Bui LT, Deb K, Abbass HA *et al.* Interleaving guidance in evolutionary multi-objective optimization. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 23(1): 44–63 Jan. 2008

Interleaving Guidance in Evolutionary Multi-Objective Optimization

Lam Thu Bui[1], Kalyanmoy Deb[2], Hussein A. Abbass[1], and Daryl Essam[1]

[1] *The Artificial Life and Adaptive Robotics Laboratory, School of ITEE, ADFA, University of New South Wales Canberra, ACT, 2600, Australia*

[2] *Mechanical Engineering Department, Indian Institute of Technology, Kanpur, PIN 208 016, India*

E-mail: Lam.bui07@gmail.com; deb@iitk.ac.in; {h.abbass, d.essam}@adfa.edu.au

Revised November 19, 2007.

Abstract In this paper, we propose a framework that uses localization for multi-objective optimization to simultaneously guide an evolutionary algorithm in both the decision and objective spaces. The localization is built using a limited number of adaptive spheres (local models) in the decision space. These spheres are usually guided, using some direction information, in the decision space towards the areas with non-dominated solutions. We use a second mechanism to adjust the spheres to specialize on different parts of the Pareto front by using a guided dominance technique in the objective space. Through this interleaved guidance in both spaces, the spheres will be guided towards different parts of the Pareto front while also exploring the decision space efficiently. The experimental results showed good performance for the local models using this dual guidance, in comparison with their original version.

Keywords evolutionary multi-objective optimization, guided dominance, local models

1 Introduction

Evolutionary multi-objective optimization (EMO) has been applied in numerous domains[1−4] with a wide range of algorithms, such as the Vector Evaluated Genetic Algorithm (VEGA)[5], the Strength Pareto Evolutionary Algorithm version 2 (SPEA2)[6], Pareto Differential Evolution (PDE)[7] and Non-dominated Sorting Genetic Algorithm version 2 (NSGA-II)[8]. Multi-objective evolutionary algorithms (MOEAs) are usually blind search techniques in the sense that they do not usually use any auxiliary functions like derivatives (as in traditional deterministic optimization techniques). To reduce the effect of this "blindness", there has been an increasing number of attempts to incorporate various guidance capabilities into MOEAs. Basically, to do this, some sort of guidance is employed to direct the search towards promising areas satisfying specific criteria, such as avoiding infeasible areas or of approaching particular parts of the Pareto front. Further, guidance can be done in either the decision or objective space.

In this paper, we hypothesize that interleaving guidance in both the decision and objective spaces can help accelerating the search process. Our proposed idea is to localize the search in the decision space by using a framework of local models[9] that divides the decision search space into a number of localized search areas, where each area can be (but not limited to being) seen as a hyper-sphere. In other words, we transform searching in MOEAs on the original search space into a sphere-oriented space, where each sphere is running its own version of an MOEA. These spheres move following some direction information to improve their local non-dominated sets. These spheres then tend to specialize and move towards different parts of the Pareto Optimal Front (POF) by using a guidance technique in the objective space called guided dominance[10]. In general, the main contributions of this paper are as follows.

- Analysis of the performance of guidance in the decision space.
- Analysis of the performance of guidance in the objective space.
- Introduction to a method of interleaved guidance in evolutionary multi-objective optimization based on local models.

The remainder of the paper is organized as follows. Background information is presented in Section 2. The methodology is given in Section 3. An experimental study is carried out in Section 4, and the last section

Regular Paper

This work is supported by the Australian Research Council (ARC) Centre for Complex Systems under Grant No. CEO0348249 and the Postgraduate Research Student Overseas Grant from UNSW@ADFA, University of New South Wales.

is devoted to the conclusion.

2 Background

2.1 Evolutionary Multi-Objective Optimization

In a k-objective optimization problem, a vector function $\boldsymbol{f}(\boldsymbol{x})$ of k objectives is defined as:

$$\boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x})] \tag{1}$$

in which $\boldsymbol{x}$ is a vector of decision variables in the n-dimensional space $\mathbb{R}^n$; n and k are not necessarily the same. Each individual is assigned a vector $\boldsymbol{x}$ and therefore the corresponding vector $\boldsymbol{f}$. The decision to select an individual is now partially dependent on all objectives (as in (1)). This usually results in several trade-off individuals. This makes it quite suitable for using MOEAs, since they are population-based approaches. MOEAs offer a set of solutions; where which solution is to be selected at the end of the day is dependent on the users' decision.

An individual x_1 is said to dominate x_2 if x_1 is better than x_2 when measured on all objectives. If x_1 does not dominate x_2 and x_2 also does not dominate x_1, they are said to be non-dominated. If we use $\preceq$ between x_1 and x_2 as $x_1 \preceq x_2$ to denote that x_1 dominates x_2 and $\lhd$ between two scalars a and b as $a \lhd b$ to denote that a is better than b (similarly, $a \rhd b$ to denote that a is worse than b, and $a \ntriangleright b$ to denote that a is not worse than b), then the (weak) domination concept is formally defined as follows.

Definition 1 (Dominance). $x_1 \preceq x_2$ *if the following conditions are held*:

1) $f_j(x_1) \ntriangleright f_j(x_2)$, $\forall j \in [1, 2, \ldots, k]$,

2) $\exists j \in [1, 2, \ldots, k]$ *in which* $f_j(x_1) \lhd f_j(x_2)$.

In general, if an individual in a population is not dominated by any other individual in the population, it is called a non-dominated individual. All non-dominated individuals in a population form the non-dominated set or the Pareto front (as formally described in Definition 2). Note that these definitions are extracted from the ones in [1].

Definition 2 (Non-Dominated Set). *A set S is said to be the non-dominated set of a population P if the following conditions are held*:

1) $S \subseteq P$,

2) $\forall s \in S$, $\nexists x \in P | x \preceq s$.

When the set P represents the entire search space, the set of non-dominated solutions S is called the *global Pareto optimal set*. If P represents a sub-space, S will be called the *local Pareto optimal set*. There is only one global Pareto optimal set, but there could be multiple local ones. However, in general, we simply refer to the global Pareto optimal set as the Pareto optimal set (or *Pareto optimal front* (POF) for the plot of this set in the objective space).

The majority of existing MOEAs employ the concept of dominance in their courses of action; therefore, we focus on the class of dominance-based MOEAs only. The key features of a dominance-based MOEA are as follows: at each iteration, in order to select potentially better solutions for the next generation, the objective values are calculated for every solution and these are then used to determine the dominance relationships within the population. In this way, the population should converge to the POF. Generally, the MOEA has to overcome two major problems[1]. The first problem is how to get close to the Pareto optimal front. This is not an easy task, because converging to the POF is a stochastic process. The second is how to keep diversity among the solutions in the obtained set. These two problems become common criteria for most current algorithmic performance comparisons.

To date, many MOEAs have been developed. Generally speaking, there are many ways to classify MOEAs. However, we follow the one used in [4] where they are classified into two broad categories: non-elitism and elitism. Within the elitism approach, MOEAs employ an external set (the archive) to store the non-dominated solutions after each generation. This set will then be a part of the next generation. With this method, the best individuals in each generation are always preserved, and this approach helps the algorithms get closer to the POF. Algorithms, such as SPEA2[6], PDE[7], and NSGA-II[8], are examples of this category. In contrast, the non elitism approach has no concept of elitism when it performs selection of individuals for the next generation from the current population[11]. Examples of this category include VEGA[5] MOGA[12], NPGA[13], and NSGA[14] (note that VEGA does not use the concept of dominance).

2.2 Non-Dominated Sorting Genetic Algorithms Version 2 — NSGA-II

NSGA-II is an elitism algorithm[1]. The main feature of NSGA-II lies in its elitism-preservation operation. Note that NSGA-II does not use an explicit archive; a population is used to store both elitist and non-elitist solutions for the next generation. However, for consistency, we still consider it as an archive. Firstly, its archive size is set equal to the initial population size. The current archive is then determined based on the combination of the current population and the previous archive. To do this, NSGA-II uses dominance ranking to classify the population into a number of layers, such that the first layer is the best layer in the population. The archive is created based on the order of the ranking layers: the best rank being selected first. If the number of individuals in the archive is smaller than the population size, the next layer will be taken into account and so forth. If adding a layer makes the number of individuals in the archive exceed the initial population size, a truncation operator is applied to that layer based on the crowding distance measure.

The crowding distance D of a solution x is calculated as follows: the population is sorted according to each objective to find adjacent solutions to x; where boundary solutions are assigned infinite values and the average of the differences between the adjacent solutions in each objective is calculated. The truncation operator then removes the individual with the smallest crowding distance, given by:

$$D(x) = \sum_{m=1}^{M} \frac{F_m^{I_x^m+1} - F_m^{I_x^m-1}}{F_m^{\max} - F_m^{\min}} \tag{2}$$

in which $\boldsymbol{F}$ is the vector of objective values, and I_x^m returns the sorted index of the solution x, according to the m-th objective.

An offspring population of the same size as the initial population is then created from the archive by using crowded tournament selection, crossover, and mutation operators. Crowded tournament selection is a traditional tournament selection method, but when two solutions have the same rank, it uses the crowding distance measure to break the tie.

2.3 Strength Pareto Evolutionary Algorithm 2 — SPEA2

SPEA2 is actually the development of an elitism EMO algorithm called "The Strength Pareto Evolution Algorithm"[11]. This subsection just concentrates on the main points of SPEA2. The initial population, representation and evolutionary operators are standard; with uniform distribution, binary representation, binary tournament selection, single-point crossover, and bit-flip mutation. However, the distinctive feature of SPEA2 lies in being used its elitism-preservation operation.

An external set is created for storing primarily non-dominated solutions. It is then combined with the current population to form the next archive that is then used to create offspring for the next generation. The size of the archive is fixed. It can be set to be equal to the population size. Therefore, there exist two special situations when filling solutions in the archive. If the number of non-dominated solutions is smaller than the archive size, other dominated solutions taken from the remaining part of the population are filled in. This selection is carried out according to a fitness value, specifically defined for SPEA. That is the individual fitness value defined for a solution, x, is defined as the total of the SPEA-defined strengths of solutions which dominate x, plus a density value.

The second situation happens when the number of non-dominated solutions is over the archive size. In this case, a truncation operator is applied. For that operator, the solution which has the smallest distance to the other solutions will be removed from the set. If solutions have the same minimum distance, the second nearest distance will be considered, and so forth. This is called the *k-th nearest distance rule*.

2.4 Guided Approaches in the Objective Space

The original motivation for using guided approaches is that in practical problems we usually need a limited number of sample points of the POF rather than the whole POF[15]. In this sense, the objective space is limited to the POF part of interest only. Based on this idea, some work has been carried out to guide the search to different parts of the POF[10,16–18]. A further advantage of guided approaches is the ability to distribute the computation effort to a number of processors (instead of only one) where each processor is designated to a sub-population and hence tracks a part of the POF. In general, we categorize the different approaches for guiding the search to different parts of the POF into two categories, which we call soft and hard guidance.

• Soft guidance: the solutions in other parts of the POF are still considered, but with lower rank (or priority). Further, it allows some overlapping between the parts. The guided dominance method[16] can be seen as an example for this approach.

• Hard guidance: this approach is a straightforward one where the boundary of each part (sub-region of the POF) is used as a constraint. Solutions that are outside a sub-region are marked as having a constraint violation. An example of this approach is cone separation[17].

Deb *et al.*[16] proposed a technique using guided dominance to divide the POF into a number of parts, where each part was tracked by a different subpopulation. With guided dominance, the dominance relation is determined from the transformed functions Ω of the

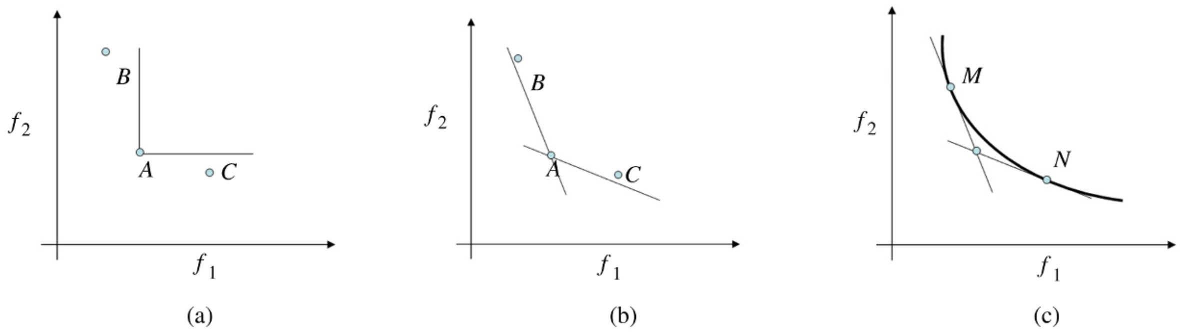


Fig.1. Space transformation using guided dominance for two-objective problems. Dominated region of a solution A using conventional dominance concept (a), guided dominance (b), and the POF part of interest (c).

original objective functions f, in which the points in the POF area of interest dominate all others in the remaining areas of the POF. Fig.1 is for the demonstration of how the dominance concept is changed via showing changes of the area that the point A dominates. The left graph shows the area being dominated using a conventional dominance relation. In this case, solutions B and C are not dominated by A; meanwhile the middle one shows the dominated areas using the transformed functions Ω. In more detail, in (3), we give an example of how to calculate Ω from f in a 2D space.

$$\begin{pmatrix} \Omega_1 \\ \Omega_2 \end{pmatrix} = \begin{pmatrix} 1 & a_{12} \\ a_{21} & 1 \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} \quad (3)$$

where a_{12} and a_{21} are predefined weights. Now (for certain a_{12} and a_{21}) B and C become dominated by A. Under this transformation scheme, the equivalent part of the POF is defined as in Fig.1(c) — the curve between M and N. The square matrix in (3) is called the weighted matrix. In order to divide the POF, an equivalent number of weighted matrices is defined to transform the original objective functions. In other words, the dominance relation in each population is defined by a separate weighted matrix. With the partition of the POF in the objective space, the search process is easily guided. The use of the weighted sum method to transform the objective functions might be less effective, in the cases of problems with non-convex POFs.

Likewise, Branke *et al.*[17] also introduced an approach to dividing the POF into a number of parts, where each part was tracked by an island in an island-based evolutionary algorithm. This approach divided the POF by a cone separation in which the targeted POF area was within a 90° angle starting from a reference point. An example is given in Fig.2 where the POF is divided into three parts by defining two straight lines from the reference point where these lines partition the angle at the reference point into three equal sub-angles.

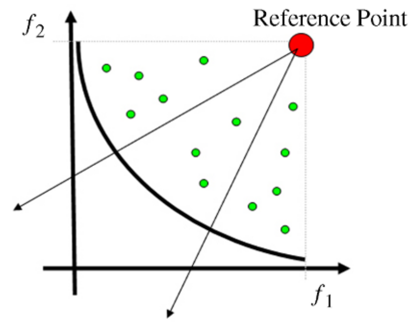


Fig.2. Demonstration of cone separation with three parts of the POF.

Further, all the borders of the divided regions are then used as constraints for the optimization problem. This is the most straightforward method of dividing the POF regardless of the convexity of the POF. Note that the problems with a discontinuous POF need careful consideration in advance (which is not always possible in practice) in order to avoid unnecessary separation.

2.5 Framework of Local Models

In this subsection, we summarize the method of local models that was proposed by Bui *et al.*[9] Technically, with the local models, the decision search space S is divided into a number of non-overlapping spheres where each sphere s_i is defined by a centroid and radius pair: $S = [s_0, s_1, \ldots s_n]$ and $s_i = (c_i, r_i)$ (see also Definition 3). Initially, all r_i are set to be the same default value r.

Definition 3. *A sphere is a local area in the decision sphere and is associated with a pair of centroid and radius.*

As stated above, the initial spheres are non-overlapping. Therefore, the distance between two spheres will be used to guarantee that they are non-overlapping. Let D_A^B be the Euclidean distance between two centroids of arbitrary spheres A and B. They can be guaranteed to not overlay when:

$$D_A^B \geqslant 2r. \tag{4}$$

Inside a sphere, points are generated uniformly and are kept further apart from each other with a predefined distance β away from each other, in which $\beta \approx r/M$ and M is the number of solutions in each sphere.

Let D_A^B be the Euclidean distance between two centroids of arbitrary spheres A and B, d_i^j be the Euclidean distance between two arbitrary points i and j inside a sphere X, and c_x be the centroid of sphere X. The required condition is that:

$$\begin{aligned} &d_i^{cx} \leqslant r, \\ &d_j^{cx} \leqslant r, \\ &d_i^j \geqslant \beta. \end{aligned} \tag{5}$$

To initialize a sphere s_i, we use the spherical coordinate system, since it easily guarantees to generate points inside the sphere and therefore speeds up the initialization process. Let $x = (x_1, x_2, \ldots, x_n)$ be a point in the Cartesian coordinate system. We can calculate x from the parameters of an equivalent spherical coordinate system including a radial coordinate r, and $n-1$ angular coordinates $\alpha_1, \alpha_2, \ldots, \alpha_{n-1}$ as follows:

$$\begin{aligned} x_1 &= c_1^i + r\cos(\alpha_1), \\ x_2 &= c_2^i + r\sin(\alpha_1)\cos(\alpha_2), \\ x_3 &= c_3^i + r\sin(\alpha_1)\sin(\alpha_2)\cos(\alpha_3), \\ &\vdots \\ x_{n-1} &= c_{n-1}^i + r\sin(\alpha_1)\cdots\sin(\alpha_{n-2})\cos(\alpha_{n-1}), \\ x_n &= c_n^i + r\sin(\alpha_1)\cdots\sin(\alpha_{n-2})\sin(\alpha_{n-1}). \end{aligned} \tag{6}$$

Therefore, for a point x in the sphere, we generate randomly a set of $n-1$ angular values α and then apply (6) to calculate Cartesian coordinate values for x.

Later, each sphere runs its own EMO algorithm and over time, the spheres are moved and guided towards the global Pareto front. Generally, the local models are formally defined as in Definition 4.

Definition 4. *Local models are defined as a system of adaptive spheres in the decision space and are kept under the control of a moving operator.*

Also, the framework is formalized as follows.

Step 1. Define spheres:

- Number of spheres.
- Initial radius for spheres.
- Centroids of spheres complying with the rules in (4).
- Minimum distance β between two points.

Step 2. Initialize spheres: using the uniform distribution while following (6) and complying with the rules in (5).

Step 3. Run one evolutionary cycle of the MOEA on each sphere.

Step 4. Move spheres with a predefined strategy.

Step 5. If stop condition is not met goto Step 3, otherwise stop the process.

It is obvious from the framework that the strategy to control the movement of the spheres (Step 4) plays an important role. For it, the authors propose several techniques including both communication and non-communication ones. For the *non-communication strategy* (called RACING), the spheres are allowed to move with their own paths without communication among them. Hence, the centroid of a sphere is calculated based on all non-dominated solutions in that sphere. Formally, the centroid c_i of a sphere s_i is recalculated as follows:

$$c_i = \frac{\sum (X_i^j)_{j=1}^N}{N} \tag{7}$$

in which X_i is the set of obtained non-dominated individuals with the size of N.

Further, as soon as the centroid of a sphere s_i is recalculated, the dominated individuals are discarded and the radius of the sphere is also adjusted as in (8). The effect of (8) is that the maximal distance (called *new_r*) from the centroid to any non-dominated individual in that sphere is assigned to be the new radius if it is greater than the default radius r. All new generated individuals are restricted to be within the sphere. Also, over time, spheres can be overlapping.

$$r_i = \begin{cases} new\_r, & \text{if } new\_r > r, \\ r_i, & \text{otherwise}. \end{cases} \tag{8}$$

We can consider the case in Fig.3 as an example of sphere movement. The current sphere is as in the graph on the left side (Fig.3(a)). Next, ignoring the dominated individuals which will be discarded, suppose that all the points in (Fig.3(b)) (the middle graph) are non-dominated. The new centroid (the big white dot) and new radius will be calculated; hence forming a new sphere. All individuals outside the new sphere are discarded and the remaining solutions are used to generate offspring (the small black dots in (Fig.3(c)) (the graph on the right side)); then the entire new sphere is built and is ready for the next round of evolution.

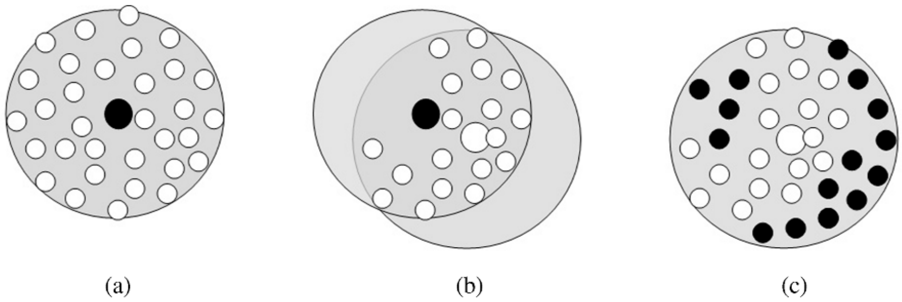


Fig.3. Changing of the centroid to determine the direction of improvement (Suppose the problem of two objectives with maximization of f1 (horizontal axis) and minimization of f2 (vertical axis)). (a) Current sphere where the big black dot is its centroid. (b) Transition state in which the big white dot is the new centroid. (c) New sphere with newly generated offspring (small black dots).

Meanwhile, for the *communication strategies*, the spheres are allowed to regularly communicate with each other during the optimization process and their movement is controlled by rules inspired from Particle Swam Optimization (PSO)[19]. Two versions of this technique were devised, called PSOV1 (that strictly follows the rules from PSO) and PSOV2 (a modified version from PSOV1, that is described in detail later in the section of methodology).

In general, the unique property of this framework is that it creates a system of local models based on localization of the decision space and allows these spheres to interact with each other following PSO-inspired rules. In [9], it is shown that the local models provide a quick convergence to the global POF. Further, they offer an effective approach to applying distributed computing to evolutionary multi-objective optimization.

3 Methodology

In our proposed framework, all the spheres are initialized following the conditions in (5). Centroids are then recalculated for all spheres after every round of evolution — a round is completed when all spheres finish one cycle of their own evolutionary process. The new and old centroids are used to determine the direction of improvement. We use PSOV2 (a strategy to guide spheres in the decision space), which in simple terms, applies a weak stochastic pressure to move the spheres towards the global optima. Details of how the direction of improvement is implemented as well as movement strategies are given in [9].

Note that the direction of improvement in the local models exploits both local and global information. However, in problems such as the ZDTs, the use of global information might cause the spheres to quickly move close to each other as they are approaching the POF. Thus, the search time might be wasted since the spheres search the same areas of the POF. Further, when spheres converge, the interaction among spheres might cause some unnecessary fluctuation. Therefore, it seems better to instead guide each sphere to occupy a different part of the POF, or at least reduce as much as possible the overlapping of the POF's parts that are discovered by the spheres. For this, we use guided dominance to specialize the spheres in the objective space.

We call this method Interleaving Guidance in Evolutionary Multi-objective Optimization. With this, our conjecture is that the local models with movement strategies in the decision space will speed up the convergence of the system, while the guided dominance will help to refine the obtained non-dominated set.

3.1 Strategy of PSOV2

For this strategy, the spheres are allowed to communicate after every some predefined number of generations. Before communication, the spheres move as in RACING. However, when communication happens, they each contribute non-dominated solutions to build a global archive. Here, we focus in the case where communication happens. The movement of each sphere s_i depends on where its centroid c_i is located. In order to determine a new location for each c_i, we need to determine two components: the new local centroid c_i^l of sphere s_i and the global archive's centroid c_i^g. The movement is modified to determine the new centroid for sphere s_i, from the new local centroid and the global one via its velocity v_i (9).

$$\begin{aligned} v_i(t+1) &= rand \times (c_i^g(t+1) - c_i^l(t+1)), \\ c_i(t+1) &= c_i^l(t+1) + v_i(t+1). \end{aligned} \tag{9}$$

We use a visual example, given in Fig.4, to demonstrate this method. Suppose that we need to determine the new centroid D of a sphere. We assume that the new local centroid is B and the global centroid is C. For the RACING strategy, D and B are identical; and for the strategy of PSOV1, D will be determined from B and C based on the rules of Particle Swarm Optimization (or in other words, the velocity and centroid in (9) will be calculated strictly following the rules from PSO). However, for PSOV2, D is just a random location between B and C. Formally, let vectors $\boldsymbol{OC}$ represent the vector to the global centroid and $\boldsymbol{OB}$ represent that to a new found local centroid:

$$\begin{aligned} \boldsymbol{BD} &= \mathit{rand} \times (\boldsymbol{OC} - \boldsymbol{OB}), \\ \boldsymbol{OD} &= \boldsymbol{OB} + \boldsymbol{BD}, \end{aligned} \tag{10}$$

$\boldsymbol{OD}$ is the vector of a new adjusted centroid for sphere s_i, and $rand$ is a random number between 0 and 1. In this way, the technique is somewhat similar to the Pareto Differential Evolution algorithm[7].

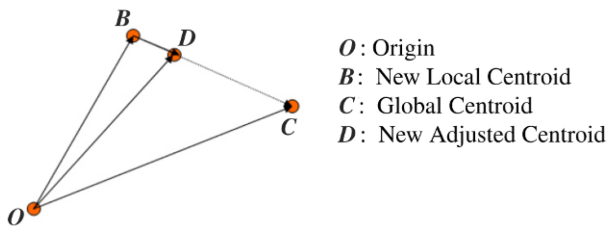


Fig.4. Changing of the centroid in PSOV2: using vectors to indicate the recalculation of the new centroid.

3.2 Interleaving Guidance

To implement the idea of interleaving guidance, we need to divide the POF into a number of parts. Each part is then used to guide a sphere. In this way, we use a number of *global* centroids instead of only one as in the original PSOV2 (see (9)); each global centroid is the center of the group of global non-dominated solutions that belong to an associated POF part. The number of parts, spheres, and global centroids are kept equal. As stated above, we have selected the guided dominance approach in [16] to guide the local models in the objective space. This is a "*soft*" division, in the sense that the parts are allowed some overlapping, but the overlapping is kept to a minimum.

In this approach, each sphere is associated with a POF part, the one that it contributes the most non-dominated solutions to, in comparison with the other parts. Each POF part is assigned only one sphere. When a sphere needs to be guided by global information, the sphere's centroid is determined from both the new local centroid, which is calculated based on all individuals of the sphere that belong to its associated POF part and the associated global centroid. In general, this task is done at Step 4 of the general framework of the local models. The task can be further described in steps as follows.

Step 1. Define spheres:
- Number of spheres (also the number of POF parts).
- Initial radius for spheres.
- Centroids of spheres complying with the rules in (4).
- Minimum distance β between two points.
- Specify the weight matrices for guided-dominance.

Step 2. Initialize spheres: using the uniform distribution while following (6) and complying with the rules in (5).

Step 3. Run one evolutionary cycle of the MOEA on each sphere.

Step 4. Make a move for each sphere
If (time for communication) {
Step 4.1. Select a sphere,
Step 4.2. Search for an available POF part, the one it contributes the most non-dominated solutions to, and assign the found part to the sphere,
Step 4.3. Calculate its new centroid based on its local new and the associated POF part's centroid,
Step 4.4. if not all spheres are assigned, goto step 1,
Step 4.5. Calculate new radix for spheres
} else {
Step 4.1. Calculate new centroid as in RACING,
Step 4.2. Calculate new radix for spheres
}.

Step 5. if stop condition is not met goto Step 3, otherwise stop the process.

4 Experimental Studies

In order to validate the proposed method, we carried out a comparative study on a set of test problems. Note that the performance of the local models with different controlling strategies (RACING, PSOV1 or PSOV2) were thoroughly analyzed in [9]. In this paper, we focus on the performance of

- local models with guidance in the decision space,
- guided dominance in the objective space,
- dual guidance in both spaces.

We selected NSGA-II as the algorithm to run in each sphere of our models. Experiments were performed and compared among for the following approaches.

• PSOV2: this approach is included in order to validate the new proposed technique of dual guidance. It has the same name as its strategy.

• GUIDED: the approach using only the strategy of guided dominance to guide the search in the objective space.

Table 1. Lists of Test Problems Used for Experiments in This Paper

Problems	Dim	Range	POF Features
BINH	2	$x_i \in [-5, 10]$ $f_1(x) = x_1^2 + x_2^2$ $f_2(x) = (x_1 - 5)^2 + (x_2 - 5)^2$	Convex Uniform Uni-modal Connected
POL	2	$x_i \in [-\pi, \pi]$ $f_1(x) = [1 + (A1 - B1)^2 + (A2 - B2)^2]$ $f_2(x) = [(x_1 + 3)^2 + (x_2 + 1)^2]$ $A1 = 0.5 \sin 1.0 - 2 \cos 1.0 + \sin 2.0 - 1.5 \sin 2.0$ $A2 = 1.5 \sin 1.0 - \cos 1.0 + 2 \sin 2.0 - 0.5 \sin 2.0$ $B1 = 0.5 \sin x_1 - 2 \cos x_1 + \sin x_2 - 1.5 \sin x_2$ $B2 = 1.5 \sin x_1 - \cos x_1 + 2 \sin x_2 - 0.5 \sin x_2$	Nonconvex Disconnected Uniform
LAU	2	$x_i \in [-50, 50]$ $f_1(x) = x_1^2 + x_2^2$ $f_2(x) = (x_1 + 2)^2 + x_2^2$	Convex Disconnected Uni-modal Uniform
LIS	2	$x_i \in [-5, 10]$ $f_1(x) = \sqrt[8]{x_1^2 + x_2^2}$ $f_2(x) = \sqrt[4]{(x_1 - 0.5)^2 + (x_2 - 0.5)^2}$	Nonconvex Uniform Uni-modal Disconnected
MUR	2	$x_1 \in [1, 4]$ and $x_2 \in [1, 2]$ $f_1(x) = 2\sqrt{x_1}$ $f_2(x) = x_1(1 - x_2) + 5$	Nonconvex Uni-modal Connected Uniform
REN1	2	$x_i \in [-3, 3]$ $f_1(x) = \frac{1}{x_1^2 + x_2^2 + 1}$ $f_2(x) = \frac{1}{x_1^2 + 3x_2^2 + 1}$	Convex Uniform Uni-modal Connected
REN2	2	$x_i \in [-3, 3]$ $f_1(x) = x_1 + x_2 + 1$ $f_2(x) = x_1^2 + 2x_2 - 1$	Convex Uniform Uni-modal Connected
KUR	3	$x_i \in [-5, 5]$ $f_1(x) = \sum_{i=1}^{n-1}(-10 \exp(-0.2\sqrt{x_i^2 + x_{i+1}^2}))$ $f_2(x) = \sum_{i=1}^{n}(\lvert x_i\rvert^{0.8} + 5 \sin(x_i)^3)$	Nonconvex Disconnected Uniform
FON	10	$x_i \in [-4, 4]$ $f_1(x) = 1 - \exp(-\sum_{i=1}^{n}(x_i - \frac{1}{\sqrt{n}})^2)$ $f_2(x) = 1 - \exp(-\sum_{i=1}^{n}(x_i + \frac{1}{\sqrt{n}})^2)$	Nonconvex Connected Uniform Uni-modal
QUA	16	$x_i \in [-5.12, 5.12]$ $f_1(x) = \sqrt{\frac{A_1}{n}}$ $f_2(x) = \sqrt{\frac{A_2}{n}}$	Non-convex Connected Uniform Multi-modal
ZDT1	30	$x_i \in [0, 1]$ $f_1(x) = x_1,\ f2(x) = g \times h$ $g(x) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$ $h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}}$	Convex Connected Uniform Uni-modal
ZDT2	30	$x_i \in [0, 1]$ $f_1(x) = x_1,\ f2 = g \times h$ $g(x) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$ $h(f_1, g) = 1 - (\frac{f_1}{g})^2$	Nonconvex Connected Uniform Uni-modal

To be continued

Continue from the previous page

Problems	Dim	Range	POF Features
ZDT3	30	$x_i \in [0,1]$	Convex
		$f_1(x) = x_1,\ f2 = g \times h$	Disconnected
		$g(x) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$	Uniform
		$h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}} - \frac{f_1}{g}\sin(10\pi f1)$	Uni-modal
ZDT4	10	$x_1 \in [0,1]$	Convex
		$x_i \in [-5,5],\ i = 2..10$	Connected
		$f_1(x) = x_1,\ f2 = g \times h$	Uniform
		$g(x) = 1 + 10(n-1) + \sum_{i=2}^{n}(x_i^2 - 10\cos(4\pi x_i))$	Multi-modal
		$h(f_1, g) = 1 - (\frac{f_1}{g})^2$	
ZDT6	10	$x_i \in [0,1]$	Nonconvex
		$f_1(x) = 1 - \exp(-4x_1)\sin^6(6\pi x_1),\ f2 = g \times h$	Connected
		$g(x) = 1 + 9[\frac{\sum_{i=2}^{10} x_i}{9}]^{0.25}$	Non-uniform
		$h(f_1, g) = 1 - (\frac{f_1}{g})^2$	Uni-modal

• IGMOEA: the approach of using the strategy of cone separation to guide the local models in the objective space. IGMORA stands for Interleaving Guided Multi-Objective Evolutionary Algorithm.

• NSGA-II: it is used as a baseline for comparison. An overview of NSGA-II was given in the previous section.

• SPEA2: another representative MOEA. An overview of SPEA2 was also given in the previous section.

4.1 Test Problems

A summary of the collection of 15 test problems that are used in this paper is given in Table 1. This is not an exhaustive list, but the problems represent different difficulties. We use these problems to observe the behavior of the algorithms. This list is divided into two sets.

• The first set includes 7 low-dimension problems (from BINH to REN2) that have 2D decision search space and the property of uni-modality.

• The second set (from KUR to ZDT6) contains more difficult problems with larger search space. They are either uni-modal or multi-modal problems.

Note that although guided-dominance approaches have been identified as working well on problems with convex POFs[16], for more information, we have still tested them on problems with non-convex POFs.

4.2 System Settings

For all experiments, the total population size was set as 200. In all tests of the proposed framework, three spheres were used (although the number of spheres was also tested from one to five with different controlling techniques to test the sensitivity of the framework to the number of local models). The global archive size was set to half the total population size. If the global archive size was too large, the centroid may have been affected by many inferior solutions and this would therefore affect the convergence speed. On the other hand, if it was too small, the archive would be too specialized and hence might lose some of the information contributed by some spheres. This alters the effect of the global information.

Further, the update frequency was kept quite small, in these experiments, it was set as 5 generations. All cases were tested in 30 separate runs for 30 different random seeds. The results have been analyzed within these 30 runs for each model and on each problem.

For GUIDED, we have also used 3 sub-populations in all experiments and have followed the settings introduced by its authors. In all experiments, for NSGA-II and SPEA2, real-valued versions were implemented using SBX crossover and polynomial mutation (see [1]). The crossover rate was 0.9 and mutation rate was 0.1. The distribution indexes for crossover and mutation operators were $\eta_m = 20$ and $\eta_c = 15$ as recommended by its authors[20]. Further, all models ran with the same number of evaluations in order to make a fair comparison.

4.3 Performance Measurement Methods

Performance metrics are usually used to compare algorithms in order to form an understanding of which one is better and in what aspects. However, it is hard to define a concise definition of algorithmic performance. In general, when doing comparisons, a number of criteria are employed[11]. We will look at two of these criteria. The first measure is the generation distance, *GD*, which is the average distance from the set of solutions found by evolution to the POF[21].

$$GD = \frac{\sqrt{\sum_{i=1}^{N} d_i^2}}{N} \tag{11}$$

where d_i is the Euclidean distance (in objective space) from solution i to the nearest solution in the POF. If there is a large fluctuation in the distance values, it is also necessary to calculate the variance of the metric. Finally, the objective values should be normalized before calculating the distance.

As recommended in [21], it is considered better to use the hyper-volume ratio, *HR*, which is measured by the ratio between the hypervolumes of hyperareas covered by the obtained POF and the true POF, called H_1 and H_2 respectively. *HR* is calculated as in (12). For this metric, the greater the value of *HR*, the better convergence the algorithm has.

$$HR = \frac{H1}{H2}. \tag{12}$$

There are some discussions on how to determine the reference point for the calculation of the hyper-volume, for example, it can be the origin[21]. However, generally it is dependent on the area of the objective space that is visited by all compared algorithms. In this paper, as suggested elsewhere[1], the reference point is the one associated with all the worst values of objectives found by all the algorithms under investigation.

4.4 Results and Discussions

Convergence is one of the most important characteristics of an optimization technique since its main use is to assess the performance of the algorithm. However, the way of looking at convergence of single objective and multi-objective optimizations are quite different[22]. If some measurements of the objective function, with regard to the number of generations, are experimentally considered as an indication for convergence in single objective optimization, it is not a suitable method for multi-objective optimizations since they do not involve the mutual assessment on all objective functions.

Further, the consideration of convergence is not only on how close the obtained POF, after a period of time, is in comparison to the true Pareto optimal front, but also the rate of convergence which is convergence over time. We consider both issues in this subsection. For the closeness of the obtained POF, we use the generation distance GD (as well as the hyper–volume ratio HR). While GD is an indication for the closeness of an obtained set of solutions to the true POF, HR is for both closeness and diversity of the obtained set.

Table 2. Predefined Level of the Hyper-Volume Ratio That Algorithms Need to Reach Within a Time Limit (They are kept as high as possible, but while still being the one that the majority of approaches could reach.)

Problems	Predefined Level
BINH	0.999
LAU	0.999
LIS	0.999
MUR	0.990
POL	0.995
REN1	0.995
REN2	0.999
KUR	0.940
FON	0.800
QUA	0.900
ZDT1	0.999
ZDT2	0.950
ZDT3	0.999
ZDT4	0.999
ZDT6	0.999

Firstly, the number of evaluations that each approach spends to reach a certain high level of hypervolume ratio (Table 2) was recorded. These were derived from the highest level of *HR* that almost all algorithms could achieve within an evaluation limit. This measurement is to provide a quantitative indication of how fast the algorithms are in converging to the true POF (by using the same high level of *HR*). These values are reported in Tables 3, 4, and 5. *t-tests* with 0.05 level of significance were used to test the difference between results of the compared approaches.

We start with the performance of the system using local models with guidance in the decision space alone — PSOV2. From Table 3, we can see that PSOV2 was obviously quicker than (or equal to) NSGA-II in the majority of problems. It was slower than NSGA-II — spending more time exploring — in only two problems (MUR and REN1) and both of these problems belong to the first set of simple problems (from BINH to REN2) where the dimension of the search space is 2D and the problems are uni-modal.

Further, the set of more difficult problems (KUR to ZDT6) showed a clearer picture. PSOV2 was no worse than NSGA-II in any problems; particularly, it was quicker in 5 problems and similar to NSGA-II in 3. The use of localization as well as PSO-oriented rules for controlling spheres gave the system much faster speed of convergence to the POF in comparison with NSGA-II.

Table 3. The Number of Evaluations SPEA2, NSGA-II and PSOV2 Needed to Reach the Predefined Levels of Hyper-Volume Ratio (Mean and Standard Deviation from 30 Runs)

	SPEA2	NSGA-II	PSOV2
BINH	**2067 ± 644**	1833 ± 320	**1863 ± 382**
LAU	**773 ± 282†**	1047 ± 286	**952 ± 172**
LIS	8213 ± 3158†	6300 ± 2678	**4916 ± 1229†**
MUR	1540 ± 150	1800 ± 197	14397 ± 31344†
POL	**700 ± 146**	693 ± 208	**932 ± 803**
REN1	**900 ± 155†**	993 ± 98	1163 ± 331†
REN2	**2547 ± 396**	2747 ± 484	**1843± 1296†**
KUR	**1460 ± 211**	1473 ± 249	**1428 ± 273**
FON	**2387 ± 210†**	6073 ± 326	**4726 ± 445†**
QUA	92533 ± 26032†	62107 ± 26139	**67517 ± 16493**
ZDT1	**19753 ± 746†**	21080 ± 527	**18108 ± 1360†**
ZDT2	8947 ± 1415†	5913 ± 422	**4678 ± 519†**
ZDT3	24600 ± 842†	23407 ± 1110	**21434 ± 1718†**
ZDT4	**18120 ± 1769**	18633 ± 1889	**19128 ± 2278**
ZDT6	**37973± 1346†**	38807 ± 1099	**4468 ± 314†**

Note: Symbol † indicates that the difference between PSOV2 and NSGA-II is significant (using a t-test with 0.05 level of significance). The texts in bold style show that the performance of SPEA2 or PSOV2 is better than or equal to that of NSGA-II.

A further comparison can be made between SPEA2 and NSGA-II where their performance was quite similar. SPEA2 was quicker than NSGA-II in 5 problems, and slower in 4 problems (divided equally among the two sets of test problems), and equal in the remaining ones. Note that the predefined levels of HR was set very high (> 0.99 for almost all problems). This ensured that the algorithms converged to the POF.

Regarding the performance of GUIDED on the test problems, we can also compare it with NSGA-II in Table 4. In our experiments, it seems that GUIDED was quite slow in converging to the POF. For example, although it had a quicker convergence in some difficult problems, such as QUA, ZDT4 or ZDT6, it was slower in many others, and sometimes did not even reach the predefined levels (such as REN2, KUR, or ZDT3). It is interesting to note that *GUIDED was inferior in all problems with a non-convex POF* (except QUA). This is somehow expected because of the use of a weighted-sum transformation which assumes convexity of the POF (as stated in [16]).

In general, the above results imply that guidance in the decision space (using the local models) gives the system a quicker convergence rate in comparison with guidance in the objective space (using guided dominance).

Now we move to discussion on the performance of IGMOEA. As shown in the previous section, IGMOEA is an extended version of PSOV2 using dual guidance

Table 4. The Number of Evaluations NSGA-II and GUIDED Needed to Reach the Certain Levels of Hypervolume Ratio (Mean and Standard Deviation from 30 Runs)

	NSGA-II	GUIDED
BINH	1833 ± 320	**2033 ± 916**
LAU	1047 ± 286	**647 ± 266†**
LIS	6300 ± 2678	9507 ± 8417†
MUR	1800 ± 197	2367 ± 328†
POL	693 ± 208	**720 ± 244**
REN1	993 ± 98	3253 ± 1146†
REN2	2747 ± 484	NA ± NA †
KUR	1473 ± 249	NA ± NA †
FON	6073 ± 326	51273 ± 51621†
QUA	62107 ± 26139	**33520 ± 32109†**
ZDT1	21080 ± 527	24187 ± 1012†
ZDT2	5913 ± 422	9387 ± 1248†
ZDT3	23407 ± 1110	NA ± NA†
ZDT4	18633 ± 1889	**17067 ± 429†**
ZDT6	38807 ± 1099	**39367 ± 1241**

Note: Symbol † indicates that the difference between NSGA-II and GUIDED is significant (using a t-test with 0.05 level of significance). The texts in bold style show that the performance of GUIDED is better than or equal to that of NSGA-II.

Table 5. The Number of Evaluations PSOV2 and IGMOEA Needed to Reach the Pre-Defined Levels of Hypervolume Ratio (Mean and Standard Deviation from 30 Runs)

	PSOV2	IGMOEA
BINH	1863 ± 382	2853 ± 1538†
LAU	952 ± 172	2387 ± 2200†
LIS	4916 ± 1229	12727 ± 20485†
MUR	14397 ± 31344	**6680 ± 17338**
POL	932 ± 803	1767 ± 1329†
REN1	1163 ± 331	**1253 ± 364**
REN2	1843 ± 1296	4733 ± 2425†
KUR	1428 ± 273	1853 ± 528†
FON	4726 ± 445	7767 ± 1770†
QUA	67517 ±16493	**34580 ± 7530†**
ZDT1	18108 ± 1360	**18268 ± 27671**
ZDT2	4678 ± 519	13480 ± 8895†
ZDT3	21434 ± 1718	**11560 ± 1192†**
ZDT4	21304 ± 495	**16613 ± 523†**
ZDT6	4468 ± 314	5300 ± 853†

Note: Symbol † indicates that the difference between PSOV2 and IGMOEA is significant (using a t-test with 0.05 level of significance). The texts in bold style show that the performance of IGMOEA is better than or equal to that of PSOV2.

in both spaces. Therefore, we recorded its convergence rate and compared it with PSOV2. All results have been reported in Table 5. With IGMOEA, we expect it to inherit the quick convergence rate from PSOV2, but with a better capability of refining the obtained non-dominated set (we will measure the quality of the set to show how it refined the set). It is very interesting that for problems where IGMOEA was quicker than PSOV2 (MUR, REN1, ZDT1, ZDT3, and ZDT4), they are all problems with a convex POF.

Further, in the other non-convex ones, IGMOEA was better on problem QUA, which has a non-convex POF and multiple local POFs. It might be the case that the division of the population allowed IGMOEA enough diversity to overcome the difficulty of multimodality.

From the above results, we can see that the approaches converged within 20 000 evaluations for almost all problems. Therefore, in order to measure the quality of the obtained non-dominated set, we calculated GD and HR after a period of 20 000 evaluations for all approaches, and have reported the mean and standard derivation among 30 runs. All the results were reported in Tables 6 and 7.

We consider the first set of simple problems (from BINH to REN2) where the dimension of the search space is 2D and the problems are uni-modal. The performance of all approaches were quite similar with small values of GD (Table 6) and large values of HR (Table 7). For these problems, the use of dual guidance did not help much in terms of convergence and diversity, since with a population size of 200 individuals, all techniques easily and quickly approached the POF.

However, the set of more difficult problems (KUR to ZDT6) showed a different story. In Table 6, IGMOEA was the best in almost all problems (except KUR FON, and QUA). There was a significant improvement of PSOV2 with the use of dual guidance. We can see that ZDT1 and ZDT3 are examples where IGMOEA converged exactly to the POF (with zero value of GD and an HR value of one).

There was inefficiency in the use of dual guidance in some problems with non-convex POFs such as KUR FON, or QUA. We attribute this inferior performance to the weighted sum approach used in guided dominance. Note that, for QUA, all approaches obtained good values of GD, but not HR. This indicates that they all converged to the POF, but did not cover all parts of the POF. Furthermore, because of the multimodality in this problem, 20 000 evaluations are not enough to obtain a diverse set of non-dominated solutions. Table 5 shows that IGMOEA became the best after 35 000 evaluations.

In this paper, we have used a simple mechanism for tracking the convergence, by measuring GD and HR over time, since it is consistent with the performance measures used above. We can compare GD and HR of all approaches; as for example in Figs. 5 $\sim$ 10, where we have visualized the averaged GD and HR of 30 runs for all approaches over a period of 20 000 evaluations (100 generations) and for several problems.

As expected, for the first set of simple problems, all approaches converged quickly to the POF (see Fig.5 for an example). It is also possible to see a regular series of drops in the curve for GUIDED approaches. These drops reflect some sort of loss of diversity in the POF. This sometimes happens when the sphere's motion changes (with the update frequency), and then the sub-population has to adjust to that change. There-

Table 6. Generation Distance (*Mean and the Standard Deviation* from 30 Runs) Recorded After 20 000 Evaluations for all Methods

	NSGA-II	SPEA2	PSOV2	IGMOEA	GUIDED
BINH	0.007±0.000	**0.006±0.000†**	0.009±0.001†	0.009±0.001†	**0.007±0.000**
LAU	0.089±0.002	0.093±0.001†	**0.090±0.003**	0.093±0.008†	**0.073±0.003†**
LIS	0.002±0.000	**0.001±0.000†**	**0.002±0.000**	**0.001±0.000†**	0.006±0.002†
MUR	0.000±0.000	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	0.001±0.000†
POL	0.002±0.005	**0.002±0.003**	**0.003±0.007**	**0.003±0.005**	**0.001±0.003**
REN1	0.001±0.000	**0.001±0.000**	**0.001±0.000**	**0.001±0.000**	**0.001±0.000**
REN2	0.001±0.000	**0.001±0.000**	**0.001±0.000**	**0.001±0.000**	0.013±0.013†
KUR	0.001±0.000	**0.001±0.000**	**0.001±0.000**	0.002±0.000†	0.060±0.045†
FON	0.001±0.000	**0.000±0.000†**	**0.001±0.000**	**0.001±0.000**	**0.001±0.000**
QUA	0.001±0.000	**0.000±0.000**	**0.001±0.000**	0.002±0.000†	**0.001±0.001**
ZDT1	0.001±0.000	**0.001±0.000**	**0.001±0.000**	**0.000±0.000†**	**0.001±0.000**
ZDT2	0.001±0.000	**0.001±0.000**	**0.001±0.000**	**0.002±0.010**	0.002±0.001†
ZDT3	0.001±0.000	**0.001±0.000**	**0.001±0.000**	**0.000±0.000†**	0.002±0.001†
ZDT4	0.063±0.045	**0.048±0.040**	0.104±0.081†	**0.031±0.022†**	0.262±0.407†
ZDT6	0.022±0.004	**0.015±0.003†**	**0.002±0.003†**	**0.002±0.003†**	0.031±0.004†

Note: The texts in bold style are to show the better or equal performance to that of NSGA-II. Symbol † indicates that the difference between NSGA-II and the others is significant (using t-test with 0.05 level of significance).

Table 7. Hyper-Volume Ratio (*Mean and the Standard Deviation* from 30 Runs) Recorded After 20 000 Evaluations for All Methods

	NSGA-II	SPEA2	PSOV2	IGMOEA	GUIDED
BINH	1.000±0.000	**1.000±0.000**	0.999±0.000†	0.999±0.000†	**1.000±0.000**
LAU	1.000±0.000	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
LIS	1.000±0.016	**1.000±0.018**	**1.000±0.013**	**1.000±0.023**	**1.000±0.021**
MUR	0.999±0.000	**0.999±0.000**	0.993±0.011†	0.994±0.006†	0.995±0.001†
POL	1.000±0.000	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
REN1	0.998±0.000	0.997±0.000†	**0.998±0.000**	**0.998±0.000**	0.994±0.001†
REN2	1.000±0.000	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	0.888±0.024†
KUR	1.000±0.000	**1.000±0.000**	0.999±0.000†	0.998±0.001†	0.850±0.066†
FON	0.962±0.004	**0.990±0.001†**	0.945±0.004†	0.948±0.011†	0.708±0.234†
QUA	0.872±0.014	0.860±0.011†	0.856±0.013†	0.868±0.015†	**0.895±0.031†**
ZDT1	0.999±0.000	**0.999±0.000**	**0.999±0.000**	**1.000±0.001**	0.998±0.000†
ZDT2	0.998±0.000	**0.998±0.000**	**0.999±0.000†**	0.971±0.025†	0.993±0.005†
ZDT3	0.998±0.000	**0.998±0.000**	**0.999±0.000†**	**1.000±0.000†**	0.994±0.002†
ZDT4	1.000±0.001	**1.000±0.001**	0.999±0.001†	**1.000±0.001†**	**1.000±0.001**
ZDT6	0.973±0.003	**0.979±0.003†**	**1.000±0.000†**	**1.000±0.000†**	0.968±0.004†

Note: The texts in bold style are to show the better or equal performance to that of NSGA-II. Symbol † indicates that the difference between NSGA-II and the others is significant (using t-test with 0.05 level of significance).

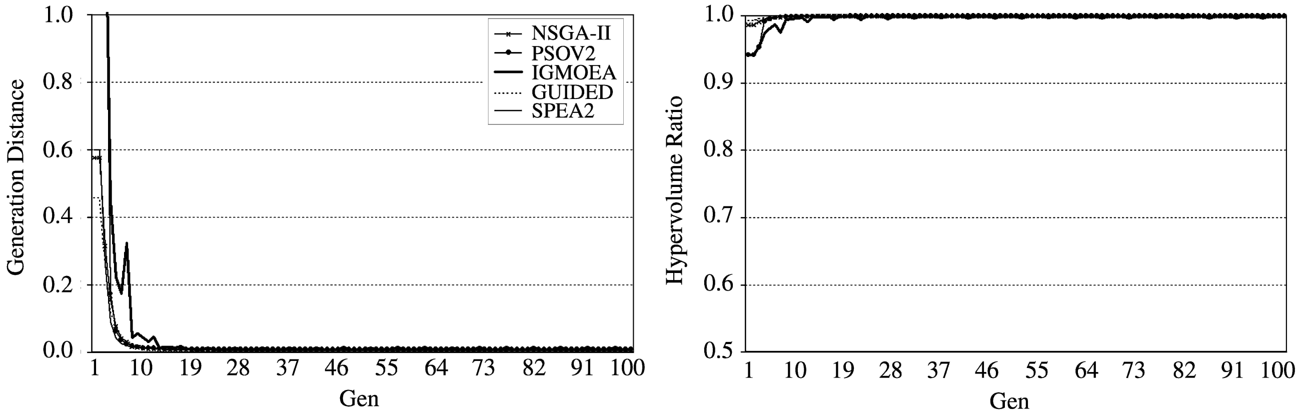


Fig.5. Generation distance (left) and hypervolume ratio of differing approaches over time (up to 20 000 evaluations or 100 generations) for BINH.

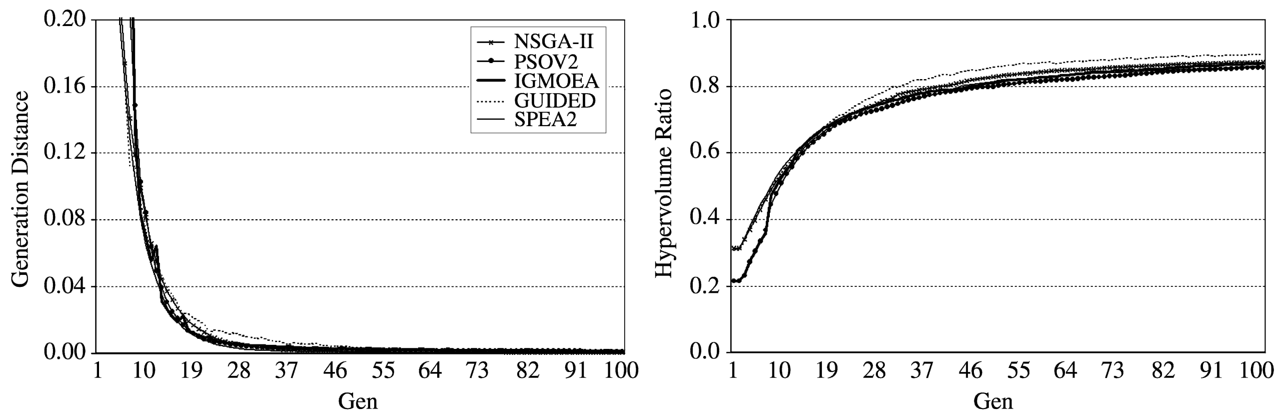


Fig.6. Generation distance (left) and hypervolume ratio of differing approaches over time (up to 20 000 evaluations or 100 generations) for QUA.

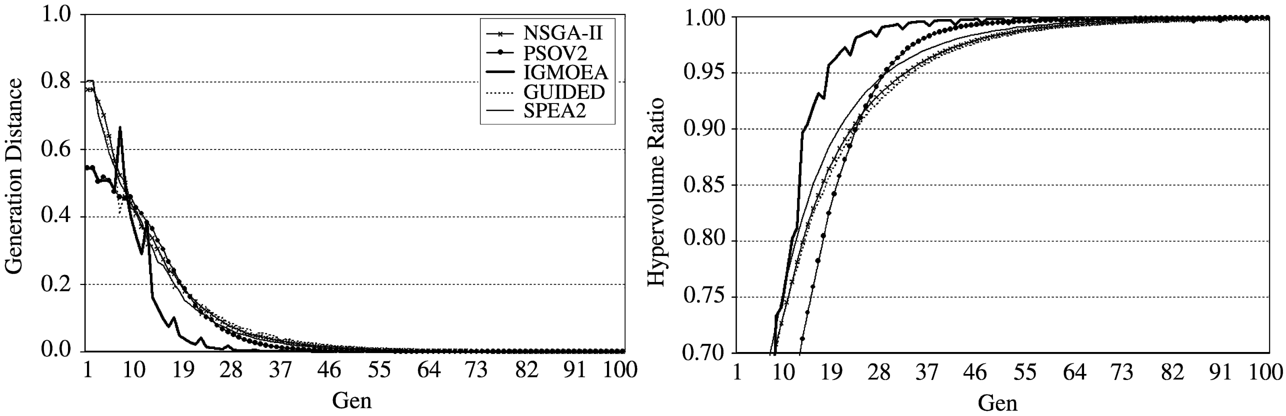


Fig.7. Generation distance (left) and hypervolume ratio of differing approaches over time (up to 20 000 evaluations or 100 generations) for ZDT1.

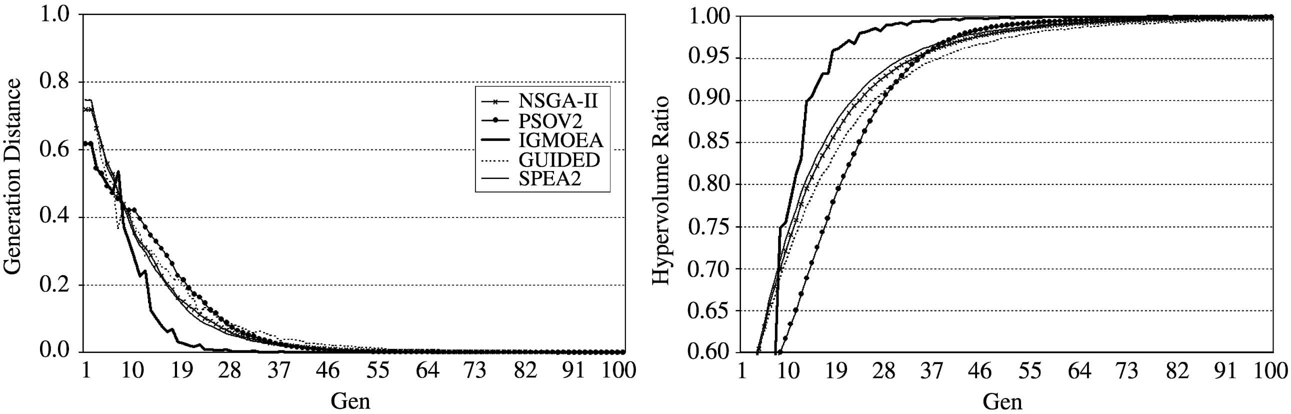


Fig.8. Generation distance (left) and hypervolume ratio of differing approaches over time (up to 20 000 evaluations or 100 generations) for ZDT3.

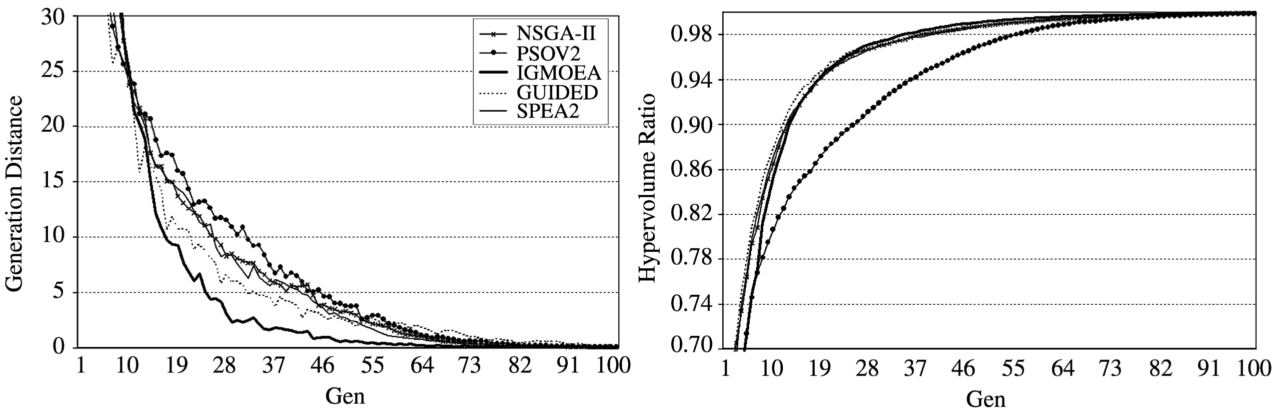


Fig.9. Generation distance (left) and hypervolume ratio of differing approaches over time (up to 20 000 evaluations or 100 generations) for ZDT4.

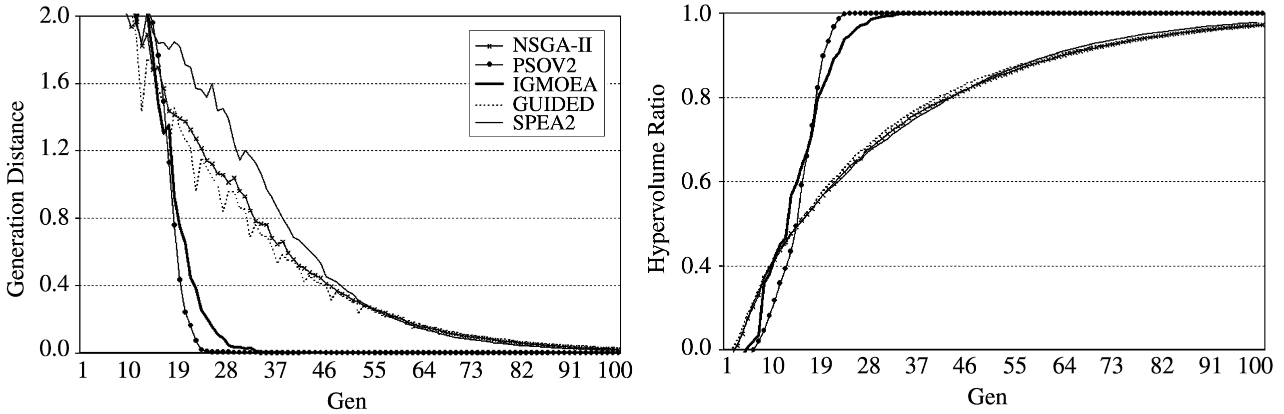


Fig.10. Generation distance (left) and hypervolume ratio of differing approaches over time (up to 20 000 evaluations or 100 generations) for ZDT6.

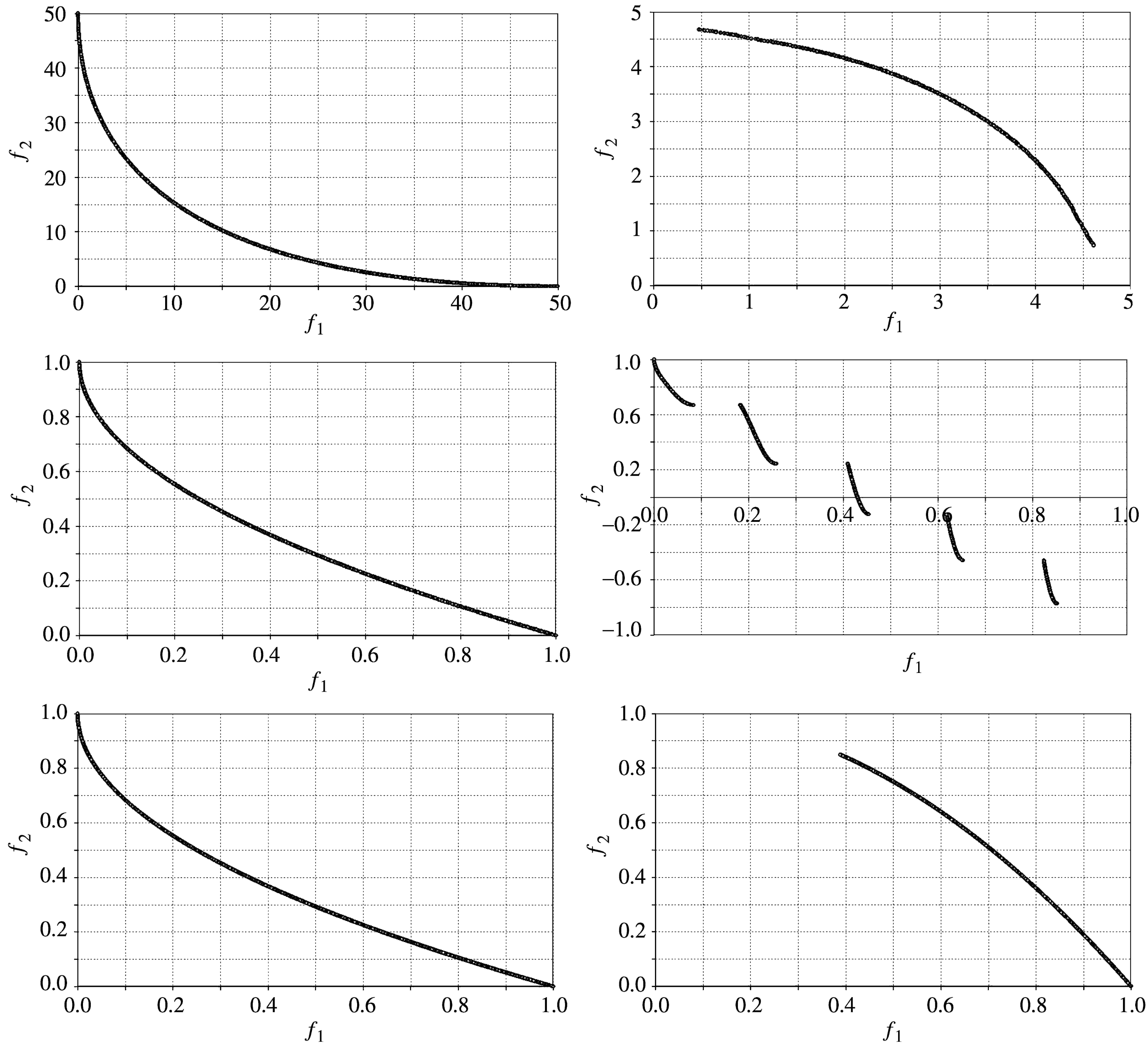


Fig.11. Non-dominated sets that IGMOEA obtained for BINH, QUA, ZDT1, ZDT3, ZDT4 and ZDT6.

fore, this is reflected in the drop-recovery cycles shown in the figure, which seems to be part of the process of first moving to a new better area in the search space, followed by an exploration phase of that area. This might be useful for IGMOEA in problems with a large, multi-modal search space.

The convergence of the GUIDED approaches is clearer in the second set of problems (some examples are given in Figs. 6 ∼ 10). In the first few generations, IGMOEA (also PSOV2) was quite slow. This is because localization starts with an exploration phase which consumes time. However, the adaptive strategies of IGMOEA and PSOV2 help the methods to adjust to move faster if the search space seems smooth enough. It then becomes clearer to see that IGMOEA converged very quickly to the optimal. In other words, PSOV2 was improved with the use of dual guidance.

All in all, the dual guidance technique shows a good ability to quickly approach the true POF. With the above test problems, it was able to obtain converged and diverse POFs (see Fig.11 as an example).

4.5 Analysis on the Division of the POF

In the previous section, we have shown the superior performance of IGMOEA over its counterparts (PSOV2 and GUIDED) in converging to the POF. Here, we make a further investigation on the effect of dual guidance under different schemes of POF division. We used guided dominance with the division of the POF into two, three, four and five parts. We kept the population size fixed (200 individuals) and the number of sub-populations equal to the number of POF parts. Again, we recorded the number of evaluations that IGMOEA needed to achieve the predefined levels of HR (see Table 8).

It is clear that in many problems, especially the difficult ones, the performance of IGMOEA was reduced when we increased the number of parts (or the number of sub-populations). Specially, it required more time to reach the predefined levels of HR. Again, IGMOEA was quicker than NSGA-II in almost all problems of the second set. It was slower only in two non-convex problems: FON and ZDT2.

Furthermore, all of the values of GD after a period of 20 000 evaluations were recorded and reported in Table 9. Once more, the performance of IGMOEA decreased when we increased the number of parts (or the number of sub-populations). In comparison with NSGA-II (the global model), IGMOEA-2 was worse in terms of GD in one problem (BINH, a most simple one), and IGMOEA-5 was worse in 5 problems (BINH, LAU, KUR, QUA, and ZDT4). Overall, IGMOEA had the best performance with two and three sub-populations.

Table 8. The Number of Evaluations NSGA-II and Versions of IGMOEA Needed to Reach the Certain Levels of Hypervolume Ratio (Mean and Standard Deviation from 30 Runs)

	NSGA-II	IGMOEA-2	IGMOEA-3	IGMOEA-4	IGMOEA-5
BINH	1833 ± 320	5320 ± 5925†	2853 ± 1538†	4529 ± 2757†	4902 ± 3418†
LAU	1047 ± 286	2667 ± 1734†	2387 ± 2200†	3822 ± 2057†	2240 ± 1709†
LIS	6300 ± 2678	9273 ± 3747†	**12727 ± 20485**	10866 ± 4812†	8282 ± 2693†
MUR	1800 ± 197	3467 ± 3587†	**6680 ± 17338**	4012 ± 5680†	**2522 ± 2009**
POL	693 ± 208	18333 ± 39334†	1767 ± 1329†	1632 ± 1226†	1466 ± 696†
REN1	993 ± 98	**1100 ± 305**	1253 ± 364†	1435 ± 413†	1331 ± 512†
REN2	2747 ± 484	7253 ± 5130†	4733 ± 2425†	5787 ± 3640†	3744 ± 932†
KUR	1473 ± 249	**1633 ± 443**	1853 ± 528†	2094 ± 497†	2259 ± 554†
FON	6073 ± 326	8393 ± 2435†	7767 ± 1770†	8418 ± 1642†	8480 ± 1175†
QUA	62107 ± 26139	**25847 ± 7826†**	**34580 ± 7530†**	**40134 ± 13983†**	**48262 ± 9071†**
ZDT1	21080 ± 527	**19593 ± 28499**	**18268 ± 27671**	**13133 ± 1631†**	**14273 ± 1723†**
ZDT2	5913 ± 422	17080 ± 7838†	13480 ± 8895†	11873 ± 9947†	**7347 ± 6482**
ZDT3	23407 ± 1110	**10667 ± 1067†**	**11560 ± 1192†**	**12678 ± 1287†**	**14153 ± 1178†**
ZDT4	18633 ± 1889	**14520 ± 383†**	**16613 ± 523†**	20387 ± 745†	22313 ± 704†
ZDT6	38807 ± 1099	**5213 ± 900†**	**5300 ± 853†**	**5474 ± 961†**	**5690 ± 796†**

Note: Symbol † indicates that the difference between NSGA-II and versions of IGMOEA is significant (using a t-test with 0.05 level of significance). The texts in bold style show that the performance of IGMOEA versions is better than or equal to that of NSGA-II.

Table 9. Generation Distance (*Mean and the Standard Deviation* from 30 Runs) Recorded After 20 000 Evaluations for all Different Schemes of POF Division

	NSGA-II	IGMOEA-2	IGMOEA-3	IGMOEA-4	IGMOEA-5
BINH	**0.007 ± 0.000**	0.008 ± 0.000†	0.009 ± 0.001†	0.010 ± 0.001†	0.011 ± 0.001†
LAU	**0.089 ± 0.002**	**0.092 ± 0.008**	0.093 ± 0.008†	0.099 ± 0.012†	0.109 ± 0.012†
LIS	0.002 ± 0.000	**0.001 ± 0.000**†	**0.001 ± 0.000**†	0.002 ± 0.000	0.002 ± 0.000
MUR	**0.000 ± 0.000**	**0.000 ± 0.000**	**0.000 ± 0.000**	**0.000 ± 0.000**	**0.000 ± 0.000**
POL	**0.002 ± 0.005**	**0.001 ± 0.000**	**0.003 ± 0.005**	**0.003 ± 0.005**	**0.003 ± 0.006**
REN1	**0.001 ± 0.000**	**0.001 ± 0.000**	**0.001 ± 0.000**	**0.001 ± 0.000**	**0.001 ± 0.000**
REN2	**0.001 ± 0.000**	**0.001 ± 0.000**	**0.001 ± 0.000**	**0.001 ± 0.000**	**0.001 ± 0.000**
KUR	**0.001 ± 0.000**	**0.001 ± 0.000**	0.002 ± 0.000†	0.002 ± 0.000†	0.002 ± 0.000†
FON	**0.001 ± 0.000**	**0.001 ± 0.000**	**0.001 ± 0.000**	**0.001 ± 0.000**	**0.001 ± 0.000**
QUA	**0.001 ± 0.000**	**0.001 ± 0.000**	0.002 ± 0.000†	0.002 ± 0.000†	0.003 ± 0.000†
ZDT1	0.001 ± 0.000	**0.000 ± 0.000**†	**0.000 ± 0.000**†	0.001 ± 0.000†	0.001 ± 0.000
ZDT2	**0.001 ± 0.000**	**0.002 ± 0.009**	**0.002 ± 0.010**	**0.001 ± 0.002**	**0.001 ± 0.000**
ZDT3	0.001 ± 0.000	**0.000 ± 0.000**†	**0.000 ± 0.000**†	**0.000 ± 0.000**†	**0.000 ± 0.000**†
ZDT4	0.063 ± 0.045	**0.008 ± 0.011**†	**0.031 ± 0.022**†	**0.062 ± 0.046**	0.103 ± 0.074†
ZDT6	0.022 ± 0.004	**0.001 ± 0.000**†	**0.002 ± 0.003**†	**0.001 ± 0.000**†	**0.001 ± 0.003**†

Note: The texts in bold style are to show the best performance. Symbol † indicates that the difference between IGMOEA and NSGA-II is significant (using t-test with 0.05 level of significance).

The above finding is understandable since the total population size was fixed when we changed the number of sub-populations. When the number of sub-populations increased, their size became small (especially the case of five sub-populations). Therefore, it caused IGMOEA difficulties in searching the fitness landscape and resulted in the downgrade of its performance, particularly for multi-modal problem such as ZDT4.

4.6 Population Size Effect

As we hypothesized above, the small sub-population sizes might have caused the poor performance of the local models, since they were unable to capture the local fitness landscape well enough when the problem is highly multi-modal, such as for ZDT4. To test this hypothesis, we used the setting as follows.

- Five sub-populations since it showed the worst performance in the previous section.
- Two different population sizes: previously, the population size for each sub-population was set to 100 individuals (500 individuals overall). Therefore, here we compare two cases of total population size: 200 and 500.
- The test problem was ZDT4, the most difficult problem as shown in the previous results.

The generation distance and hyper-volume ratio that each method achieved over time (up to 100 generations) was visualized in Fig.12.

It is clear from Fig.12 that in the case of 200 individuals, the performance of GUIDED and PSOV2 was worse than that of NSGA-II, while IGMOEA was neutral (it converged quicker than NSGA-II, but was not good at getting a diverse non-dominated set over time). However, when the population size was increased to 500, the situation became different. IGMOEA and PSOV2 were much faster than NSGA-II, particularly IGMOEA was better than all the others in terms of convergence and diversity. These figures also showed a deficiency of GUIDED in refining the non-dominated set over time. GUIDED was better than NSGA-II and PSOV2 during early times (in both cases of population size), but over time, it lost the lead and became slower than the others.

5 Conclusion

This paper proposed a novel technique to guide a localized MOEA using the framework of local models with a controlling strategy that we called PSOV2. Each sphere (local model) was simultaneously focusing on separate areas of the decision and objective space following some guidance. We have called this the technique of dual guidance. A strategy was tested for the guidance of the spheres in the objective space, called guided dominance (GUIDED).

The system using the proposed technique (IGMOEA) was tested against its original version (PSOV2), GUIDED, SPEA2, and NSGA-II over a wide range of test problems. The obtained results showed the superior performance of the technique

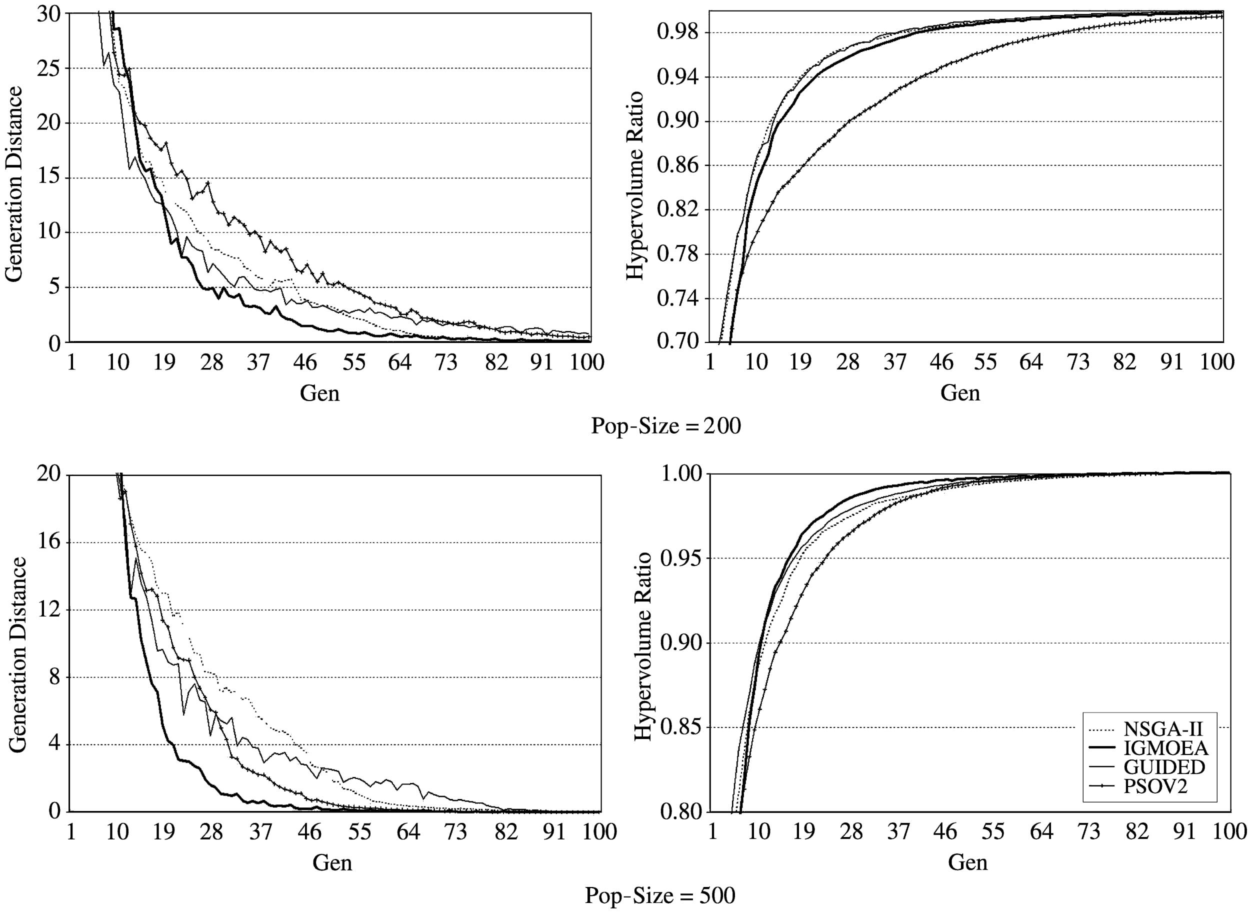


Fig.12. Generation distance and hyper-volume ratio of the four techniques over time for ZDT4 with a total population size of 200 (above) and 500 (below) individuals and up to 100 generations.

using dual guidance in comparison with PSOV2, GUIDED, also SPEA2 and NSGA-II in many problems (particularly, IGMOEA was the best in the second set of test problems). The sensitivity of the system to the POF division schemes and the population size was also investigated. For future work, we intend to investigate the performance of the approach with regard to the property of scalability (in both the decision and objective spaces).

Acknowledgement The authors would like to thank Prof. Carlos C. Coello for providing data sets of some test problems.

References

[1] Deb K. Multiobjective Optimization Using Evolutionary Algorithms. John Wiley and Son Ltd., New York, 2001.

[2] Tan K C, Lee T H, Khor E F. Evolutionary algorithms for multi-objective optimization: Performance assessments and comparisons. *Artificial Intelligence Review*, 2002, 17(4): 251–290.

[3] Tan K C, Khor E F, Lee T H. Multiobjective Evolutionary Algorithms and Applications. Springer-Verlag, 2005.

[4] Coello C A C. Evolutionary multi-objective optimization: A historical view of the field. *IEEE Computational Intelligence Magazine*, 2006, 1(1): 28–36.

[5] Schaffer J D. Multiple objective optimization with vector evaluated genetic algorithms. In *Proc. the First International Conference on Genetic Algorithms*, Hillsdale, New Jersey, 1985, pp.93–100.

[6] Zitzler E, Laumanns M, Thiele M. SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In *Proc. EUROGEN 2001 — Evolutionary Methods for Design, Optimisation and Control with Applications to Industrial Problems*, Athens, Greece, 2001, pp.95–100.

[7] Abbass H A, Sarker R, Newton C. PDE: A Pareto frontier differential evolution approach for multiobjective optimization problems. In *Proc. CEC-2001*, Seoul, Korea, vol 2, IEEE Press, 2001, pp.971–978.

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

[9] Bui L T, Abbass H A, Essam D. Local models: An approach to distributed multi-objective optimization. *Journal*

of Computational Optimization and Applications, Springer. [In Press, DOI: 10.1007/s10589-007-9119-8], 2007.

[10] Deb K, Zope P, Jain A. Distributed computing of Pareto optimal solutions using multi-objective evolutionary algorithms. Technical Report, No. 2002008, KANGAL, IITK, India, 2002.

[11] Zitzler E, Thiele L, Deb K. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 2000, 8(1): 173–195.

[12] Fonseca C M, Fleming P J. Genetic algorithms for multi-objective optimization: Formulation, discussion and generalization. In *Proc. the Fifth International Conference on Genetic Algorithms, San Mateo, California*, Morgan Kauffman Publishers, 1993, pp.416–423.

[13] Horn J, Nafpliotis N, Goldberg D E. A niched Pareto genetic algorithm for multiobjective optimization. In *Proc. The First IEEE Conference on Evolutionary Computation*, Vol.1, IEEE World Congress on Computational Intelligence, Piscataway, New Jersey, 1994, pp.82–87.

[14] Srinivas N, Deb K. Multiobjective optimization using non-dominated sorting in genetic algorithms. *Evolutionary Computation*, 1994, 2(3): 221–248.

[15] Branke J, Kaufler T, Schmeck H. Guiding multi-objective evolutionary algorithms towards interesting regions. Technical Report No. 399. Technical Report, Institute AIFB, University of Karlsruhe, Germany, 2000.

[16] Deb K, Zope P, Jain A. Distributed computing of Pareto optimal solutions with evolutionary algorithms. In *Proc. Evolutionary Multi-Criterion Optimization, LNCS 2632*, 2003, pp.535–549.

[17] Branke J, Schmeck H, Deb K, Maheshwar R S. Parallelizing multiobjective evolutionary algorithms: Cone separation. In *Proc. the Congress on Evolutionary Computation*, Portland, Oregon, USA, IEEE Press, 2004, pp.1952–1957.

[18] Deb K, Sundar J. Reference point based multi-objective optimization using evolutionary algorithms. In *Proc. The 8th Annual Conference on Genetic and Evolutionary Computation, GECCO'06*, New York, NY, USA, 2006, ACM Press, pp.635–642.

[19] Eberhart R C, Shi Y. Particle swarm optimization: Developments, applications and resources. In *Proc. the Congress on Evolutionary Computation*, Piscataway, NJ, USA, IEEE Press, 2001, pp.81–86.

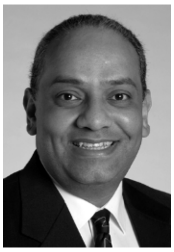
[20] KanGal. Kangal laboratory website. http://www.iitk.ac.in/kangal/codes.shtml, 2006.

[21] Veldhuizen D A V. Multiobjective evolutionary algorithms: Classifications, analyses, and new innovation [Dissertation]. Department of Electrical Engineering and Computer Engineering, Airforce Institute of Technology, Ohio, 1999.

[22] Tan K C, Lee T H, Khor E F. Evolutionary algorithms with dynamic population size and local exploration for multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 2001, 5(6): 565–588.

Lam Thu Bui is a research fellow at the school of ITEE, UNSW@ADFA, University of New South Wales. He holds a Ph.D. degree in computer science from the University of New South Wales. His current research is in the field of evolutionary computation, specialized with evolutionary multi-objective optimization. He is the co-editor of the book "Multi-objective Optimization in Computational Intelligence: Theory and Practice", which will be published by IGI Global, Information Science Reference series in 2008.

Kalyanmoy Deb is a Deva Raj chair professor of mechanical engineering at Indian Institute of Technology, Kanpur, India. His main research interests are in the area of computational optimization, modeling and design, and evolutionary algorithms. He has written two text books on optimization and more than 200 international journal and conference research papers. He is associate editor of two major international journals and an editorial board members of five major journals. He is the recipient of the prestigious Shanti Swarup Bhatnagar Prize in Engineering Sciences for the year 2005. He has also received the "Thomson Citation Laureate Award" from Thompson Scientific for having highest number of citations in computer science during the past ten years in India. He is a fellow of Indian National Academy of Engineering (INAE), Indian National Academy of Sciences, and International Society of Genetic and Evolutionary Computation (ISGEC). He has received Fredrick Wilhelm Bessel Research award from Alexander von Humboldt Foundation in 2003. More information about his research can be found from http://www.iitk.ac.in/kangal/deb.htm.



Hussein A. Abbass is a professor and chair of information technology at the University of New South Wales at the Australian Defence Force Academy (UNSW@ADFA) in Canberra, Australia. He has a B.A., B.Sc., PG-Dip, and Masters degrees all from Cairo University Egypt, an M.Sc. degree from Edinburgh University Scotland, and a Ph.D. degree from QUT Australia. He is the director of the Artificial Life and Adaptive Robotics Laboratory at UNSW@ADFA, an advisory professor at Vietnam National University, Ho-Chi Minh City, a senior member of the IEEE, a senior member of the Australian Computer Society (ACS), the chair of ACS National Committee on Complex Systems, the chair of the IEEE Task Force on Complex Adaptive Systems and Artificial Life, and a member of a number of national and international committees including the IEEE technical committee on Data Mining and the IEEE working group on soft computing in the SMC society. He has 170+ refereed papers and is particularly interested in artificial neural networks, ensemble learning, evolutionary computation, multi-agent systems, and multi-objective optimisation. He has been a technical co-chair and a member of the technical committee for many conferences in the field including a PC co-chair for CEC'07, SEAL 06, and IEEE-Alife'07.

Daryl Essam received his B.Sc. degree from the University of New England in 1990 and his Ph.D. degree from the University of New South Wales in 2000. He was an associate lecturer at the University of New England from 1991 and has been with the Australian Defense Force Academy campus of the University of New South Wales since 1994, where he is now a senior lecturer.