are illustrated graphically in Figure 1 and listed in the following:

- Proximity: This tells the DM how close the approximation set is to the true Pareto-optimal front. An ideal approximation set should be as close as possible in proximity to the true Pareto-optimal front. In practise, proximity cannot be used as a measure of quality of the approximation set during the optimisation process, because the true Pareto set is not known.

- Diversity: This characterises the distribution of the approximation set both in the extent and uniformity of that distribution. The ideal approximation set should be uniformly distributed across the trade-off surface of the problem.

- Pertinence: This criteria measures the relevance of the approximation set to the DM. Ideally the approximation set should contain a number of solutions which satisfy the DM's expressed preferences.

Conventional multi-objective optimisation techniques often fail to satisfy these criteria. For example, the goal-attainment method [33] and the weighted-sum method [38] both only provide single solutions to the optimisation problem - thus failing to provide a diverse distribution of solutions. Population based

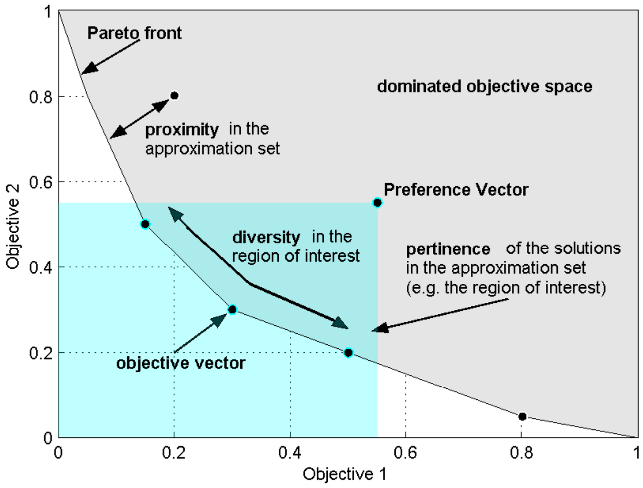


Figure 1: Proximity, diversity, and pertinence characteristics in an approximation set in two-objective space.

meta-heuristics, such as Evolutionary Multi-Objective Optimisation (EMO) algorithms, are naturally more suitable to tackle multi-objective problems since they process a population of solutions which can then represent the Pareto set, see [22, 44, 54, 64, 78, 94, 97], also when coupled to local search components, e.g. see [77]. Furthermore, population-based algorithms for multi-objective optimisation can be easily endowed with simple and effective components to maintain a diversity of high quality solutions. Recently, in [47], it is proposed a selection mechanism which satisfies at first the diversity of the solutions and then promotes those with the highest proximity. An alternative approach would make use of a mathematical model to generate extra surrogate solutions, e.g. see [13].

A study on the effectiveness of variation in EMO algorithms is reported in [2].

1.1. Many-objective Optimisation

The higher the number of objectives, the more challenging the pairwise comparison of solutions and the subsequent selection process. A multi-objective optimisation problem with more than three objectives is referred to as *many-objective* optimisation problem [28], [42, 56].

Analogous to the curse of dimensionality when large scale problems are considered, many-objective problems can introduce challenging difficulties. These challenges have been analysed in the literature, e.g. see [16, 17, 42], and summarised in the following list:

- It is likely that almost all candidate solutions found throughout the optimisation process will be non-dominated, this poses an issue for EMO algorithms which rely on Pareto-dominance for selection pressure, [29, 53, 73].

- The number of candidate solutions required to produce an approximation set which reliably represents the trade-off surface increases exponentially [40, 49].
- The number of generations required to produce an approximation set increases, thus making the calculation time of a single run infeasible.
- Search operators become ineffective at detecting new non-dominated solutions in the presence of many-objectives [40].
- Approaches which attempt to promote the diversity in the objective space, see [73], can cause issues in terms of convergence, see [84], [58] and [56]. Indeed, convergence and diversity are conflicting in the many-objective case, see [73] and [56].
- The visualisation of candidate solutions becomes difficult, often resulting in the use of heat-maps or parallel-coordinate plots. This poses a difficulty to the DM as the selection of a final candidate solution may become non-intuitive [85].

The transition between the multi-objective problem domain and the many-objective problem domain is not straightforward, such that the methods used to optimise solutions for a multi-objective problem have little to none of the desired effect when applied to a many-objective problem. A fundamental example of this is that the selection mechanisms based dominance that perform well on multi-objective problems (two or three objectives) [18, 20, 79], often do not perform well when four or more problem objectives are considered as shown in [3, 36, 37, 41, 46, 50, 63, 73, 103]. Selection based on dominance is inefficient at producing a strong selection pressure toward the Pareto-optimal front in the presence of many objectives, as throughout the optimisation process it is likely that the entire population will consist of entirely non-dominated solutions.

Several alternative algorithmic solutions have been proposed to perform the selection. The following non-exhaustive classification is here proposed.

- **Selection methods that use a reference vector.** These methods are focussed on the diversity and ideally aim at achieving an approximation set equally spaced on the Pareto front. In order to achieve this aim, these methods use a reference vector (or weight) and normal distributions to select along each coordinate (in the objective space) the points that are sufficiently distant. A prominent family of algorithms based on this logic consists of the algorithms based on Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) introduced in [95] and initially proposed for multi-objective problems. A further development of the MOEA/D algorithm has been presented in [96] where the MOAE/D with Dynamic Resource Allocation (MOEA/D-DRA) has been introduced. MOEA/D-DRA decomposes the many-objective space into multiple single-objective spaces (sub-problems) and then assigns them different computational budgets. This algorithm, which has been a competition winner at IEEE CEC is currently one of the most effective and robust solutions at tackling complex multi-objective and at least five-objective problems. Another interesting example of this category is the Non-dominated Sorting Genetic Algorithm III (NSGA-III) [21, 45]. A further feature of NSGA-III is the use of niching during recombination. This mechanism has been proposed to increase the exploitation of the algorithm. A recent example of an algorithm based on this logic is given in [93]. Another modern example is given in [48] where although a reference vector is used, a diversity-first convergence-second selection strategy.

- **Selection methods that divide/classify the population.** These methods have the same purpose of the methods based on reference vectors but achieve this goal by mapping and dividing the objective space. A mapping is then used as a reference to select the points so that they are equally spaced. A historical example in multi-objective optimisation is the Adaptive Grid Algorithm (AGA), see [52], where a grid in the objective space is used to control the population diversity. Another important example is [73] which employed the mechanism to promote diversity. The employment of a grid in the objective space has been reinterpreted and implemented in [60, 91]. Modern studies are reported in [57] where a computationally efficient implementation of a grid-based Pareto approximation is proposed, [58] which is related also to reference vector methods but makes use of distances in the objective space to shift the solutions and guarantee equal spacing, and [56] where the population is structured in multiple layers.

- **Selection methods that integrate the DM into the search.** These methods bias the selection by preferring some objectives rather than others. The goal of this approach is to have a high representation of points falling within the ROI. In [98] the direction of the knee points of the Pareto set (those points which represent the lack of a clear preference towards a specific objective) is preferred. This approach shares with [97] the philosophy of avoiding extensive pairwise comparisons and thus handle better large objective spaces. Furthermore, this approach can be seen as a modern re-interpretation of a weighted sum approach where each weight coefficient is set to 1. In [14, 15] the reference vector, i.e. a vector containing the preference weights for each objective, is integrated within the selection process, in order to exclude solutions which do not fall within the ROI from the search. The idea of the incorporation of DM preferences into the algorithmic selection has been proposed in other contexts, e.g. [62, 75, 76].

- **Selection methods that make use of indicators.** These methods collect other pieces of information to extract a scalarized parameter which then assists the selection. An important family is that of Indicator Based Evolutionary Algorithms (IBEAs) [43, 99, 100]. Recently, in [59] a performance comparison indicator to evaluate the quality of approximation sets with the aid of a reference point has been proposed. An interesting indication which does not require any knowledge about the problem is the hypervolume indicator, e.g. see [30]. The hypervolume indicator measures how much of

> the objective space is dominated by an approximation set. Although the hypervolume indicator can bias the search towards extremal and knee points of the Pareto, see [59], the way the hypervolume indicator is constructed makes it very appealing in real-world problems. The main attractive point is that this indicator requires no information regarding the theoretical Pareto-front (which in real-world problems is often unknown). Successful examples of incorporating the hypervolume indicator into the optimisation process during the selection process are in [26] and [6] where an adapted version of the hypervolume indicator, named the contributing hypervolume indicator, is used. Other successful cases of the hypervolume used during the selection process are given in [39] and [82] within the context of the Multi-Objective Covariance Matrix Adaptive Evolution Strategy (MO-CMA-ES).

Although the hypervolume indicator is a very valuable sorting criterion for selection, its calculation presents the drawback that it depends exponentially on the number of objectives. Consequently, selection methods incorporating the hypervolume indicator become computationally infeasible in the presence of many-objectives, see [56, 83]. This paper proposes a method to overcome the high computational cost of the hypervolume calculation. This aim is achieved by means of a novel selection mechanism, namely Hypervolume Adaptive Grid Algorithm (HAGA). The proposed HAGA calculates the hypervolume by using only a small set of solutions by using a grid logic inspired by AGA [52]. With the application of HAGA, reliable results in many-objective problems are obtained within a feasible calculation time.

As an analogy, the trade-off between the accuracy of the hypervolume indicator and its computational cost is much like the issues which have been encountered in collision detection in the area of computational physics. The computational cost is due to the accuracy required for collision detection, and the number of objects considered in the simulations. A popular and efficient solution to this issue is the use of a two-phase approach: a *broad phase* and a *narrow phase* [66]. The broad-phase is often a crude but computationally fast method of eliminating objects from an object pool which are to be considered for collision detection, in this sense the broad-phase determines which objects do not require further inspection. The narrow phase will then consider the remaining objects for the actual collision detection.

When moving to many-objective problems, optimisation algorithms can benefit from taking inspiration from the two-phase collision detection approach, especially in regards to the computational feasibility of the hypervolume indicator. In a two-phase approach, the contributing hypervolume indicator is well suited for the narrow-phase. An ideal broad-phase algorithm would be computationally lightweight and allow partitioning of the objective space in regards to an approximation set available at each generation of the optimisation process.

In order to have a reasonable computational cost without compromising on the performance of hypervolume indicator, this paper proposes a novel approach inspired by the collision detection analogy. The proposed method makes use of a two-phase approach to selection which incorporates the contributing hypervolume indicator. The contributing hypervolume indicator is incorporated such that it is only used to calculate the contributing hypervolume indicator values for sub-populations which are specified by an AGA. In this sense, the proposed approach makes use of an estimation of the hypervolume indicators combined with a population mapping. With this approach, population-wide subjection to the contributing hypervolume indicator is avoided and the computational cost of using the contributing hypervolume indicator for achieving selection pressure remains feasible for many-objectives.

The remainder of this article consists of the following sections. Section 2 describes the HV indicator, the related concept of Contributing Hupervolume (CHV) indicator, and the AGA. Section 3 introduces the proposed selection mechanism for many-objective problems. Section 4 provides a full statistical analysis of the results of the proposed novel two-phase approach in comparison to the CHV indicator, and selection by random permutation. Section 5 applies the proposed selection mechanism to a real-world many-objective optimisation problem concerning concealed weapon detection.. Section 6 gives the conclusions of our work, highlights the effectiveness of the proposed approach and its suitability to many-objective optimisation.

2. Background: Hypervolume Indicators and Adaptive Grid Algorithm

2.1. Hypervolume Indicator

The HV indicator (or s-metric) is a performance metric for indicating the quality of a non-dominated approximation set, introduced by [101] where it is described as the "*size of the space covered or size of dominated space*". It can be defined as [82]:

$$HV\big(f^{ref}, X\big) = \Lambda\left(\bigcup_{X_n \in X} \left[f_1(X_n), f_1^{ref}\right] \times \cdots \times \left[f_m(X_n), f_m^{ref}\right]\right) \tag{1}$$

where $HV\left(f^{ref}, X\right)$ resolves the size of the space covered by an approximation set X, $f^{ref} \in \mathbb{R}$ refers to a chosen reference point, and $\Lambda\left(.\right)$ refers to the Lebesgue measure [55]. This has been illustrated in Figure 2 in two-dimensional objective space (to allow for easy visualisation) with a population of three solutions.

The Hypervolume (HV) indicator is appealing because it is compatible with any number of problem objectives and requires no prior knowledge of the true Pareto-optimal front, this is important when working with real-world problems which have not yet been solved. The HV indicator is currently used in the field of multi-objective optimisation as both a proximity and diversity performance metric and also in the decision making process [34, 81].

Unlike dominance-based criteria which require only two solutions for performing a comparison (which can be used on an ordinal scale), a reference vector is required to calculate the HV indicator value (i.e. it requires the objective to be measured on an interval scale). When used for pairwise or multiple comparison of optimisation algorithms, this reference vector must be the same, otherwise the resulting HV indicator

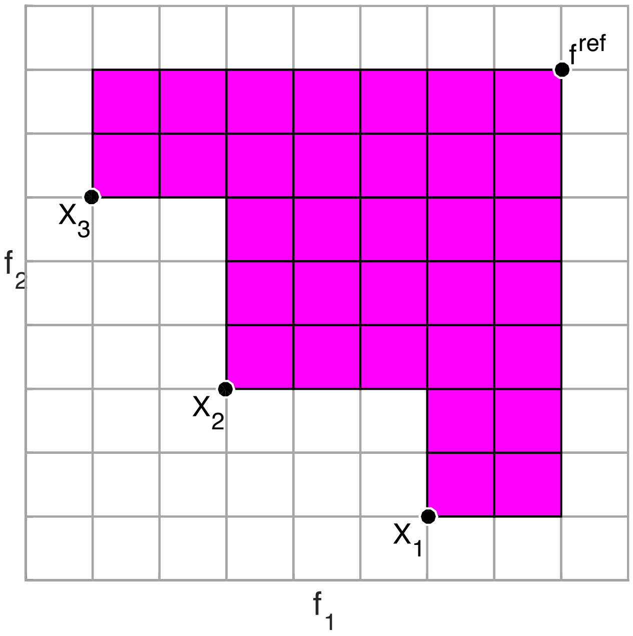


Figure 2: An example of the hypervolume indicator in two-dimensional objective space.

values are not comparable. This reference vector can be approximated as large values for each problem objective in order for all objective values in any approximation set to be within the reference vector. A more accurate method for selecting a reference vector, is to use the worst objective values from the union of approximation sets produced on a particular test problem, for each algorithm which is being considered for comparison. This has been described in the following:

$$\forall m \in \langle 1, \ldots M \rangle. \quad f_m^{ref} \leftarrow \max \left\langle X_m \cup f_m^{ref} \right\rangle \tag{2}$$

where f_m^{ref} is the highest value for objective m in the previous generations of the optimisation process, and X_m is a set of objective values for objective m from the current population.

Much effort has been made to speed up the HV calculation, e.g. see [5, 30, 87, 88]

Various implementations of the HV indicator have been presented in [5, 10, 30, 87, 88, 90], all with the aim to speed up its calculation. The HV indicator implementation employed throughout this study is a faster method for exact calculation authored by the Walking Fish Group [87]. The HV indicator has been employed in the performance assessment of algorithms in much of the multi-objective optimisation and evolutionary computation literature (e.g. [24, 36, 67, 74, 80]).

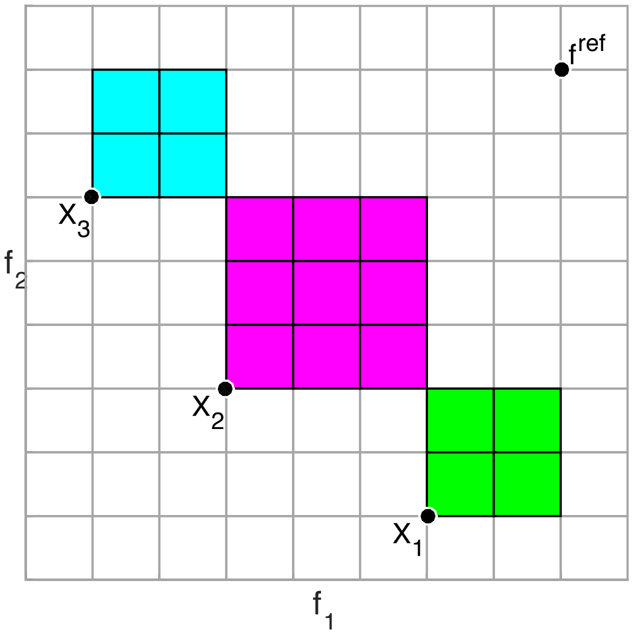


Figure 3: An example of the CHV indicator in two-dimensional objective space.

2.2. Contributing Hypervolume

The Contributing Hypervolume (CHV) indicator is a population sorting mechanism based on an adaptation of the HV indicator, see [26], [39], and [6]. The HV indicator works by calculating the size of the objective space that has been dominated by an entire approximation set with respect to a specified reference point, whereas the CHV indicator assigns each solution in an approximation set with the size of the space that has been dominated by each solution exclusively. With this information the population can be sorted by the most dominant and diverse solutions. This has been illustrated in Figure 3 in two-dimensional space with a population of three solutions.

Calculating the exact CHV indicator is attractively simple. The method begins by first calculating the HV indicator quality of a population X, and then for each solution in the population, the solution is removed and the HV indicator quality is again calculated for the new population. The new HV indicator value is then subtracted from the HV indicator value of the whole population, which results in the CHV indicator value of the solution which was removed. It is then possible to calculate the CHV indicator values of all the $\mu + \lambda$ solutions in the population (μ parents and λ offspring), order them by descending value so that they are ordered by the greatest explicit HV indicator contribution, and select the first μ solutions to form the next parent population. This approach has been listed in Algorithm 1.

Although many optimisation algorithms use the CHV as a sorting criterion for selection, its calculation becomes computationally infeasible as the number of objectives considered increases. Monte Carlo

Algorithm 1 CHV Indicator execution cycle

```
CHVIndicator($f^{ref}, X$)
$X_{HV} \leftarrow HV(f^{ref}, X)$
for n = 1 : $\lambda$ do
    $X_t \leftarrow X \backslash X_n$
    $HV_n \leftarrow HV(f^{ref}, X_t)$
    $CHV_n \leftarrow X_{HV} - HV_n$
end for
return CHV
```

approximations have been used to speed up the calculation of the CHV in [11], which through empirical experiments has shown that the method does not impair the quality of the approximation set. However, the speed increase provided by the Monte Carlo approximation method, still results into an infeasibility of the CHV indicator on problems consisting of five objectives or more.

2.3. Adaptive Grid Algorithm

The Adaptive Grid Algorithm (AGA) is a diversity preservation scheme which uses an adaptive grid to keep track of the density of solutions within divisions of the objective space [52]. To achieve this, a grid with a pre-set number of divisions is used to divide the objective space, and when a solution is generated, its grid location is identified and associated with it. Each grid location is considered to contain its own sub-population, and information on how many solutions in the archive are located within a certain grid location is available during the optimisation process, this has been illustrated in Figure 4.

3. A New Selection Mechanism: The Hypervolume Adaptive Grid Algorithm

The proposed Hypervolume Adaptive Grid Algorithm (HAGA) is a selection mechanism consisting of two-phases aiming of being computationally feasible in the presence of many objectives.

HAGA incorporates the use of a novel AGA implementation containing a number of features in order to make the AGA implementation suitable for many-objective optimisation. These novel features consist of:

- A new data structure for storing a solution's grid number up to any number of problem objectives;

- A new grid-proximity method for grid selection when searching for a solution to remove;

- A new scheme for the maintenance of global extremes for problem objectives.

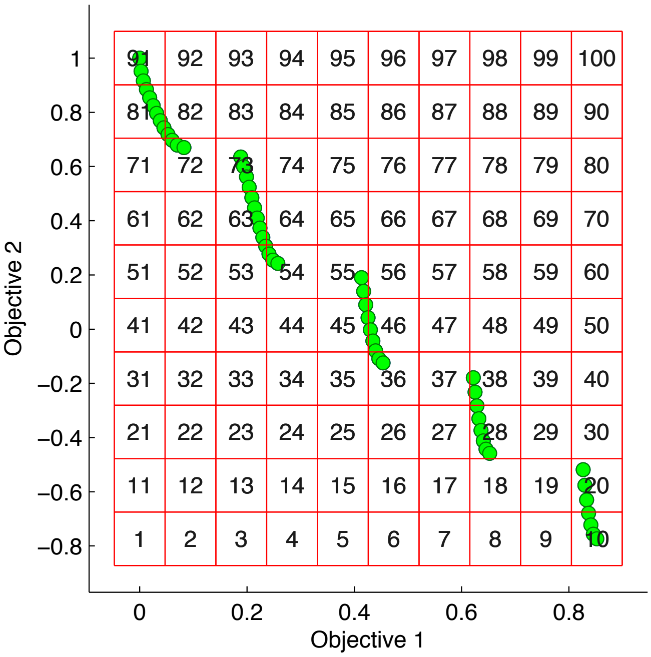


Figure 4: An example plot of a population and visualisation of the grid divisions managed by an AGA.

The execution cycle of HAGA is listed in Algorithm 2. With reference to Algorithm 2, HAGA processes an approximation set composed of $\mu + \lambda$ solutions with the aim of selecting μ of them. More specifically, the inputs of HAGA are $\mu + \lambda$ vectors of objective values (X_n, with $n = 1, 2, \ldots \mu + \lambda$) and the scalar μ is the number of solutions to be selected. The data structure containing all these vectors is indicated as X. The output of HAGA is then a new approximation set whose objective values are vectors allocated in a data structure $A \subset X$ composed of μ solutions. The output approximation set is named parent set and its corresponding solutions are named parent solutions. The notation $|X|$ refers to the cardinality of the population X.

Γ is a grid location consisting of multiple solutions and $CHV_{\mathbf{\Gamma}_t}$ is a vector of entries resulting from the execution of the CHV indicator on the solutions in the grid population $\mathbf{\Gamma}_t$, such that $\min(CHV_{\mathbf{\Gamma}_t})$ would resolve the solution which offers the lowest explicit HV indicator contribution in regards to the observed grid population.

In Algorithm 2, Line 5 identifies the grid location closest to the candidate solution and resolves a grid population, the method for which has been described in Section 3.2. This is considered to be the broad-phase of the two-phase approach. Once a grid location has been identified, it is used in the calculation of the narrow-phase (Line 7), which depends on the CHV indicator described in Section 2.2.

The basic principle of HAGA is to benefit from the CHV algorithm's ability to discriminate solutions based on the explicit HV they contribute to a population, but to do so in a way that doesn't introduce the

Algorithm 2 HAGA execution cycle

```
HAGA($X, \mu$)
$\lambda = |X| - \mu$
for n $= 1 : \mu + \lambda$ do
    if $|A| < \mu$ then
        $A \leftarrow A \cup X_n$
    else
    # Where ($\Gamma_n$) is the grid location of the maximally populated grid, described in (4).
        $\boldsymbol{\Gamma_t} \leftarrow$ ClosestGridFromMaximallyPopulated($\boldsymbol{\Gamma_n}$)
        $\boldsymbol{\Gamma_t} \leftarrow \boldsymbol{\Gamma_t} \cup X_n$
        $CHV_{\boldsymbol{\Gamma_t}} \leftarrow$ CHVIndicator($\boldsymbol{\Gamma_t}$)

        # Where $\min(CHV_{\boldsymbol{\Gamma_t}})$ is the solution offering the lowest CHV in the grid.
        if $\min(CHV_{\boldsymbol{\Gamma_t}}) = X_n$ then
            Discard the candidate solution $X_n$
        else
            Discard $\min(CHV_{\boldsymbol{\Gamma_t}})$ solution
            $A \leftarrow A \cup X_n$
        end if
    end if
end for
return A
```

computational infeasibility of using CHV on populations consisting of many-objective solutions.

In order to keep the computational cost feasible, HAGA makes use of the adaptive grid, where the objective space covered by a population is divided into a grid consisting of grid areas. This grid has a capacity for the number of vectors it can store, this capacity is set to μ, which is the number of parent solutions desired for selection. The solutions within the offspring are then added to this grid one by one. Throughout this process, there is no computation of the CHV algorithm until the grid reaches capacity. When the capacity is reached (the grid is populated by μ vectors) the inclusion of a new vector X_n must result into the eviction of another vector. In this case, the new vector is temporarily added to the population (which would be temporarily composed of $\mu + 1$ vectors). Then, a CHV is performed between the newly added vector and all the other solutions belonging to the same sub-population in the grid. With reference to Fig. 5, if we add X_4 we would need to apply the CHV for X_3, X_4, X_5 to evict one solution.

This procedure ensures that the CHV is computed for only a small number of solutions. This is done in order to determine which solution is to be evicted from the grid area to prevent the grid from exceeding capacity. In contrast, the CHV algorithm simply computes the CHV indicator value for every solution in the population.

3.1. New Grid Number Data Structure

The mathematical procedure for the proposed AGA structure in its entirety is described herein [52]. M defines the number of problem objectives and N defines the population size, whilst Δ defines the number of desired grid divisions for a problem objective within the objective space. The data structure X is an M by N matrix whose generic element x_{mn} refers to a solution's objective value:

$$\mathrm{X_n} = \langle x_{1n}, x_{2n}, \ldots, x_{Mn} \rangle \tag{3}$$

Γ is an M by N matrix of entries γ_{mn}, where every γ_{mn} refers to the grid location of an objective value x_{mn} in the divided objective space.

$$\boldsymbol{\Gamma_n} = \langle \gamma_{1n}, \gamma_{2n}, \ldots, \gamma_{Mn} \rangle \tag{4}$$

To calculate $\boldsymbol{\Gamma_n}$, grid location γ_{mn} of each objective value x_{mn} for each solution X_n needs to be resolved. To calculate a solution's grid location, the padded grid length Λ, where

$$\Lambda = \langle \lambda_1, \lambda_2, \ldots, \lambda_M \rangle \tag{5}$$

for each objective needs to be calculated using the lowest and highest objective value for each objective

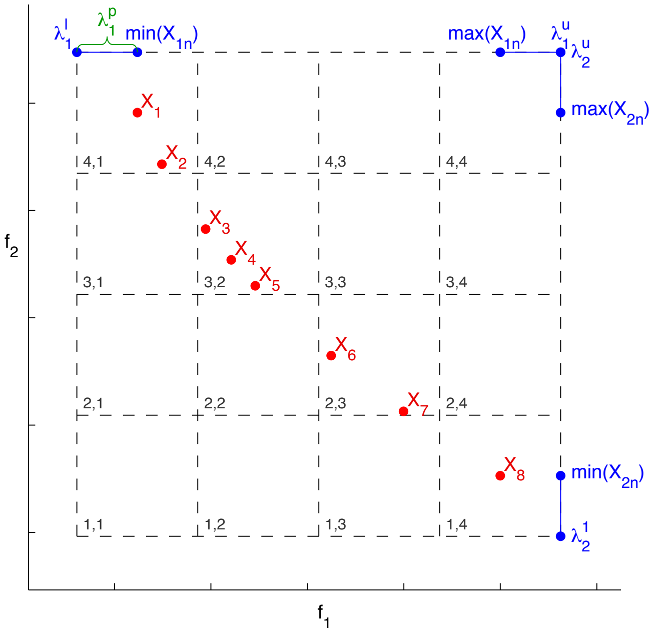


Figure 5: Illustration of the AGA and its elements on a two-objective approximation-set with $\Delta = 4$.

in the population:

$$\begin{aligned} \lambda_m^p &= \frac{|\min(X_m) - \max(X_m)|}{2(\Delta - 1)} \\ \lambda_m^l &= \min(X_m) - \lambda_m^p \\ \lambda_m^u &= \max(X_m) + \lambda_m^p \\ \lambda_m &= |\lambda_m^l - \lambda_m^u| \end{aligned} \tag{6}$$

where λ_m^l is the start point of the grid for objective m in the objective space, and λ_m^u is the end point of the grid for objective m in the objective space. These elements have been illustrated in Figure 5.

With the grid length and range calculated, it is possible to get the grid location of each solution's objective value using:

$$\gamma_{mn} = \left\lceil \frac{x_{mn} - \lambda_m^l}{\frac{\lambda_m}{\Delta}} \right\rceil \tag{7}$$

When the entries of $\boldsymbol{\Gamma_n}$ have been calculated, it can be used to identify the grid location of a solution X_n. In this new method, the grid location $\boldsymbol{\Gamma_n}$ is defined by a vector rather than a scalar, for example in a five-objective problem a grid location can be described by being at location $\boldsymbol{\Gamma_n} = \langle 2, 4, 1, 1, 2 \rangle$.

Grid Number Data Structure Worked-Example

As an example, a population X of five ($N = 5$) solutions X_n for a five-objective problem ($M = 5$) has been presented in Table 1 and Figure 6.

Table 1: An example population X of objective values x_{mn} to be subjected to the proposed AGA scheme.

	x_{1n}	x_{2n}	x_{3n}	x_{4n}	x_{5n}
X_1	0.5	0.5	5.0	2.5	1.5
X_2	0.6	0	5.0	3.0	1.4
X_3	0.5	3.5	4.5	2.5	1.5
X_4	0.8	3.2	4.2	3.0	1.2
X_5	1	3	4	2	1

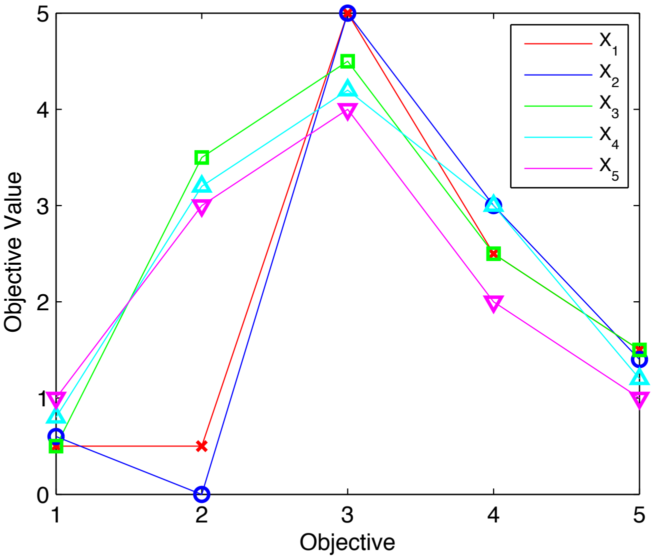


Figure 6: Parallel-coordinate plot of the Population X used in the proposed AGA example.

This population X has been subjected to the proposed AGA scheme to resolve the grid location $\boldsymbol{\Gamma_n}$ of each solution X_n, with an AGA parameter setting of ($\Delta = 4$) grid divisions. The grid locations resolved by the AGA scheme have been presented in Table 2 and the objective values x_{mn} have been plotted in their respective grid locations γ_m in Figure 7, where the plot markers correspond to those used in Figure 6.

Table 2: Grid locations Γ for the example population X of objective values x_{mn}.

	γ_{1n}	γ_{2n}	γ_{3n}	γ_{4n}	γ_{5n}
$\boldsymbol{\Gamma_1}$	1	1	4	3	4
$\boldsymbol{\Gamma_2}$	2	1	4	4	3
$\boldsymbol{\Gamma_3}$	1	4	2	3	4
$\boldsymbol{\Gamma_4}$	3	4	2	4	2
$\boldsymbol{\Gamma_5}$	4	4	1	1	1

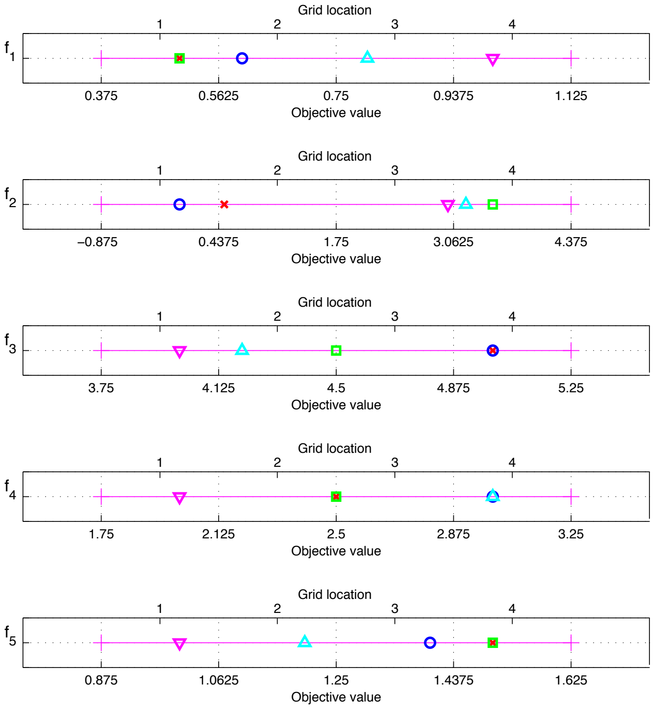


Figure 7: One dimensional plots illustrating the grid locations resolved by the AGA scheme for each objective value, where the plot markers correspond to those used in Figure 6.

The results from this example show that the example population does not consist of any solutions which are in the same grid square (otherwise their Γ entries would be identical).

3.2. Selecting a Solution to Replace

The method for selecting a grid location to remove a solution when the archive is at capacity is important when moving to many-objective problems. Selecting at random from grid locations which are at the same population density increases the probability of causing genetic drift, and decreases the diversity quality of the population. This undesirable effect is scaled exponentially as the number of problem objectives increase, and it is for this reason that many modern EMO algorithms now incorporate a niching approach when concerned with selection [21, 96]. Therefore, it is desirable to find the grid location which is close to both solutions in the objective space and also at a higher density.

Storing grid locations as a single scalar value is not helpful when calculating distance between grid locations or for storing grid locations for a many-objective problem. The grid location structure used in the proposed AGA scheme described above enables an intuitive method for finding the distance between grid locations. By establishing the grid location which a candidate solution would be assigned to, it is possible to find the difference between its grid location and other grid locations which are at high density. This procedure is performed to find out which one it is closest to by summing the difference of the grid location vectors.

Selecting a Solution to Replace Worked-Example

For example, if a new solution $X_6 = \langle 0.6, 0.5, 4, 3, 1.1 \rangle$ was to be included as a candidate solution as part of the proposed AGA scheme, it would resolve a grid location of $\mathbf{\Gamma_6} = \langle 2, 1, 1, 4, 2 \rangle$. The distance δ_6 between this grid location ($\mathbf{\Gamma_6}$) and the grid locations of the other solutions can be found by finding the absolute difference of each corresponding entry of the candidate solution's grid location and another solution from the population, and then summing those values.

$$\delta(i, j) = \sum_{m=1}^{M} |\gamma_{m,i} - \gamma_{m,j}| \tag{8}$$

The distances δ_n between the grid location $\mathbf{\Gamma_6}$ of solution X_6 and all the other solutions in the population presented in Table 1 have been presented in Table 3. The results show that the solution closest in proximity to solution X_6 is solution X_2, this has been visualised in Figure 8.

Table 3: Grid locations Γ for the example population X of objective values x_{mn}.

	$\mathbf{\Gamma_1}$	$\mathbf{\Gamma_2}$	$\mathbf{\Gamma_3}$	$\mathbf{\Gamma_4}$	$\mathbf{\Gamma_5}$
δ_6	7	4	8	5	9

Using this method of resolving densely populated grid locations which are closest in proximity to a candidate solution, it is possible to implement the approach listed in Algorithm 3.

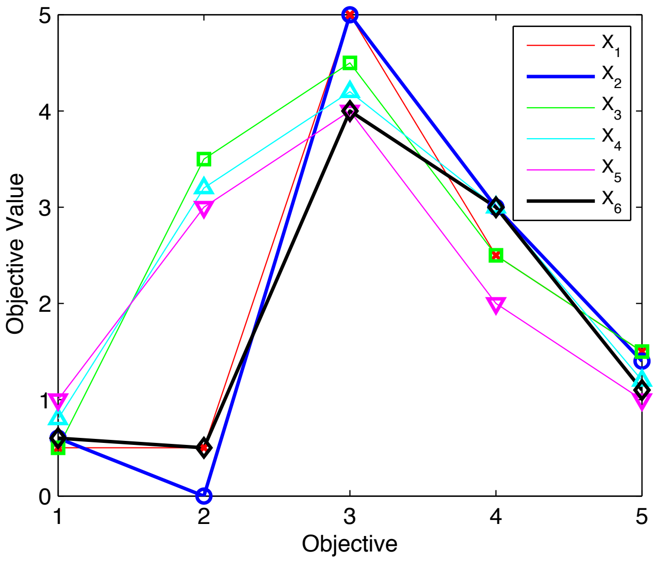


Figure 8: Parallel-coordinate plot of the Population X used in the proposed AGA example. X_6 and the solution closest in proximity to it, X_2, have been presented using thicker lines.

Algorithm 3 Closest grid location from maximally populated grid locations

```
   ClosestGridFromMaximallyPopulated($\boldsymbol{\Gamma_n}$)
   # If the number of grid locations containing the maximal number of solutions exceeds 1
1: if $|\max(\Gamma)| > 1$ then[5]
   # Where max Gamma is the set of grid locations containing the maximal number of solutions.
2:     for $\gamma$ in $\max(\Gamma)$ do
   # Calculate the differences between the grid locations
   # Where $\gamma_n$ is the equivalent of $\gamma$ in $\max(\Gamma)$ but for $\boldsymbol{\Gamma_n}$
3:         $\delta_c \leftarrow \delta(\gamma_n, \gamma)$[6]
4:     end for
5:     $\boldsymbol{\Gamma_t} \leftarrow \boldsymbol{\Gamma_{\min(\delta_c)}}$
6: else
7:     $\boldsymbol{\Gamma_t} \leftarrow \max(\Gamma)$
   # Where $\max(\Gamma)$ at this point is the single grid location resolved from the singleton set.
8: end if
   return $\boldsymbol{\Gamma_t}$
```

[5] In this case, $|\Gamma|$ refers to the cardinality of Γ

[6] In these cases, ← refers to the assignment of the right operand to the left operand.

This algorithm is invoked by Line 5 of Algorithm 2 by $ClosestGridFromMaximallyPopulated(X_n)$. Line 1 checks to see if there is a single grid location which contains the most solutions, if there is more than one grid location that contains the most number of solutions then the algorithm identifies the grid square which is closest in proximity to the candidate solution. δ_{nc} refers to the difference between the candidate solution's grid location and other grid locations, and $\Gamma_{\min(\delta_c)}$ refers to the grid location which is the closest in proximity among those containing the maximal number of solutions.

With this feature, it is possible to remove solutions from grid locations which have the highest number of solutions, to make room for solutions in grid locations which have a low number of solutions. This encourages diversity in the overall population.

3.3. Maintaining Extreme Solutions

In previous AGA implementations, the extreme values for each objective were preserved at grid level. In the proposed AGA scheme, solutions containing extreme values for problem objectives (with the candidate solution taken into consideration) are removed from the population before it is subjected to the AGA. No special treatment is given to solutions containing locally (within their grid location) extreme objective values. This ensures candidate solutions are given a better chance of entering the archive than they would have had if they had come up against those solutions containing extreme values. This preserves the overall spread whilst encouraging new solutions to enter the archive.

3.4. Further Remarks

- The proposed HAGA has been designed to be integrated within generational schemes. On steady-state schemes, HAGA would require an extensive calculation at the insertion of each new solution. Consequently, HAGA could result in being computationally expensive when compared to methods employing a different logic.

- The proposed HAGA, although makes use of a grid, is deeply different in its logic when compared to grid-based selection mechanisms, see e.g. [91]. While other methods use the grid to select the population so that it is diverse (in the objective space), the proposed HAGA uses the grid to approximate the calculation of the hypervolume indicator which is then guiding the selection.

4. Numerical Results

The observations made throughout this study have led to the definition of the following testable hypotheses:

Hypothesis 1. *A two-phase approach to selection which incorporates the HV indicator can produce selection pressure towards the Pareto front.*

Hypothesis 2. *A two-phase approach to selection which incorporates the HV indicator can outperform the CHV indicator in regards to execution time.*

Hypothesis 3. *A two-phase approach to selection which incorporates the HV indicator can perform comparably within a 10% difference in regards to an achieved HV indicator value.*

Section 4.1 describes the experimental setup for a number of synthetic test functions used to design test cases for experimentation for the purpose of proving the hypotheses outlined above. The methods of performance assessment have been described in Section 4.2, including the performance metrics employed, the selected sample size, and the use of non-parametric testing. The experiments conclude with an analysis of the results in Section 4.3.

4.1. Experimental setup

In order to test the proposed HAGA selection scheme, the selection process of a generic algorithm generation has been simulated. Our test has been performed on the popular Walking Fish Group (WFG) Tool-kit (WFG1 through to WFG9), see [1, 4, 7, 68, 74]

In order to test the capability of the proposed selection approach to handle very diverse scenarios, this test suite has been used for the proposed method and its competitors, for several levels of dimensionality of the objective space. The test suite has been run using configurations for experiments which consider 2, 3, 5, 10, and 15 objectives.

The selection approaches considered for comparison using these test cases are:

- **Selection by Random Permutation** (RNDPERM), where a random permutation of the integers from 1 to λ (in the case of our experiments, $\lambda = 200$) are generated, and then the first μ (in the case of our experiments, $\mu = 100$) integers are used as indices for selecting solutions in the λ population;
- **Selection by the Contributed Hypervolume Indicator** (CHV) described in Section 2.2 according to the implementation described in [8, 26], see (1). The λ population is sorted in descending order by the returned CHV indicator values and the first μ solutions are selected;
- **Selection by the here proposed Hypervolume Adaptive Grid Algorithm** (HAGA) according to the implementation described in Section 3.

In order to compare the performance of the proposed selection approach to the considered alternatives, the experiments are designed around a comparison between the considered selection approaches independent of any host optimisation process. As a result of this, the performance of the considered approaches to selection will not be influenced by external factors. The purpose of these selection approaches is to select a sub-set of μ (in the case of our experiments, $\mu = 100$) solutions from an existing set of λ (in the case of our experiments, $\lambda = 200$) solutions. In the context of an evolutionary algorithm, the λ population would be the offspring population from the previous generational iteration, and the μ population would be the parent population which is to be used as parents for evolutionary operators such as recombination.

It must be observed that the Selection by Random Permutation is a naive approach which is based on a simple randomisation. The computational cost of this selection mechanism is expected to be very modest. On the other hand, due to the lack of intelligent and effective components, the performance of this selection scheme is expected to be not too high. The Selection by CHV indicator is a sophisticated mechanism that is likely to offer very good performance. Unfortunately, the very high computational cost of this method makes it inapplicable for the many-objective case. Ideally, a selection method for many-objective problems would have the accuracy of CHV and the computational cost of RNDPERM.

The proposed HAGA mechanism attempts to approximate this ideal situation by calculating the CHV indicator only for limited groups of vectors. HAGA is a trade off between CHV and RNDPERM philosophies and is here proposed to be a method that, albeit conceptually sophisticated, is computationally cheap and capable of offering very good performance in various multi-objective and many-objective problems.

Of the considered approaches, only HAGA requires a parameter setting, the desired number of grid divisions Δ (which is a discrete parameter).

The chosen value of $\Delta = 3$ has been set subsequently to a tuning where the values between 2 and 20 have been taken into consideration, Figure 9 illustrates this exploration for the ten-objective WFG1 test problem.

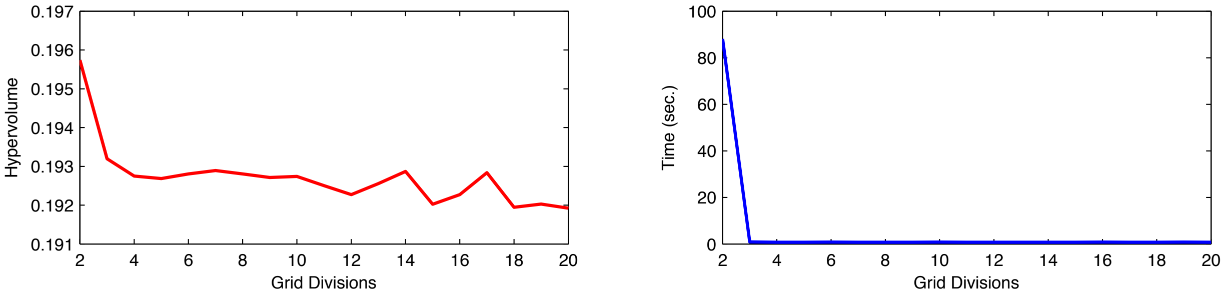


Figure 9: The effect of the number of grid divisions (Δ) on the Hypervolume performance and execution time of HAGA, demonstrated for the ten-objective WFG1 test problem.

The $\Delta = 3$ setting has been chosen as it proved to give the best performance in terms of computational cost. A lower value of Δ would result in fewer grid squares and therefore larger grid square populations. This will result in worst performance in the computation of the CHV indicator which is used at grid level. In contrast, higher numbers for Δ will result in many more grid locations which may hold only a few

Table 4: Settings used throughout the experiments.

Parameter name	Parameter value
Test functions considered	WFG1 to WFG9
No. objectives considered	$\langle 2, 3, 5, 10, 15 \rangle$
Execution sample size	25
λ population size	200
μ population size	100
Performance metrics	1. HV Indicator 2.Execution Time (secs)

solutions and often only a single solution. This scenario does not ensure the best performance out of the CHV indicator, as often only a single solution will be subjected to it.

It is important to note that for all problems, e.g. a fifteen-objective problem, although the value of Δ being set to 3 results in 3^{15} grid locations available within the grid's structure, these are only possible grid locations. The majority of the possible grid locations available in the grid's structure are not populated because they are located at: dominated areas of the objective space; unpopulated areas of the trade-off surface (either as a result of the design of the objective function or poor diversity); or impossible regions of the objective space. If a grid location does not contain any solutions, it is not considered in any comparisons and no space is reserved for this location during the algorithms execution.

The conditions under which these mechanisms are tested are summarized in Table 4.

Each test case involves using one of the considered approaches for the selection of μ individuals from a population of size λ, in all our test cases this means selecting the best 100 solutions from a population of 200 solutions. These test cases exist for permutations of all considered problem objectives, test functions, and with 25 samples for each. Each test case uses a unique λ population of 200 solutions, these solutions have been generated using the competition winning MOEA/D-DRA algorithm (see [96]). The size of the populations produced by MOEA/D-DRA is restricted to 200 by initialising the algorithm with 200 normalised and evenly distributed weights.

This approach to designing test cases and generating sample λ populations ensured that each considered approach will be subjected to approximation sets consisting of varying proximity and diversity (with a sufficient sample size of 25), for every considered test function, and for every considered problem objective configuration. The HV indicator performs better (in regards to execution time) on some test functions more than others, this difference in performance is not definitively a result of which test function is used, but instead the Pareto-optimal shape produced by solutions to the problem. For this reason it is important that 25 unique samples are used, in particular to evaluate each selection approach on "worst-case" scenarios - as

Table 5: Parameter settings used for the MOEA/D-DRA algorithm used to generate the λ populations.

Parameter name	Parameter value
Niche size	20
Population size	200
Maximum update number	2
Function evaluations	20,000

Table 6: Hardware and software configurations of the computing cluster used to generate the results.

Configuration name	Configuration value
Architecture	Linux-x64
RAM	63 GB
CPU	Intel(R) Xeon(R) CPU E5-2650 @ 2.00GHz ×16
Total CPU Cores	192
Nodes	16
MATLAB version	R2014a 64-bit (glnxa64)
Hypervolume Indicator	WFG HV 1.0.3

each approach is only as good as its "worst-case" execution in regards to computational cost.

All test cases have been executed on a computing cluster with the configuration listed in Table 6. A computing cluster was required due to the computational cost of executing the CHV indicator on many-objective approximation sets, the same hardware and software was used for all the selection approaches.

4.2. Performance Assessment

Selecting a relevant and sufficient method of performance assessment is a necessity when evaluating or comparing EMO algorithms. These methods of performance assessment can be used to gauge an EMO algorithm's performance in regards to the proximity, diversity, and pertinence of the final approximation set. Section 4.2.1 describes the performance metrics used to evaluate the considered selection approaches. Section 4.2.2 justifies the sufficiency of the selected sample size for comparison of the considered selection approaches. Section 4.2.3 describes the methods of non-parametric testing employed for the statistical analysis of the selection approaches considered.

4.2.1. Performance Metrics

In order to verify the hypotheses described in Section 4, performance metrics must be employed to resolve the performance of each selection approach considered. Two performance metrics have been considered in this study.

To gauge and compare the performance of a selection approach in selecting a population which achieves a high HV indicator quality, the HV indicator $HV\left(f^{ref}, X\right)$ described in equation (1) (and here synthetically referred to as HV) has been used as the first performance metric. The HV indicator (described in Section 2.1) is used in conjunction with the reference vector f^{ref} which is generated by taking the worst objective values from the union of approximation sets produced on a particular test problem after 20,000 function evaluations (calculated using (2)), for each algorithm which is being considered for comparison. This approach to generating the reference point has been chosen to allow a fair comparison amongst all considered hypervolume driven approaches, and to simulate a real-world optimisation scenario where such auxiliary information is often not available before the optimisation process. Through this performance metric the amount of the objective space covered by each selection approach can be observed.

To gauge and compare the performance of a selection approach in regards to its execution time, the MATLAB R2014a implementation of the stopwatch timer has been used. This can be achieved using the command "*tic*;", followed by the execution of the selection approach, followed by the command "*executionTime* = *toc*;". In the following sections of this paper, to keep the notation simple, the cumulative execution time is indicated with T. This cumulative execution time indicates the time taken in seconds by the selection approach on a single CPU thread.

The means and standard deviations for both of the selected performance metrics have been presented in the results section, in order to show the average performance as well as the dispersion of the results.

The CHV indicator can be used to find the μ solutions in a population of size λ which offer the highest possible HV indicator quality, in regards to the μ solutions which score the highest CHV indicator values. This is achieved by selecting the first μ solutions from an ordered (descending) population based on their CHV indicator values (calculated with Algorithm 1). The lowest possible HV indicator quality which can be achieved by μ solutions from a population of size λ can also be resolved, in regards to the μ solutions which score the lowest CHV indicator values. This is achieved by selecting the last μ solutions from the same ordered population. By having the worst and best possible μ solutions in regards to the HV indicator, it is possible to calculate a score indicating the percentage accuracy of a selection approach. The calculation of this score has been defined in Algorithm 4.

This score of accuracy is required as it can be difficult to interpret the magnitude of difference between selection approaches, this is because the magnitude of difference varies from one problem to another, and is heavily dependant on the solutions subjected to the HV indicator. Therefore, the accuracy score will be

Algorithm 4 Calculating the HV indicator accuracy

```
**HVIndicatorAccuracy**$(f^{ref}, X, \mu, HV)$
1: $CHV \leftarrow$CHVIndicator$(f^{ref}, X))$
2: $X_U \leftarrow$ selected from X using first $\mu$ CHV indexes
3: $X_L \leftarrow$ selected from X using last $\mu$ CHV indexes
4: $X_U \leftarrow HV\left(f^{ref}, X_U\right)$
5: $X_L \leftarrow HV\left(f^{ref}, X_L\right)$
6: $A = \frac{HV - X_L}{(X_U - X_L)} \times 100$
**return** A
```

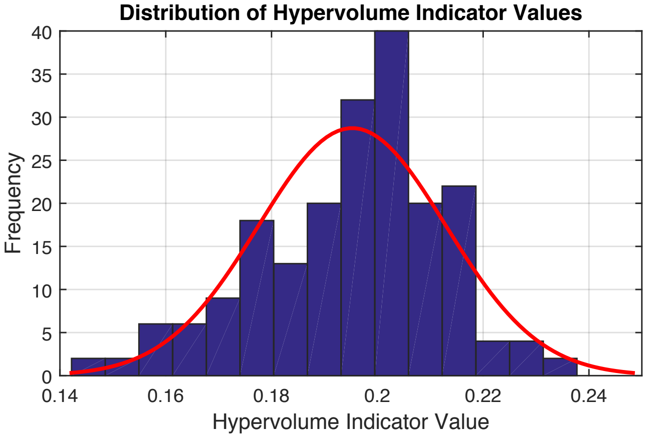


Figure 10: Histogram showing the distribution of the HV indicator values from 200 executions of MOEA/D-DRA on the WFG6 synthetic test problem.

included in the results for clarity of the difference between the considered selection approaches.

4.2.2. Sample Size Sufficiency

Selecting a sufficient number of samples when comparing optimisers is critical. The sample size of 25, in order to reduce stochastic noise, is re-occurring in the evolutionary computation literature (e.g.[19, 32, 86, 92, 94]). The sufficiency of this sample size has been tested by producing a large number of hypervolume indicator value samples by executing MOEA/D-DRA 200 times (the distribution of which has been illustrated in Figure 10) on the WFG6 synthetic test problem.

These 200 samples were then used to identify the relationship between the Standard Error of the Mean (SEM) and the sample size using:

$$SEM = \frac{SD}{\sqrt{N}} \tag{9}$$

This relationship has been illustrated in Figure 11 which shows the limited benefit of more than 25 independent executions of the algorithm on the synthetic test problem.

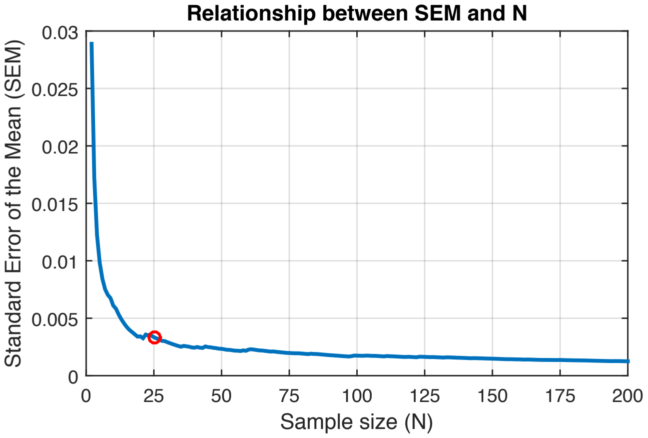


Figure 11: Relationship between Standard Error of the Mean (SEM) and the sample size of HV indicator values from 200 executions of MOEA/D-DRA on the WFG6 synthetic test problem.

4.2.3. Statistical Set-up

In order to find the significance in contrast amongst the results obtained by algorithms considered for comparison, a non-parametric test (encouraged by [23, 27]) for pairwise statistical comparison has been employed. The Wilcoxon signed-ranks [89] non-parametric test (counterpart of the paired t-test) has been used with the statistical significance value ($\alpha = 0.05$), this is able to rank the difference in performance between two algorithms (pairwise comparisons) over each approximation set.

The Wilcoxon signed-ranks non-parametric test has been employed to contrast the significance of the results generated by the proposed HAGA selection approach to every other considered selection approach, in regards to both the HV indicator and execution time performance metrics. These statistics will allow greater confidence in the verification of the hypotheses described in Section 4.

The choice of the use of a non-parametric test is experimentally justified by the fact that the data are not normally distributed. The latter statement has been shown by means of the one sample Kolmogorov-Smirnov test with the statistical significance value ($\alpha = 0.05$).

Furthermore, in order to strengthen the statistical significance of the results presented and have a general statistical sense of the significance of the algorithms across all problems, the Holm-Bonferroni procedure has been applied with confidence level of 0.05, see [35], [32], and [31].

Both Wilcoxon tests and Holm-Bonferroni procedure have been applied twice per each algorithm in order to evaluate the quality of the results and the time required to achieve them. The quality is expressed as the Hypervolume metric HV and cumulative execution time (CPU-time) T.

4.3. Results

This section presents the numerical results considered in this study. For each algorithm and each test problem, the average values of the HV indicator $\overline{HV}$ and the ± standard deviations σ are reported in Table 7. In the same table, average values of the calculation time with the corresponding standard deviation values are also displayed. In Table 7 the number of objectives under consideration is indicated with M. In the following table, a higher mean HV indicator value denoted by $\overline{HV}$ indicates greater performance in regards to area of the objective space dominated. In contrast, a lower mean execution time value denoted by $\overline{T}$ indicates greater performance in regards to the time taken to complete the execution for a selection approach, meaning the greater performance indicates less time taken.

The HV indicator values HV have been normalised to avoid the generation of large numbers and to allow for better comparison between selection approaches. The method of normalisation used has been defined in (10).

$$HV = \frac{HV}{\prod_{m=1}^{M} f^{ref_m}} \tag{10}$$

With this method of normalisation, the mean HV indicator values $\overline{HV}$ will always reside between 0 and 1.

The results of the Wilcoxon signed-ranks test have also been included in Table 7, where a '+' indicates that the null hypothesis was rejected, and the HAGA selection approach displayed statistically superior performance at the 95% significance level ($\alpha = 0.05$) on the considered test case. A '$-$' indicates that the null hypothesis was rejected, and the HAGA selection approach displayed statistically inferior performance. An '=' indicates that there was no statistically significant difference between both of the considered selection approaches on the considered test case.

Table 8 displays the results of the Holm-Bonferroni procedure in terms of both the hypervolume indicator HV and the execution time T.

It can be observed that HAGA has intermediate features with respect to CHV and RNDPERM. As expected, CHV always returns the best HV value with respect to λ because it performs a search over all the possible λ values. In contrast, RNDPERM always has the worst HV performance, and thus never outperforms the proposed selection approach (HAGA). Furthermore, the intermediate performance of HAGA is much closer to CHV performance than to the RNDPERM performance as displayed in Table 7, although HAGA performance on the HV cannot outperform the extensive search performed by CHV.

Regarding the execution time performance, it can be immediately observed that the three methods employ different logics with very different computational costs, especially when the complexity of the methods with respect to the dimensionality of the objective space is considered. Since RNDPERM simply performs random permutations it is computationally very cheap and its cost does not depend on the dimensionality

Table 7: Mean values ± standard deviation values of hypervolume indicator and calculation time for RANDPERM, CHV, and HAGA. A value in bold indicates the highest performing approach for the corresponding performance metric. A greater HV score indicates greater HV indicator performance, and a lesser T score indicates faster performance.

		RNDPERM						CHV						HAGA			
M	Fn	$\overline{HV}$	$\pm\sigma$		$\overline{T}$	$\pm\sigma$		$\overline{HV}$	$\pm\sigma$		$\overline{T}$	$\pm\sigma$		$\overline{HV}$	$\pm\sigma$	$\overline{T}$	$\pm\sigma$
2	WFG1	0.26103	±0.01048	=	**0.00089**	**±0.00010**	–	**0.26261**	**±0.01083**	=	0.01396	±0.00648	–	0.26261	±0.01084	0.21728	±0.10040
2	WFG2	0.63461	±0.01615	=	**0.00087**	**±0.00007**	–	**0.63628**	**±0.01622**	=	0.00759	±0.00428	–	0.63622	±0.01623	0.05468	±0.00986
2	WFG3	0.63135	±0.00609	+	**0.00090**	**±0.00007**	–	**0.63572**	**±0.00581**	=	0.01944	±0.00713	–	0.63460	±0.00753	0.36607	±0.07442
2	WFG4	0.46150	±0.00478	+	**0.00084**	**±0.00006**	–	**0.46533**	**±0.00468**	=	0.01147	±0.00475	–	0.46529	±0.00457	0.06388	±0.01354
2	WFG5	0.44960	±0.00223	+	**0.00086**	**±0.00006**	–	**0.45342**	**±0.00186**	=	0.02302	±0.00923	–	0.45319	±0.00194	0.41180	±0.05766
2	WFG6	0.49072	±0.00836	+	**0.00087**	**±0.00007**	–	**0.49440**	**±0.00834**	=	0.02138	±0.00742	–	0.49418	±0.00830	0.57747	±0.06463
2	WFG7	0.50823	±0.00497	+	**0.00088**	**±0.00008**	–	**0.51239**	**±0.00444**	=	0.01388	±0.00476	–	0.51235	±0.00442	0.25407	±0.07760
2	WFG8	0.44568	±0.00535	=	**0.00088**	**±0.00008**	–	**0.44944**	**±0.00538**	=	0.01462	±0.00599	–	0.44843	±0.00597	0.23082	±0.06540
2	WFG9	0.49869	±0.00482	+	**0.00086**	**±0.00007**	–	**0.50237**	**±0.00431**	=	0.01595	±0.00645	–	0.50195	±0.00443	0.32247	±0.11638

		RNDPERM						CHV						HAGA			
M	Fn	$\overline{HV}$	$\pm\sigma$		$\overline{T}$	$\pm\sigma$		$\overline{HV}$	$\pm\sigma$		$\overline{T}$	$\pm\sigma$		$\overline{HV}$	$\pm\sigma$	$\overline{T}$	$\pm\sigma$
3	WFG1	0.34652	±0.01176	=	**0.00086**	**±0.00006**	–	**0.35109**	**±0.01223**	=	0.15059	±0.00890	–	0.35036	±0.01253	0.59704	±0.04166
3	WFG2	0.81993	±0.01605	=	**0.00086**	**±0.00006**	–	**0.82707**	**±0.01454**	=	0.11019	±0.00946	–	0.82688	±0.01454	0.42262	±0.04688
3	WFG3	0.58817	±0.00993	+	**0.00087**	**±0.00009**	–	**0.59665**	**±0.00981**	=	0.11637	±0.00874	–	0.59614	±0.01125	0.52284	±0.04523
3	WFG4	0.54612	±0.01508	+	**0.00085**	**±0.00006**	–	**0.56339**	**±0.01269**	=	0.11447	±0.01082	–	0.56026	±0.01301	0.33683	±0.04832
3	WFG5	0.56191	±0.01090	+	**0.00087**	**±0.00008**	–	**0.57983**	**±0.00689**	=	0.15198	±0.00758	–	0.57434	±0.00965	0.53927	±0.03212
3	WFG6	0.60525	±0.00811	+	**0.00084**	**±0.00005**	–	**0.61545**	**±0.00996**	=	0.16443	±0.00732	–	0.61525	±0.00834	0.60009	±0.02691
3	WFG7	0.55759	±0.01213	+	**0.00085**	**±0.00006**	–	**0.57357**	**±0.01148**	=	0.13220	±0.00936	–	0.56834	±0.01256	0.46005	±0.05513
3	WFG8	0.50919	±0.01250	+	**0.00086**	**±0.00008**	–	**0.52295**	**±0.01162**	=	0.13297	±0.00937	–	0.52137	±0.01184	0.45089	±0.05183
3	WFG9	0.58318	±0.01089	+	**0.00088**	**±0.00009**	–	**0.59875**	**±0.00941**	=	0.13620	±0.01184	–	0.59536	±0.00946	0.48412	±0.05889

		RNDPERM						CHV						HAGA			
M	Fn	$\overline{HV}$	$\pm\sigma$		$\overline{T}$	$\pm\sigma$		$\overline{HV}$	$\pm\sigma$		$\overline{T}$	$\pm\sigma$		$\overline{HV}$	$\pm\sigma$	$\overline{T}$	$\pm\sigma$
5	WFG1	0.35384	±0.00489	+	**0.00085**	**±0.00006**	–	**0.35604**	**±0.00495**	=	1.65278	±0.16109	+	0.35599	±0.00495	0.78154	±0.10825
5	WFG2	0.84441	±0.01309	=	**0.00088**	**±0.00010**	–	**0.84972**	**±0.01316**	=	1.18229	±0.15747	+	0.84966	±0.01317	0.75523	±0.13534
5	WFG3	0.52060	±0.00915	=	**0.00088**	**±0.00009**	–	**0.53367**	**±0.00777**	–	0.67558	±0.07894	+	0.52522	±0.01116	0.56657	±0.03422
5	WFG4	0.60365	±0.01257	+	**0.00086**	**±0.00007**	–	**0.63447**	**±0.01208**	=	1.11964	±0.09373	+	0.62973	±0.01319	0.44297	±0.04703
5	WFG5	0.63766	±0.01701	+	**0.00088**	**±0.00007**	–	**0.67332**	**±0.01543**	=	1.36957	±0.07382	+	0.66853	±0.01631	0.57354	±0.03647
5	WFG6	0.70205	±0.01062	+	**0.00085**	**±0.00005**	–	**0.72360**	**±0.00946**	=	1.50883	±0.23307	+	0.72322	±0.01302	0.59686	±0.02849
5	WFG7	0.56674	±0.01387	+	**0.00088**	**±0.00006**	–	**0.59519**	**±0.01190**	=	1.33294	±0.10874	+	0.59138	±0.01479	0.53652	±0.03597
5	WFG8	0.50688	±0.01941	+	**0.00087**	**±0.00009**	–	**0.53508**	**±0.02046**	=	1.17091	±0.12492	+	0.53255	±0.02056	0.46476	±0.05669
5	WFG9	0.63646	±0.01386	+	**0.00088**	**±0.00009**	–	**0.66422**	**±0.01396**	=	1.25577	±0.09913	+	0.65887	±0.01413	0.54510	±0.03100

		RNDPERM						CHV						HAGA			
M	Fn	$\overline{HV}$	$\pm\sigma$		$\overline{T}$	$\pm\sigma$		$\overline{HV}$	$\pm\sigma$		$\overline{T}$	$\pm\sigma$		$\overline{HV}$	$\pm\sigma$	$\overline{T}$	$\pm\sigma$
10	WFG1	0.20132	±0.00333	=	**0.00084**	**±0.00007**	–	**0.20261**	**±0.00323**	=	1944.52962	±531.90048	+	0.20253	±0.00325	1.20437	±1.25042
10	WFG2	0.81638	±0.02935	=	**0.00085**	**±0.00009**	–	**0.82305**	**±0.02876**	=	379.51965	±216.71658	+	0.82292	±0.02854	0.83236	±0.53270
10	WFG3	0.36728	±0.01762	+	**0.00083**	**±0.00006**	–	**0.39405**	**±0.01654**	=	245.89231	±208.88644	+	0.38837	±0.01741	0.68271	±0.02954
10	WFG4	0.52966	±0.02916	+	**0.00084**	**±0.00005**	–	**0.57416**	**±0.02744**	=	1561.46369	±559.03596	+	0.56968	±0.02868	0.49096	±0.07500
10	WFG5	0.47830	±0.02242	+	**0.00085**	**±0.00006**	–	**0.54329**	**±0.02461**	=	2448.10859	±818.45709	+	0.53862	±0.02456	0.65928	±0.03369
10	WFG6	0.68586	±0.01854	+	**0.00083**	**±0.00006**	–	**0.71780**	**±0.01744**	=	1032.64211	±453.56564	+	0.71643	±0.01901	0.58623	±0.06133
10	WFG7	0.46453	±0.02266	+	**0.00083**	**±0.00006**	–	**0.50875**	**±0.01732**	=	2188.14782	±756.95283	+	0.50482	±0.01700	0.59102	±0.04497
10	WFG8	0.39281	±0.03009	+	**0.00083**	**±0.00004**	–	**0.44048**	**±0.02385**	=	988.63944	±442.54752	+	0.43631	±0.02313	0.44876	±0.05142
10	WFG9	0.47978	±0.03400	+	**0.00083**	**±0.00003**	–	**0.52364**	**±0.03176**	=	1653.95165	±508.52791	+	0.51589	±0.03149	0.59211	±0.03683

		RNDPERM						CHV						HAGA			
M	Fn	$\overline{HV}$	$\pm\sigma$		$\overline{T}$	$\pm\sigma$		$\overline{HV}$	$\pm\sigma$		$\overline{T}$	$\pm\sigma$		$\overline{HV}$	$\pm\sigma$	$\overline{T}$	$\pm\sigma$
15	WFG1	0.05828	±0.00083	=	**0.00082**	**±0.00003**	–	**0.05879**	**±0.00083**	=	907721.87235	±503090.31807	+	0.05871	±0.00080	0.85870	±0.19637
15	WFG2	0.67912	±0.05525	=	**0.00082**	**±0.00003**	–	**0.69114**	**±0.05205**	=	26611.33190	±25193.59089	+	0.69113	±0.05203	0.85797	±0.34234
15	WFG3	0.14061	±0.02104	+	**0.00082**	**±0.00003**	–	**0.16198**	**±0.01944**	=	7091.19596	±13241.35130	+	0.16113	±0.01967	0.75482	±0.02921
15	WFG4	0.47475	±0.03276	+	**0.00083**	**±0.00004**	–	**0.52097**	**±0.03117**	=	701963.39534	±378065.47797	+	0.51678	±0.03041	0.59002	±0.05213
15	WFG5	0.38401	±0.02065	+	**0.00082**	**±0.00002**	–	**0.44142**	**±0.01504**	=	119015.48008	±204799.57140	+	0.43516	±0.01571	0.74370	±0.02826
15	WFG6	0.56713	±0.04370	+	**0.00082**	**±0.00003**	–	**0.61667**	**±0.04145**	=	161766.09125	±188396.91632	+	0.61407	±0.03985	0.58675	±0.09660
15	WFG7	0.35930	±0.01814	+	**0.00082**	**±0.00003**	–	**0.41198**	**±0.01766**	=	347654.60867	±529981.05882	+	0.40279	±0.01988	0.67919	±0.04909
15	WFG8	0.33059	±0.02538	+	**0.00082**	**±0.00003**	–	**0.38010**	**±0.01967**	=	151056.00468	±169033.02908	+	0.36997	±0.01853	0.83264	±1.19337
15	WFG9	0.31253	±0.02062	+	**0.00082**	**±0.00002**	–	**0.36172**	**±0.02264**	=	303512.07185	±280140.17679	+	0.35640	±0.02137	0.71350	±0.04720

Table 8: Holm-Bonferroni procedure

j	Method	Rank	z_j	p_j	δ/j	
HV value (reference: HAGA, Rank = 2.02e+00)						
1	CHV	$3.00e+00$	$4.64e+00$	$< e-05$	$5.00e-02$	H_A
2	RND	$1.02e+00$	$-4.74e+00$	$< e-0.5$	$2.50e-02$	H_A
T value (reference: HAGA, Rank = 1.60e+00)						
1	RND	$3.00e+00$	$6.64e+00$	$< e-05$	$2.50e-02$	H_A
2	CHV	$1.40e+00$	$-0.95e+00$	$0.17e00$	$5.00e-02$	H_0

of the objective space.

On the contrary, CHV performs a HV check of all the entire set of λ solutions. Analogous to what happens in single-objective optimisation when the dimensionality of the decision space is scaled up, the size of objective space exponentially grows with the number of objectives. Hence, the computational cost of CHV heavily depends on the number of objectives of the problem and grows exponentially with it. As shown in Table 7, the computational cost of CHV is very modest in the two-objective case. In the three-objective case the computational cost of CHV is about ten times higher than for a two-objective problem. In the five-objective case, the cost of CHV is over a second which may be already approaching a limitation for real-time applications (taking into account that a many-objective optimisation algorithm requires thousands of CHV selections). For a higher dimensionality of the objective space, the cost of CHV is very high making CHV unusable, or at least not very attractive. It can be observed that for a fifteen-objective problem a single CHV selection may require more than ten calculation days.

The proposed HAGA selection approach performs a search and has a non-null computational cost. Furthermore, the computational cost of HAGA depends on the number of objectives of the problem. However, its computational cost does not grow exponentially with the size of the objective space. On the contrary, the computational cost of HAGA grows extremely slowly when compared to the computational cost of CHV and is around, or within, a second for all of the problems considered in this study. Furthermore, HAGA appears to reliably offer a fairly high HV performance with a modest computational cost. In terms of its computational behaviour, HAGA can be regarded as a surrogate assisted strategy, e.g. see [61] and [69], that operates in the multi-dimensional objective space. For example a population selection for WFG1 for fifteen objectives can be solved by HAGA more than five million times faster that CHV selection still attaining a very good approximation of the theoretical HV accuracy (0.05871 instead of 0.05879), see Table 7.

These features make the proposed HAGA selection approach attractive for real-world application which require the solution of many-objective problems in a reasonable time.

Figure 12 depicts the computational cost depending on the number of objectives for the three selection schemes under consideration. It must be noted that in order to make the difference in cost between CHV and the other methods visible, the plot is represented in logarithmic scale. The trend displayed in Figure

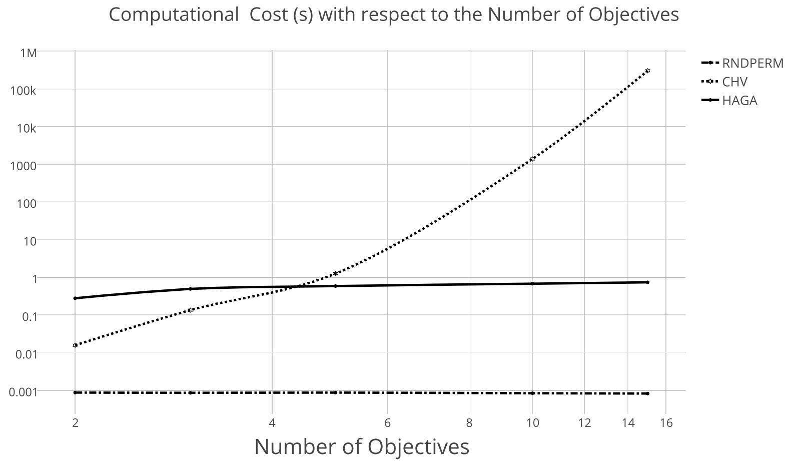


Figure 12: Computational cost ($\overline{T}$) of RNDPERM, CHV, and HAGA (logarithmic scale).

12 can be seen as a complexity analysis where the scalability is studied in the objective space instead of in the decision space.

Figure 13 depicts the HV indicator accuracy depending on the number of objectives for the three selection schemes under consideration. The HV indicator accuracy has been resolved using the procedure listed in Algorithm 4 in order to offer a normalised measure of performance between the considered selection approaches. The CHV selection approach scores each solution in the population based on the explicit area of the objective space which it dominates, with these scores it is possible to identify either the combination of solutions which cover the most objective space, and also the combination of solutions which cover the least of the objective space. With these two populations which dominate the most and least of the objective space, it is possible to use the HV indicator in each of them to resolve the worst achievable HV indicator score and the best achievable HV indicator score for the test case under consideration. It is possible to assume that any combination of μ solutions based on CHV indicator values from the λ population will score somewhere between these upper and lower achievable HV indicator values, and with this knowledge it is possible to arrive at a normalised (between 0 and 1) measure of performance of the considered selection approaches across all considered test cases. This HV indicator accuracy can also be seen as a percentage, whereby a HV indicator accuracy of 0% indicates the worst possible combination of μ solutions and a HV indicator accuracy of 100% indicates the best possible combination of μ solutions in regards to the HV indicator.

It can be observed in Figure 13 that the CHV selection approach scores a consistent HV indicator accuracy of 1 (or 100%), this is to be expected as theoretically the CHV indicator by design is able to order a population based on explicit domination of the objective space. The RNDPERM selection approach is shown to achieve a mean HV indicator accuracy of 0.7 (or 70%), with the lowest HV indicator accuracy scoring at 0.67 (or

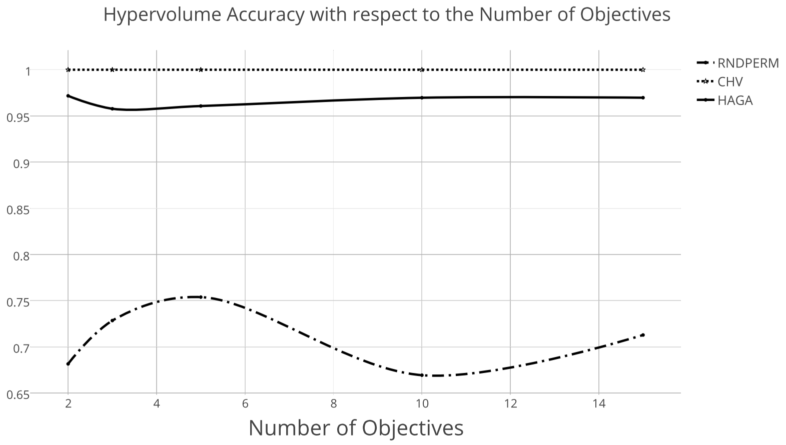


Figure 13: Hypervolume Accuracy ($\overline{A}$) of RNDPERM, CHV, and HAGA.

67%), this is an expected and undesirable score. If the RNDPERM approach to selection was employed throughout an optimisation process, it can be assumed that at each generation a new parent population of size μ would be selected from the offspring population of size λ, and that this new parent population would consist of solutions that on average represent 70% of the objective space which can be dominated by the offspring population. It can therefore be assumed that many of the solutions forming this new offspring population are: solutions which are dominated by other solutions in the offspring population; solutions which offer very little in terms of explicit contribution to the HV indicator; or identical solutions which offer nothing to the quality of the new parent population. Through this process of using the RNDPERM selection approach, it is possible that the HV indicator quality of populations resolved at each generation oscillate, or regress due to the lack of selection pressure. The HAGA selection approach is shown to achieve a mean HV indicator accuracy of 0.966 (or 96.6%), with no HV indicator accuracy score below 0.957 (or 95.7%), this performance remains consistent from problems consisting of two problem objectives up to at least fifteen problem objectives. An average HV indicator accuracy score of 96.6% suggests that regardless of the number of considered problem objectives, the HAGA approach to selection is able to offer great selection pressure towards dominant areas of the objective space.

4.4. Comparison against another approximated HV indicator

In order to to further prove the efficiency of the proposed HAGA, we have compared it with another selection method for many-objective optimisation based on an approximated estimation of the hypervolume indicator, i.e. HypE reported in [6]. The same experiments as above have been repeated and HAGA has the same parameters used in the previous sections. Regarding HypE, its parameter has been set to achieve an error of only 1% as suggested in its paper: number of samples 10^5. Table 9 displays the results of this experiment.

Numerical results in Table 9 show that HAGA and HypE achieve comparable results for all the numbers of objectives in terms of $\overline{HV}$. As shown, for all the experiments carried out there is neither of the method outperforms the other in terms hypervolume. On the other hand, the advantages of HAGA with respect to HypE are evident when the calculation time is considered. HAGA outperforms HypE in terms of calculation time in all instances by obtaining similar results in less than a fifth of the HypE calculation time.

5. Application to Neural Systems in Concealed Weapon Detection

The optimisation of the accuracy and efficiency of classifiers in pattern recognition is a complex problem that is often poorly understood. Whilst numerous techniques exist for the optimisation of weights in Artificial Neural Networks (ANNs) (such as the Widroff-Hoff least mean squares algorithm and back propagation), there do not exist any hard and fast rules for choosing the structure of an ANN - in particular for choosing both the size (in terms of the number of neurons) and the number of hidden layers used in the network. However, this internal structure is one of the key factors in determining the efficiency of the network and the accuracy of the classification.

This section presents the optimisation of the ANN architecture used for Concealed Weapon Detection (CWD) in a seven-objective problem. CWD is an important area of research in the defence and security community. This is due to a number of high profile terrorist attacks which have resulted in loss of life and damage to public infrastructure. The optimisation of concealed weapon detection classifiers is important because even a marginal gain in performance can improve the safety and security for the area in which the system is implemented.

The method here employed for detecting concealed weapons in crowded areas in real-time is by means of coordinated small portable radars. A number of radar systems have been developed for this purpose [9] and [70]. These radars use multiple methods of detecting concealed weapons such as time domain reflectometry and the exploitation of polarisation changes induced by complex objects concealed on the human torso under clothing.

The radar used in this work is constructed of a Vector Network Analyser (VNA) with pyramid horn antennas connected to the VNA using suitable cabling. A laptop is used to control the VNA and then classify the signals. The radar signals are analysed on the laptop using pattern recognition applied to the time resolved signals in the form of an Artificial Neural Network (ANN), this set-up has been illustrated in Figure 14. This method has been discussed in detail in a previous publication [71].

One of the shortcomings of this method is that the optimisation of the ANN architecture has previously been performed using trial and error. This has been done by increasing the number of hidden layers in the ANN and also increasing the number of neurons on each of these layers. A set of validation data was fed

Table 9: Mean values ± standard deviation values of hypervolume indicator and calculation time for HypE and HAGA in the two, three, five, ten, and fifteen-objective case. A value in bold indicates the highest performing approach for the corresponding performance metric. A greater HV score indicates greater hypervolume indicator performance, and a lesser T score indicates faster performance.

		HypE						HAGA			
M	Fn	$\overline{HV}$	$\pm\sigma$		$\overline{T}$	$\pm\sigma$		$\overline{HV}$	$\pm\sigma$	$\overline{T}$	$\pm\sigma$
2	WFG1	0.26113	±0.01285	=	1.82753	±0.13867	+	**0.26261**	**±0.01084**	**0.21728**	**±0.10040**
2	WFG2	0.63536	±0.01664	=	2.16441	±0.13458	+	**0.63622**	**±0.01623**	**0.05468**	**±0.00986**
2	WFG3	**0.63469**	**±0.00569**	=	1.95012	±0.11419	+	0.63460	±0.00753	**0.36607**	**±0.07442**
2	WFG4	0.46391	±0.00541	=	1.68275	±0.10467	+	**0.46529**	**±0.00457**	**0.06388**	**±0.01354**
2	WFG5	0.45122	±0.00517	=	1.70633	±0.11013	+	**0.45319**	**±0.00194**	**0.41180**	**±0.05766**
2	WFG6	0.49308	±0.00965	=	1.74978	±0.11795	+	**0.49418**	**±0.00830**	**0.57747**	**±0.06463**
2	WFG7	0.51148	±0.00438	=	1.78621	±0.12065	+	**0.51235**	**±0.00442**	**0.25407**	**±0.07760**
2	WFG8	**0.44846**	**±0.00531**	=	1.52428	±0.16853	+	0.44843	±0.00597	**0.23082**	**±0.06540**
2	WFG9	0.50159	±0.00406	=	1.28945	±0.03999	+	**0.50195**	**±0.00443**	**0.32247**	**±0.11638**
		HypE						HAGA			
M	Fn	$\overline{HV}$	$\pm\sigma$		$\overline{T}$	$\pm\sigma$		$\overline{HV}$	$\pm\sigma$	$\overline{T}$	$\pm\sigma$
3	WFG1	0.35022	±0.01226	=	2.19671	±0.10713	+	**0.35036**	**±0.01253**	**0.59704**	**±0.04166**
3	WFG2	0.82646	±0.01445	=	2.19649	±0.10913	+	**0.82688**	**±0.01454**	**0.42262**	**±0.04688**
3	WFG3	0.59555	±0.00984	=	1.97480	±0.12437	+	**0.59614**	**±0.01125**	**0.52284**	**±0.04523**
3	WFG4	0.55458	±0.01205	=	1.89898	±0.13124	+	**0.56026**	**±0.01301**	**0.33683**	**±0.04832**
3	WFG5	**0.57638**	**±0.00812**	=	1.86686	±0.14023	+	0.57434	±0.00965	**0.53927**	**±0.03212**
3	WFG6	0.61379	±0.00888	=	1.91159	±0.12394	+	**0.61525**	**±0.00834**	**0.60009**	**±0.02691**
3	WFG7	0.56819	±0.01339	=	2.02938	±0.19062	+	**0.56834**	**±0.01256**	**0.46005**	**±0.05513**
3	WFG8	0.51735	±0.01108	=	1.90151	±0.55118	+	**0.52137**	**±0.01184**	**0.45089**	**±0.05183**
3	WFG9	0.59525	±0.01100	=	1.55483	±0.12383	+	**0.59536**	**±0.00946**	**0.48412**	**±0.05889**
		HypE						HAGA			
M	Fn	$\overline{HV}$	$\pm\sigma$		$\overline{T}$	$\pm\sigma$		$\overline{HV}$	$\pm\sigma$	$\overline{T}$	$\pm\sigma$
5	WFG1	0.35593	±0.00490	=	3.70656	±0.32810	+	**0.35599**	**±0.00495**	**0.78154**	**±0.10825**
5	WFG2	0.84962	±0.01318	=	3.63484	±0.26045	+	**0.84966**	**±0.01317**	**0.75523**	**±0.13534**
5	WFG3	**0.53106**	**±0.00791**	=	3.20888	±0.37518	+	0.52522	±0.01116	**0.56657**	**±0.03422**
5	WFG4	0.62804	±0.01174	=	3.10994	±0.35588	+	**0.62973**	**±0.01319**	**0.44297**	**±0.04703**
5	WFG5	**0.67049**	**±0.01567**	=	3.02063	±0.30718	+	0.66853	±0.01631	**0.57354**	**±0.03647**
5	WFG6	0.72059	±0.01134	=	3.13576	±0.38586	+	**0.72322**	**±0.01302**	**0.59686**	**±0.02849**
5	WFG7	0.59135	±0.01316	=	3.46482	±0.99753	+	**0.59138**	**±0.01479**	**0.53652**	**±0.03597**
5	WFG8	0.52986	±0.02035	=	2.16481	±0.26168	+	**0.53255**	**±0.02056**	**0.46476**	**±0.05669**
5	WFG9	0.65638	±0.01398	=	2.05873	±0.16856	+	**0.65887**	**±0.01413**	**0.54510**	**±0.03100**
		HypE						HAGA			
M	Fn	$\overline{HV}$	$\pm\sigma$		$\overline{T}$	$\pm\sigma$		$\overline{HV}$	$\pm\sigma$	$\overline{T}$	$\pm\sigma$
10	WFG1	**0.20260**	**±0.00323**	=	6.89606	±1.31032	+	0.20253	±0.00325	**1.20437**	**±1.25042**
10	WFG2	**0.82300**	**±0.02876**	=	7.28971	±0.98003	+	0.82292	±0.02854	**0.83236**	**±0.53270**
10	WFG3	**0.39299**	**±0.01647**	=	6.51789	±1.17428	+	0.38837	±0.01741	**0.68271**	**±0.02954**
10	WFG4	0.56809	±0.02614	=	6.23768	±1.11081	+	**0.56968**	**±0.02868**	**0.49096**	**±0.07500**
10	WFG5	**0.53959**	**±0.02424**	=	6.39568	±1.21359	+	0.53862	±0.02456	**0.65928**	**±0.03369**
10	WFG6	0.71402	±0.01719	=	6.85013	±1.21046	+	**0.71643**	**±0.01901**	**0.58623**	**±0.06133**
10	WFG7	0.50415	±0.01786	=	6.82064	±1.46340	+	**0.50482**	**±0.01700**	**0.59102**	**±0.04497**
10	WFG8	**0.43684**	**±0.02318**	=	6.02258	±2.17669	+	0.43631	±0.02313	**0.44876**	**±0.05142**
10	WFG9	0.51419	±0.03100	=	3.64024	±0.45495	+	**0.51589**	**±0.03149**	**0.59211**	**±0.03683**
		HypE						HAGA			
M	Fn	$\overline{HV}$	$\pm\sigma$		$\overline{T}$	$\pm\sigma$		$\overline{HV}$	$\pm\sigma$	$\overline{T}$	$\pm\sigma$
15	WFG1	0.05870	±0.00085	=	10.77472	±1.65668	+	**0.05871**	**±0.00080**	**0.85870**	**±0.19637**
15	WFG2	0.68984	±0.05154	=	10.00911	±1.26968	+	**0.69113**	**±0.05203**	**0.85797**	**±0.34234**
15	WFG3	**0.16148**	**±0.01952**	=	9.54261	±1.58181	+	0.16113	±0.01967	**0.75482**	**±0.02921**
15	WFG4	**0.51709**	**±0.02985**	=	9.32213	±1.52301	+	0.51678	±0.03041	**0.59002**	**±0.05213**
15	WFG5	**0.43823**	**±0.01505**	=	9.29506	±1.55050	+	0.43516	±0.01571	**0.74370**	**±0.02826**
15	WFG6	0.60976	±0.04341	=	9.70836	±1.70848	+	**0.61407**	**±0.03985**	**0.58675**	**±0.09660**
15	WFG7	**0.40807**	**±0.01708**	=	9.79999	±1.58785	+	0.40279	±0.01988	**0.67919**	**±0.04909**
15	WFG8	**0.37620**	**±0.01978**	=	10.21248	±4.10443	+	0.36997	±0.01853	**0.83264**	**±1.19337**
15	WFG9	**0.35752**	**±0.02055**	=	5.17540	±0.25897	+	0.35640	±0.02137	**0.71350**	**±0.04720**

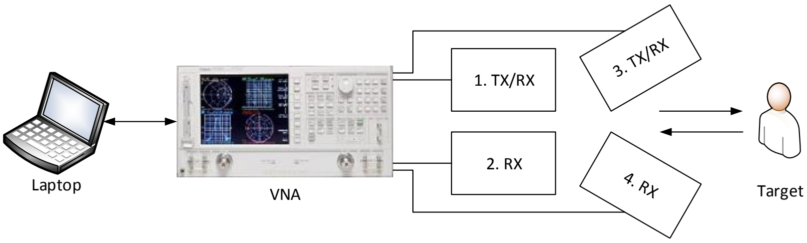


Figure 14: System block diagram of the arrangement of the transmitter and receiver horn antennas.

into the ANN each time a new layer or neuron was added, and the true positive and false positive rates were recorded. The best architecture was selected by weighing the achieved true positive rate against the cost in false alarm rate. Another issue exists with the training of the ANN, which tends to be inconsistent. This is caused when an initial guess at the weights and biases is taken. As a result of the randomness of this guess the convergence can be to a local minima rather than the global.

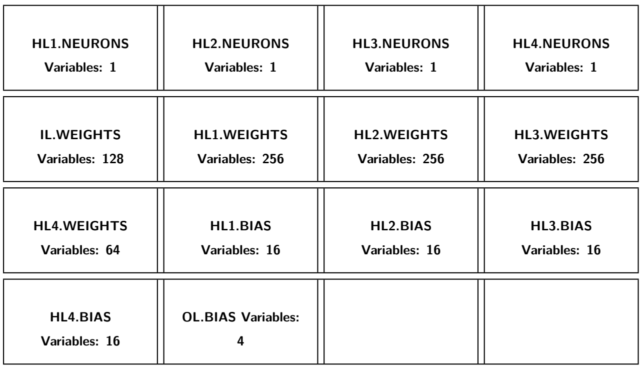
It is of great importance that the false alarm rate is kept low, typically below a few percent, for security screening of large volumes of people. This is due to the action that must be taken once a potential threat has been identified. This action could range from further investigation, e.g. stop and search, to the evacuation of a crowded public area. If the false positive rate goes above a few percent the inconvenience to the security forces and general public would render the method ineffective. Therefore the primary objective in optimising the ANN architecture must be the reduction of the false positive rate.

The second objective in the optimisation of the ANN architecture must be the preservation of the true positive rate for targets of interest. The targets that should be detected by the radar include knives and guns. It is unfortunate that knives and guns are seized by the security forces far too frequently and pose a significant threat to the safety of the general public. The damage that can be caused with these weapons is considerable and these targets are easily concealed upon the human body.

A weapon detection system that is capable of classifying a detected threat into target groups would be an extremely valuable tool to security forces. Such a system would enable the authorities to react to a detection in a controlled and proportional manner. The action that must be taken to confront an individual concealing a threat object depends very much on the threat object itself. An individual carrying a knife could be dealt with easier than an individual with an improvised explosive device.

Besides obvious threat objects such as guns and knives, the detection of objects such as mobile phones is desirable as they can be seen as a threat in the case of a controlled courtroom (where photographing witnesses and communicating with witnesses waiting to testify is an issue) where they are banned and when entering a controlled site.

The extent of the ROI will be determined by the DM based on some pre-determined criteria, for example the radar may be deployed in an environment where the client has specified a minimum detection rate and



HL1.NEURONS Variables: 1	HL2.NEURONS Variables: 1	HL3.NEURONS Variables: 1	HL4.NEURONS Variables: 1
IL.WEIGHTS Variables: 128	HL1.WEIGHTS Variables: 256	HL2.WEIGHTS Variables: 256	HL3.WEIGHTS Variables: 256
HL4.WEIGHTS Variables: 64	HL1.BIAS Variables: 16	HL2.BIAS Variables: 16	HL3.BIAS Variables: 16
HL4.BIAS Variables: 16	OL.BIAS Variables: 4		

Figure 15: Encoded chromosome for the seven-objective ANN consisting of 4 hidden layers (HL), input layer (IL), 4 neurons on the output layer (OL), and associated biases, totalling to 1032 variables.

maximum false alarm rate. It is possible that no solutions may be found within an ROI which has been confined based on a client's specifications, in which case the specification would be deemed beyond the performance of the radar and another solution would be required.

5.1. Encoding the Problem

In order to use evolutionary optimisation to optimise the topology and weights of the ANN classifier for concealed weapon detection, the ANNs topology and weights must be encoded into a real-valued chromosome, which can then be subjected to the various evolutionary operators used in the optimisation process and then decoded for evaluation. Figure 15 illustrates the chromosome structure used to store the problem parameters for an ANN with 4 output neurons, a maximum of 4 layers, and 16 input neurons.

Regardless of the topology of the candidate solution ANN (which in this case is defined by the first 2 genes of the encoded chromosome) the maximum number of weights and biases will be stored with each chromosome, however not all genotypes will manifest and be expressed as phenotypes as only the weights and biases required to configure the candidate solutions ANN topology will be decoded and used. These unused weights and biases will remain unexpressed in the phenotype until the first two genes allow them to manifest and can go through many generations as dormant genes. This introduces the interesting feature of atavism[1] into this problem.

At each function evaluation, a chromosome is decoded from its encoded state described in Figure 15 and used to instantiate an ANN. This ANN is then used to classify the training data and the objective information is extracted and used to assess the chromosome's fitness based on the ANN result set.

[1] "Atavism is the tendency to revert to ancestral type. In biology, an atavism is an evolutionary throwback, such as traits reappearing which had disappeared generations before."

5.2. Experiment

The proposed solution in the experiment uses an implementation of CMA-PAES-II enabled with the ability to express DM preferences named WZ-CMA-PAES. The algorithm itself is described and benchmarked in [75] and [76] where it is shown to perform well on problems containing many objectives, this is a requirement for this problem as it consists of seven problem objectives. The configurations for WZ-CMA-PAES have been listed in Table 10.

Table 10: Parameter configurations used for the WZ-CMA-PAES algorithm.

Parameter	**Configuration**
Archive Capacity	50
Grid Divisions	3
μ Population	10
Number of Variables	1032
Number of Objectives	7
Z_{thresh}	50
Function Evaluation Limit	1000
Region of Interest	0.05, 0.4, 0.5, 0.5, 0.05, 0.05, 0.05

WZ-CMA-PAES was selected to allow the optimisation process to search within a ROI with the True Positive (TP) and False Positive (FP) preferences defined as:

- TP rate on classification of bodies $\geq$ 95%.
- TP rate on classification of guns $\geq$ 60%.
- TP rate on classification of knives $\geq$ 50%.
- TP rate on classification of mobile phones $\geq$ 50%.
- FP rate on classification of guns $\leq$ 5%.
- FP rate on classification of knives $\leq$ 5%.
- FP rate on classification of mobile phones $\leq$ 5%.

The HAGA approach to selection introduced in this paper has been used as the method of selection throughout the optimisation process. The ROI is used as the reference point that is provided to HAGA allowing selection based on how much of the search space within the ROI has been covered. HAGA is also

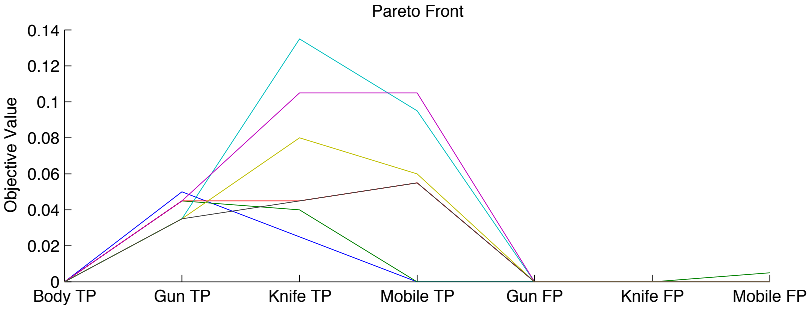


Figure 16: Parallel-coordinates plot of objective value results for seven-objective threat detection.

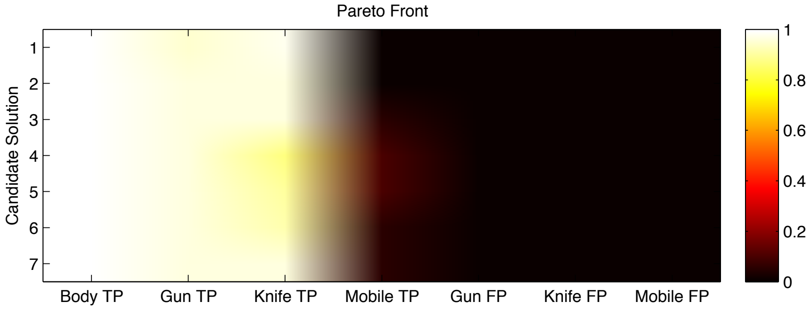


Figure 17: Colour-map of objective value results for seven-objective threat detection, for use by the DM.

used to reduce the size of the final approximation set produced by the optimiser to a size that will not overwhelm and confuse the DM.

The population of solutions from the WZ-CMA-PAES optimisation are plotted in Figure 16, this plot shows the solutions for the seven-objective problem. The seven objectives are split into true positives for the body, gun, knife, mobile phone and the false positives for the gun, knife, and mobile. In this problem the reduction of false positive rates are the three main objectives, these should be weighted equally. The remaining objectives are the maximization of the true positive rates for each target included in the training and validation sets. Also presented is a colour-map, Figure 17, this was used to aid the DM in choosing a solution from the population. In each of the presented plots there are 7 candidate solutions, one of which should be chosen by the DM to be implemented in the weapon detection system.

6. Conclusion

This paper proposes a novel mechanism for population selection in many-objective optimisation. The proposed method overcomes the time limitation of the CHV indicator calculation, especially when more than five problem objectives are considered, without significantly compromising on the quality of selected solutions in regards to their HV indicator quality. Extensive experiments have been carried out by studying the scalability of the objective space ranging from the two-objective case to the fifteen-objective case. Numerical results show that whilst the proposed Hypervolume Adaptive Grid Algorithm (HAGA) is not competitive in contrast to the CHV indicator when only a few problem objectives are considered, HAGA achieves a reliable accuracy (over 95% of the theoretically achievable value by CHV) when five or more problem objectives are considered, often being one million times faster than the CHV indicator calculation. In practice this means that HV indicator driven selection in the many-objective case can be performed in less than a second instead of days of calculation. This feature makes the proposed approach especially attractive for real-time applications or, more generally, applications that require the results in a reasonable time.

The proposed selection mechanism has been successfully integrated within the WZ-CMA-PAES optimisation algorithm to the training and optimisation of an Artificial Neural Network topology or use in concealed weapon detection. The optimisation has been shown to provide the DM with a number of solutions (trained ANNs) that all have independent trade-offs which are equally distributed across the Pareto-optimal front. The DM then selected an optimised solution which provided a reduction in false alarm rate and an increase in detection rate.

All of the hypotheses defined in Section 4 of this paper have been experimentally confirmed.

Future works will consider the integration of information theory techniques within our framework. More specifically, future works will include an estimation of Shannon-like entropy of the population in order to enhance the accuracy within the selection process.

References

[1] S. F. Adra, T. J. Dodd, I. A. Griffin, P. J. Fleming, Convergence acceleration operator for multiobjective optimization, IEEE Transactions on Evolutionary Computation 13 (4) (2009) 825–847.

[2] I. Alberto, C. A. C. Coello, P. M. Mateo, A comparative study of variation operators used for evolutionary multi-objective optimization, Information Sciences 273 (0) (2014) 33 – 48.

[3] M. Antonelli, P. Ducange, F. Marcelloni, A fast and efficient multi-objective evolutionary learning scheme for fuzzy rule-based classifiers, Information Sciences 283 (0) (2014) 36 – 54.

[4] A. Auger, J. Bader, D. Brockhoff, E. Zitzler, Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications, Theoretical Computer Science 425 (2012) 75–103.

[5] J. Bader, K. Deb, E. Zitzler, Faster hypervolume-based search using monte carlo sampling, in: Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems, Springer, 2010, pp. 313–326.

[6] J. Bader, E. Zitzler, Hype: An algorithm for fast hypervolume-based many-objective optimization, Evolutionary Computation 19 (1) (2011) 45–76.

[7] R. L. Becerra, C. A. Coello Coello, Solving hard multiobjective optimization problems using ε-constraint with cultured differential evolution, in: Parallel Problem Solving from Nature-PPSN IX, Springer, 2006, pp. 543–552.

[8] N. Beume, B. Naujoks, M. Emmerich, SMS-EMOA: Multiobjective selection based on dominated hypervolume, European Journal of Operational Research 181 (3) (2007) 1653 – 1669.

[9] N. Bowring, M. Southgate, D. Andrews, N. D. Rezgui, S. Harmer, D. O'Reilly, Development of a longer range standoff

millimetre wave radar concealed threat detector, in: Proc. SPIE 8714, Radar Sensor Technology XVII, SPIE, 2013, pp. 1–8.

[10] K. Bringmann, T. Friedrich, Approximating the least hypervolume contributor: Np-hard in general, but fast in practice, in: Evolutionary Multi-Criterion Optimization, Springer, 2009, pp. 6–20.

[11] K. Bringmann, T. Friedrich, C. Igel, T. Voß, Speeding up many-objective optimization by monte carlo approximations, Artificial Intelligence 204 (2013) 22–29.

[12] F. Caraffini, F. Neri, L. Picinali, An analysis on separability for memetic computing automatic design, Information Sciences 265 (2014) 1–22.

[13] R. Cheng, Y. Jin, K. Narukawa, B. Sendhoff, A multiobjective evolutionary algorithm using gaussian process-based inverse modeling, IEEE Transactions on Evolutionary Computation 19 (6) (2015) 838–856.

[14] R. Cheng, Y. Jin, M. Olhofer, B. Sendhoff, A reference vector guided evolutionary algorithm for many-objective optimization, IEEE Transactions on Evolutionary Computation PP (99) (2016) 1–1, to appear.

[15] R. Cheng, M. Olhofer, Y. Jin, Reference vector based a posteriori preference articulation for evolutionary multiobjective optimization, in: IEEE Congress on Evolutionary Computation, 2015, pp. 939–946.

[16] C. A. C. Coello, Evolutionary multi-objective optimization: some current research trends and topics that remain to be explored, Frontiers of Computer Science in China 3 (1) (2009) 18–30.

[17] C. A. Coello Coello, Research directions in evolutionary multi-objective optimization, Evolutionary Computation Journal 3 (3) (2012) 110–121.

[18] C. A. Coello Coello, G. B. Lamont, Applications of multi-objective evolutionary algorithms, vol. 1, World Scientific, 2004.

[19] I. De Falco, A. Della Cioppa, D. Maisto, U. Scafuri, E. Tarantino, Biological invasion–inspired migration in distributed evolutionary algorithms, Information Sciences 207 (2012) 50–65.

[20] K. Deb, Multi-objective optimization, Multi-objective optimization using evolutionary algorithms (2001) 13–46.

[21] K. Deb, 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 18 (4) (2014) 577–601.

[22] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197.

[23] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm and Evolutionary Computation 1 (1) (2011) 3–18.

[24] J. J. Durillo, J. García-Nieto, A. J. Nebro, C. A. Coello Coello, F. Luna, E. Alba, Multi-objective particle swarm optimizers: An experimental comparison, in: Evolutionary Multi-Criterion Optimization, Springer, 2009, pp. 495–509.

[25] A. E. Eiben, J. E. Smith, Introduction to Evolutionary Computation, Springer-Verlag, Berlin, Germany, 2003.

[26] M. Emmerich, N. Beume, B. Naujoks, An emo algorithm using the hypervolume measure as selection criterion, in: Evolutionary Multi-Criterion Optimization, Springer, 2005, pp. 62–76.

[27] M. Epitropakis, V. Plagianakos, M. Vrahatis, Evolving cognitive and social experience in particle swarm optimization through differential evolution: A hybrid approach, Information Sciences: an International Journal 216 (2012) 50–92.

[28] M. Farina, P. Amato, On the optimal solution definition for many-criteria optimization problems, in: J. Keller, O. Nasraoui (eds.), Proceedings of the NAFIPS-FLINT International Conference, 2002, pp. 233 – 238.

[29] C. M. Fonseca, P. J. Fleming, Multiobjective optimization and multiple constraint handling with evolutionary algorithms. i. a unified formulation, IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 28 (1) (1998) 26–37.

[30] C. M. Fonseca, L. Paquete, M. López-Ibánez, An improved dimension-sweep algorithm for the hypervolume indicator, in: IEEE Congress on Evolutionary Computation, 2006., 2006, pp. 1157–1163.

[31] S. Garcia, A. Fernandez, J. Luengo, F. Herrera, A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability, Soft Computing 13 (10) (2008) 959–977.

[32] S. García, D. Molina, M. Lozano, F. Herrera, A study on the use of non-parametric tests for analyzing the evolutionary algorithms behaviour: a case study on the cec2005 special session on real parameter optimization, Journal of Heuristics 15 (6) (2009) 617–644.

[33] F. W. Gembicki, Vector optimization for control with performance and parameter sensitive indices, Ph.D. thesis, Case Western Reserve University, Cleveland, Ohio (1974).

[34] M. Helbig, A. P. Engelbrecht, Performance measures for dynamic multi-objective optimisation algorithms, Information Sciences 250 (2013) 61–81.

[35] S. Holm, A simple sequentially rejective multiple test procedure, Scandinavian Journal of Statistics 6 (2) (1979) 65–70.

[36] E. J. Hughes, Evolutionary many-objective optimisation: many once or one many?, in: IEEE Congress on Evolutionary Computation, 2005., vol. 1, IEEE, 2005, pp. 222–227.

[37] E. J. Hughes, Msops-ii: A general-purpose many-objective optimiser, in: IEEE Congress on Evolutionary Computation, 2007., IEEE, 2007, pp. 3944–3951.

[38] C.-L. Hwang, A. S. M. Masud, Multiple Objective Decision Making - Methods and Applications, vol. 164 of Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, Berlin, 1979.

[39] C. Igel, N. Hansen, S. Roth, Covariance matrix adaptation for multi-objective optimization, Evolutionary computation 15 (1) (2007) 1–28.

[40] H. Ishibuchi, N. Akedo, Y. Nojima, Behavior of multiobjective evolutionary algorithms on many-objective knapsack problems, IEEE Transactions on Evolutionary Computation 19 (2) (2015) 264–283.

[41] H. Ishibuchi, N. Tsukamoto, Y. Hitotsuyanagi, Y. Nojima, Effectiveness of scalability improvement attempts on the performance of NSGA-II for many-objective problems, in: Proceedings of the 10th annual conference on Genetic and evolutionary computation, ACM, 2008, pp. 649–656.

[42] H. Ishibuchi, N. Tsukamoto, Y. Nojima, Evolutionary many-objective optimization: A short review., in: IEEE Congress on Evolutionary Computation, 2008, pp. 2419–2426.

[43] H. Ishibuchi, N. Tsukamoto, Y. Sakane, Y. Nojima, Indicator-based evolutionary algorithm with hypervolume approximation by achievement scalarizing functions, in: Proceedings of the 12th annual conference on Genetic and evolutionary computation, ACM, 2010, pp. 527–534.

[44] H. Ishibuchi, T. Yoshida, T. Murata, Balance between Genetic Search and Local Search in Memetic Algorithms for Multiobjective permutation Flow shop Scheduling, IEEE Transactions on Evolutionary Computation 7 (2) (2003) 204–223.

[45] H. Jain, K. Deb, An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: Handling constraints and extending to an adaptive approach, IEEE Transactions on Evolutionary Computation 18 (4) (2014) 602–622.

[46] A. Jaszkiewicz, On the computational efficiency of multiple objective metaheuristics. the knapsack problem case study, European Journal of Operational Research 158 (2) (2004) 418–433.

[47] S. Jiang, S. Yang, Convergence versus diversity in multiobjective optimization, in: Proceedings of the 14th International Conference on Parallel Problem Solving from Nature (PPSN XIV), vol. 9921.

[48] S. Jiang, S. Yang, A strength Pareto evolutionary algorithm based on reference direction for multi-objective and many-objective optimization, IEEE Transactions on Evolutionary Computation,To appear.

[49] Y. Jin, B. Sendhoff, Connectedness, regularity and the success of local search in evolutionary multi-objective optimization,

in: Evolutionary Computation, 2003. CEC '03. The 2003 Congress on, vol. 3, 2003, pp. 1910–1917.

[50] V. Khare, X. Yao, K. Deb, Performance scaling of multi-objective evolutionary algorithms, in: Evolutionary Multi-Criterion Optimization, Springer, 2003, pp. 376–390.

[51] J. Knowles, Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems, IEEE Transactions on Evolutionary Computation 10 (1) (2006) 50–66.

[52] J. Knowles, D. Corne, The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimisation, in: Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on, vol. 1, IEEE, 1999.

[53] J. Knowles, D. Corne, Quantifying the effects of objective space dimension in evolutionary multiobjective optimization, in: Proceedings of the 4th International Conference on Evolutionary Multi-criterion Optimization, EMO'07, Springer-Verlag, 2007, pp. 757–771.

[54] A. Lara, G. Sanchez, C. A. Coello Coello, O. Schutze, Hcs: a new local search strategy for memetic multiobjective evolutionary algorithms, IEEE Transactions on Evolutionary Computation 14 (1) (2010) 112–132.

[55] H. Lebesgue, Intégrale, longueur, aire, Annali di matematica pura ed applicata 7 (1) (1902) 231–359.

[56] B. Li, J. Li, K. Tang, X. Yao, Many-objective evolutionary algorithms: A survey, ACM Comput. Surv. 48 (1) (2015) 13:1–13:35.

[57] M. Li, S. Yang, X. Liu, Diversity comparison of Pareto front approximations in many-objective optimization, IEEE Transactions on Cybernetics 44 (12) (2014) 2568–2584.

[58] M. Li, S. Yang, X. Liu, Shift-based density estimation for Pareto-based algorithms in many-objective optimization, IEEE Transactions on Evolutionary Computation 18 (3) (2014) 348–365.

[59] M. Li, S. Yang, X. Liu, A performance comparison indicator for Pareto front approximations in many-objective optimization, in: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, GECCO '15, ACM, 2015, pp. 703–710.

[60] M. Li, J. Zheng, K. Li, Q. Yuan, R. Shen, Enhancing diversity for average ranking method in evolutionary many-objective optimization, in: Parallel Problem Solving from Nature, vol. 6238 of Lecture Notes in Computer Science, 2010, pp. 647–656.

[61] D. Lim, Y. Jin, Y.-S. Ong, B. Sendhoff, Generalizing surrogate-assisted evolutionary computation, IEEE Transactions on Evolutionary Computation 14 (3) (2010) 329–355.

[62] A. López-Jaimes, C. A. Coello Coello, Including preferences into a multiobjective evolutionary algorithm to deal with many-objective engineering optimization problems, Information Sciences 277 (2014) 1–20.

[63] A. Lpez-Jaimes, C. A. C. Coello, Including preferences into a multiobjective evolutionary algorithm to deal with many-objective engineering optimization problems, Information Sciences 277 (0) (2014) 1 – 20.

[64] E. Mezura-Montes, M. Reyes-Sierra, C. A. Coello Coello, Multi-Objective Optimization using Differential Evolution: A Survey of the State-of-the-Art, in: U. K. Chakraborty (ed.), Advances in Differential Evolution, vol. 143 of Studies in Computational Intelligence, Springer, 2008, pp. 173–196.

[65] K. Miettinen, Nonlinear multiobjective optimization, vol. 12, Springer, 1999.

[66] B. Mirtich, Efficient algorithms for two-phase collision detection, Practical motion planning in robotics: current approaches and future directions (1997) 203–223.

[67] A. J. Nebro, J. Durillo, J. Garcia-Nieto, C. A. Coello Coello, F. Luna, E. Alba, Smpso: A new pso-based metaheuristic for multi-objective optimization, in: Computational intelligence in miulti-criteria decision-making, 2009. mcdm'09. ieee symposium on, IEEE, 2009, pp. 66–73.

[68] A. J. Nebro, J. J. Durillo, C. A. Coello Coello, F. Luna, E. Alba, A study of convergence speed in multi-objective metaheuristics, in: Parallel Problem Solving from Nature–PPSN X, Springer, 2008, pp. 763–772.

[69] F. Neri, X. del Toro Garcia, G. L. Cascella, N. Salvatore, Surrogate assisted local search on PMSM drive design, COMPEL:

International Journal for Computation and Mathematics in Electrical and Electronic Engineering 27 (3) (2008) 573–592.

[70] D. O'Reilly, N. Bowring, S. Harmer, Signal processing techniques for concealed weapon detection by use of neural networks, in: IEEE 27th Convention of Electrical and Electronic Engineers in Israel IEEE, IEEE, 2012, pp. 1–4.

[71] D. O'Reilly, N. Bowring, N. D. Rezgui, D. Andrews, S. Harmer, Remote concealed threat detection by novel classification algorithms applied to multipolarimetric uwb radar, in: Proc. SPIE 8714, Radar Sensor Technology XVII, SPIE, 2013, pp. 1–8.

[72] R. C. Purshouse, On the evolutionary optimisation of many objectives, Ph.D. thesis, Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, UK, S1 3JD (2003).

[73] R. C. Purshouse, P. J. Fleming, On the evolutionary optimization of many conflicting objectives, IEEE Transactions on Evolutionary Computation 11 (6) (2007) 770–784.

[74] R. C. Purshouse, C. Jalbă, P. J. Fleming, Preference-driven co-evolutionary algorithms show promise for many-objective optimisation, in: Evolutionary Multi-Criterion Optimization, Springer, 2011, pp. 136–150.

[75] S. Rostami, Preference focussed many-objective evolutionary computation, Ph.D. thesis, School of Engineering, Manchester Metropolitan University, Manchester, UK, M15 6HB (2014).

[76] S. Rostami, D. OReilly, A. Shenfield, N. Bowring, A novel preference articulation operator for the evolutionary multi-objective optimisation of classifiers in concealed weapons detection, Information Sciences 295 (0) (2015) 494 – 520.

[77] K. Sindhya, K. Miettinen, K. Deb, A hybrid framework for evolutionary multi-objective optimization, IEEE Transactions on Evolutionary Computation 17 (4) (2013) 495–511.

[78] K. C. Tan, S. C. Chiam, A. A. Mamun, C. K. Goh, Balancing exploration and exploitation with adaptive variation for evolutionary multi-objective optimization, European Journal of Operational Research 197 (2009) 701–713.

[79] K. C. Tan, E. F. Khor, T. H. Lee, Multiobjective Evolutionary Algorithms and Applications: Algorithms and Applications, Springer, 2006.

[80] Y. Tang, P. Reed, T. Wagener, et al., How effective and efficient are multiobjective evolutionary algorithms at hydrologic model calibration?, Hydrology and Earth System Sciences Discussions 10 (2) (2006) 289–307.

[81] K. Van Moffaert, M. M. Drugan, A. Nowé, Hypervolume-based multi-objective reinforcement learning, in: Evolutionary Multi-Criterion Optimization, Springer, 2013, pp. 352–366.

[82] T. Voß, N. Hansen, C. Igel, Improved step size adaptation for the MO-CMA-ES, in: Proceedings of the 12th annual conference on Genetic and evolutionary computation, ACM, 2010, pp. 487–494.

[83] M. Wagner, F. Neumann, A fast approximation-guided evolutionary multi-objective algorithm, in: Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation, GECCO '13, ACM, 2013, pp. 687–694.

[84] T. Wagner, N. Beume, B. Naujoks, Pareto-, aggregation-, and indicator-based methods in many-objective optimization, in: Evolutionary Multi-Criterion Optimization, vol. 4403 of Lecture Notes in Computer Science, 2007, pp. 742–756.

[85] D. J. Walker, R. Everson, J. E. Fieldsend, Visualizing mutually nondominating solution sets in many-objective optimization, IEEE Transactions on Evolutionary Computation 17 (2) (2013) 165–184.

[86] Y. Wang, Z. Cai, Q. Zhang, Differential evolution with composite trial vector generation strategies and control parameters, IEEE Transactions on Evolutionary Computation 15 (1) (2011) 55–66.

[87] L. While, L. Bradstreet, L. Barone, A fast way of calculating exact hypervolumes, IEEE Transactions on Evolutionary Computation 16 (1) (2012) 86–95.

[88] L. While, P. Hingston, L. Barone, S. Huband, A faster algorithm for calculating hypervolume, IEEE Transactions on Evolutionary Computation 10 (1) (2006) 29–38.

[89] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics bulletin (1945) 80–83.

[90] Q. Yang, S. Ding, Novel algorithm to calculate hypervolume indicator of Pareto approximation set, in: Advanced Intelligent Computing Theories and Applications. With Aspects of Contemporary Intelligent Computing Techniques,

Springer, 2007, pp. 235–244.

[91] S. Yang, M. Li, X. Liu, J. Zheng, A grid-based evolutionary algorithm for many-objective optimization, IEEE Transactions on Evolutionary Computation 17 (5) (2013) 721–736.

[92] Z. Yang, K. Tang, X. Yao, Large scale evolutionary optimization using cooperative coevolution, Information Sciences 178 (15) (2008) 2985–2999.

[93] Y. Yuan, H. Xu, B. Wang, X. Yao, A new dominance relation-based evolutionary algorithm for many-objective optimization, IEEE Transactions on Evolutionary Computation 20 (1) (2016) 16–37.

[94] A. Zamuda, J. Brest, B. Boskovic, V. Zumer, Differential evolution for multiobjective optimization with self adaptation., in: IEEE Congress on Evolutionary Computation, 2007, pp. 3617–3624.

[95] Q. Zhang, H. Li, Moea/d: A multiobjective evolutionary algorithm based on decomposition, IEEE Transactions on Evolutionary Computation 11 (6) (2007) 712–731.

[96] Q. Zhang, W. Liu, H. Li, The performance of a new version of moea/d on cec09 unconstrained mop test instances, in: IEEE Congress on Evolutionary Computation, 2009. CEC'09., IEEE, 2009, pp. 203–208.

[97] X. Zhang, Y. Tian, R. Cheng, Y. Jin, An efficient approach to nondominated sorting for evolutionary multiobjective optimization, IEEE Transactions on Evolutionary Computation 19 (2) (2015) 201–213.

[98] X. Zhang, Y. Tian, Y. Jin, A knee point-driven evolutionary algorithm for many-objective optimization, IEEE Transactions on Evolutionary Computation 19 (6) (2015) 761–776.

[99] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, Q. Zhang, Multiobjective evolutionary algorithms: A survey of the state of the art, Swarm and Evolutionary Computation 1 (1) (2011) 32–49.

[100] E. Zitzler, S. Künzli, Indicator-based selection in multiobjective search, in: Parallel Problem Solving from Nature-PPSN VIII, Springer, 2004, pp. 832–842.

[101] E. Zitzler, L. Thiele, An evolutionary algorithm for multiobjective optimization: The strength Pareto approach, IEEE Transactions on Evolutionary Computation 3 (4) (1998) 257–271.

[102] E. Zitzler, L. Thiele, M. Laumanns, C. M., V. G. da Fonseca, Performance assessment of multiobjective optimizers: An analysis and review, IEEE Transactions on Evolutionary Computation 7 (2) (2003) 117–132.

[103] X. Zou, Y. Chen, M. Liu, L. Kang, A new evolutionary algorithm for solving many-objective optimization problems, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics.