CHAPTER 30

USING DIVERSITY TO GUIDE THE SEARCH IN MULTI-OBJECTIVE OPTIMIZATION

J.D. Landa Silva and E.K. Burke
Automated Scheduling, Optimisation and Planning Research Group
School of Computer Science and Information Technology
University of Nottingham, UK
E-mail: jds@cs.nott.ac.uk, ekb@cs.nott.ac.uk

The overall aim in multi-objective optimization is to aid the decision-making process when tackling multi-criteria optimization problems. In an *a posteriori* approach, the strategy is to produce a set of non-dominated solutions that represent a good approximation to the Pareto optimal front so that the decision-makers can select the most appropriate solution. In this paper we propose the use of diversity measures to guide the search and hence, to enhance the performance of the multi-objective search algorithm. We propose the use of diversity measures to guide the search in two different ways. First, the diversity in the objective space is used as a helper objective when evaluating candidate solutions. Secondly, the diversity in the solution space is used to choose the most promising strategy to approximate the Pareto optimal front. If the diversity is low, the emphasis is on exploration. If the diversity is high, the emphasis is on exploitation. We carry out our experiments on a two-objective optimization problem, namely space allocation in academic institutions. This is a real-world problem in which the decision-makers want to see a set of alternative diverse solutions in order to compare them and select the most appropriate allocation.

1. Introduction

This paper is concerned with the application of the class of approaches known as meta-heuristics to tackle multi-objective optimization problems. We assume that the reader is familiar with the fields of multi-criteria decision-making[2,39] and multi-objective optimization[7,10]. Recent surveys on the application of meta-heuristics to multi-objective optimization problems are those provided by Jones *et al.*[19], Tan *et al.*[42] and Van Veldhuizen and Lamont[45]. Multi-objective optimization is a very active research area that has received increased attention from the scientific community and from practitioners in the last ten years or so. One main reason for this is that many real-world problems are multi-criteria optimization problems. This means that in these problems, the quality of solutions is measured taking into account several criteria that are in partial or total conflict. Therefore, there is no such global optimum solution but a number of them that represent a trade-off between the various criteria. It is also commonly the case that more than one decision-maker is involved in the selection of the most appropriate solution to the multi-criteria problem. Then, the overall aim in multi-objective optimization is to aid the decision-makers to tackle this type of problems. One of the strategies for this is to produce a set of solutions that represent a good approximation to the trade-off surface. Then, the decision-makers can decide which of the solutions in this set is the most adequate for the problem at hand. In general terms, a good approximation set should be as close as possible to the optimal front and it should also give a good coverage of the optimal front. The goal of achieving a good coverage of the trade-off surface, i.e. maintain the diversity and spread of solutions, is of particular interest in multi-objective optimization. A number of techniques to accomplish this goal have been proposed in the literature, e.g. weighted vectors, clustering or niching methods (fitness sharing, cellular structures, adaptive grids, *etc.*), restricted mating, relaxed forms of dominance, helper objectives, and objective-driven heuristic selection (hyper-heuristics). Most of these techniques are targeted towards maintaining diversity in the objective space. However, in some scenarios, the decision-makers are also concerned with the diversity of solutions in the solution space. Then, to serve as a useful tool in tackling multi-criteria optimization problems, the multi-objective optimization algorithm should have the mechanisms to find the set of solutions that satisfy the requirements of the decision-makers. That is, solutions that are close to the optimal front and have the desired diversity in the objective space, the solution space or both spaces. One goal in this paper is to present an overview of a number of techniques that have been proposed in the literature to maintain a diverse set of solutions when tackling multi-objective optimization problems. Another goal here is to describe some mechanisms that we implemented to help a multi-objective search algorithm to obtain a diverse set of solutions for a real-world optimization problem with two objectives. These mechanisms consist on using diversity measures, in both the objective space and the solution space, to guide the search and enhance the performance of the multi-objective search algorithm. We carry out experiments on three tests instances of the space allocation in academic institutions. In this problem, a set of entities (staff, computer rooms, teaching rooms, *etc.*) must be allocated into a set of available areas of space or offices and a number of additional constraints should also be satisfied. In the space allocation problem, the decision-makers are interested in the diversity of solutions in both the objective space and the solution space. The results of our experiments show that the proposed mechanisms help the algorithm to produce a set of compromise solutions that better satisfies the requirements from the decision-makers. The rest of this paper is organized as follows. Section 2 discusses the issue of diversity in the context of multi-objective optimization. Section 3 gives an overview of some of the mechanisms incorporated into modern multi-objective search algorithms to achieve a good coverage of the trade-off surface. A description of the two-objective space allocation problem and the way in which diversity in the objective space and diversity in the solution space are measured in this problem are the subject of Sec. 4. The diversity control mechanisms implemented to guide the search and the algorithm in which these mechanisms were incorporated are described in Sec. 5. The experiments and results are presented and discussed in Sec. 6 while Sec. 7 gives a summary of this paper.

2. Diversity in Multi-objective Optimization

Given two solutions x and y for a k-criteria optimization problem, x is said to *weakly dominate* y if x is as good as y in all the k criteria and better in at least one of them. In the case that x is better than y in all the k criteria, x is said to *strictly dominate* y. In the following, we refer to *weak dominance* simply as *dominance*. A solution x is said to be *non-dominated* with respect to a set of solutions S is there is no solution in S that dominates x. The Pareto optimal front denoted S_P is the set of all non-dominated solutions with respect to the whole set of feasible solutions S_F. Then, the goal of a multi-objective search algorithm is to find a set S_{ND} of non-dominated solutions for a given multi-criteria optimization problem. The non-dominated set S_{ND} should represent a good approximation to the Pareto optimal front S_P. This means that the solutions in S_{ND} should be:

- As close as possible to the Pareto optimal front S_P,
- widely spread across the entire trade-off surface, and
- uniformly distributed across the entire trade-off surface.

The closeness of S_{ND} to the Pareto optimal front S_P gives an indication of how good is the convergence towards the optimal front. The spread and distribution of S_{ND} give an indication of how good is the coverage of the Pareto optimal front S_P. This is illustrated in Fig. 1 where various non-dominated sets are depicted for a two-objective minimization problem. Using the notation in Fig. 1, it is clear that an effective multi-objective search algorithm should find an approximation set with the characteristics of S1(c^+,s^+,d^+). Moreover, in some real-world scenarios the decision-makers are interested on a set of alternative solutions like those in S1 but at the same time, they want to see solutions that have a certain diversity with respect to the solution space. This is the case for the problem tackled in this paper, space allocation in academic institutions, as it will be explained later. Then, in order to achieve the aim of assisting the decision-making process, a multi-objective search algorithm must also take into account the diversity of S_{ND} with respect to the solution space.

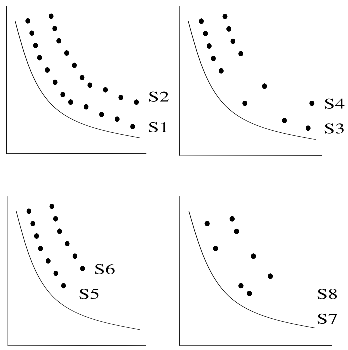

Fig. 1. The quality of the non-dominated set is given by the closeness to the Pareto optimal front (c^+ is close, c^- is far), the spread of solutions (s^+ is good spread, s^- is poor spread) and the distribution of solutions (d^+ is good distribution, d^- is poor distribution). Then, the quality of the non-dominated sets in this Fig. can be described as follows: S1(c^+,s^+,d^+), S2(c^-,s^+,d^+), S3(c^+,s^+,d^-), S4(c^-,s^+,d^-), S5(c^+,s^-,d^+), S6(c^-,s^-,d^+), S7(c^+,s^-,d^-), and S8(c^-,s^-,d^-).

3. Maintaining Diversity in Multi-Objective Optimization

The majority of meta-heuristics proposed for multi-objective optimization incorporate a specialized mechanism to help achieving a good diversity with respect to the objective space. As it was pointed out by Laummans *et al.*[28], this is not a straightforward task because many algorithms that implement specific mechanisms to maintain diversity suffer from deterioration which affect their convergence ability. This Sec. gives an overview of a number of strategies that have been proposed in the literature to maintain diversity in multi-objective optimization. For more references to multi-objective optimization algorithms that incorporate mechanisms for diversification not discussed here, see the survey by Tan *et al.*[42] and also the books by Coello Coello *et al.*[7] and Deb[10].

3.1. *Weighted Vectors*

One of the first techniques that were proposed to achieve a better diversity of solutions in multi-objective optimization is the use of weighted vectors to specify the search direction and hence, aim a better coverage of the trade-off surface. This method consists on setting a vector of k weights $W = [w_1, w_2, \ldots w_k]$ where $0 \leq w_i \leq 1$, k is the number of objectives in the problem and the sum of all w_i equals 1. The fitness of a solution x is calculated as $f(x) = w_1 f_1(x) + w_2 f_2(x) + \ldots w_k f_k(x)$ where $f_i(x)$ measures the quality of x with respect to the i^{th} criterion. The strategy is to systematically generate a set of vectors in order to approach the trade-off surface from all directions. Weighted vectors is a popular technique that has been used in a number of algorithms like the multi-objective cellular genetic algorithm of Murata *et al.*[35] and the multiobjective simulated annealing algorithm of Ulungu *et al.*[43]. Another approach that uses weighted vectors to encourage diversity is the Pareto simulated annealing algorithm of Czyzak and Jazkiewicz[8]. Their strategy is to modify the weights for a solution x so that x is moved away from its closest neighbor x_{cn} by increasing the weights of those objectives in which x is better than x_{cn} and decreasing the weights for those objectives in which x is worse than x_{cn}. In another approach implemented by Gandibleux *et al.*[14], the set of supported solutions is first computed. Then, the information obtained from these solutions is used to guide the search and improve the performance of a population heuristic. Ishibuchi *et al.*[15] used weight vectors in a different way to encourage diversity. Instead of generating a weighted vector to specify a search direction for a solution, they choose an appropriate solution for a randomly specified weight vector. The selection of the solution for a given vector is based on the position of the solution in the objective space. That is, they attempt to set an appropriate search direction for each new solution in order to achieve a better approximation set.

3.2. *Fitness Sharing*

In this mechanism the idea is to decrease the fitness of individuals that are located in crowded regions in order to benefit the proliferation of solutions in sparse regions. Usually, the fitness of an individual is reduced if the distance to its closer neighbor is smaller than a predefined value. Fitness sharing can be implemented in the objective space or in the solution space. However, most of the implementations of fitness sharing reported in the literature are on the objective space. For example, Zhu and Leung[47] implemented fitness sharing in their asynchronous self-adjustable island genetic algorithm. Talbi *et al.*[41] implemented fitness sharing mechanisms in both the objective space and the solution space. In their experiments, they observed that fitness sharing in the objective space appears to have a stronger influence on the search than fitness sharing in the solution space, but they also noted that the combination of both fitness sharing mechanisms improved the search.

3.3. *Crowding/Clustering Methods*

These methods attempt to control the number of solutions in each region of the trade-off surface. The general idea here is to limit the proliferation of solutions in crowded or over-populated areas and at the same time, to encourage the proliferation of solutions in sparse or under-populated areas. An example of this type of mechanisms is the adaptive grid implemented by Knowles and Corne[22] in their Pareto archived evolutionary strategy. They divide the k-objective space into $2^{l \cdot k}$ regions where l is the number of bisections in each of the k dimensions. Then, based on the crowdedness of the region in which the new solution lies, a heuristic procedure is used to decide if the new solution is accepted or not. Lu and Yen[29,30] used a modified version of the adaptive grid of Knowles and Corne. In their algorithm they modify the fitness of solutions based on the density value of the population. They also associate an age indicator to each solution x in the population in order to control its life span.

An agent-based crowding mechanism was proposed by Socha and Kisiel Dorohinicki[40] in which agents interact between them in order to encourage the elimination of too similar solutions or agents. Each agent in the population contains an amount of energy and the crowding mechanism seeks to maintain a uniform agent distribution along the trade-off surface and prevent agent clustering in particular areas by discouraging agents from creating groups of similar solutions. In their mechanism, an agent A communicates with another agent B and then the solutions from both agents,

x_A and x_B respectively, are compared. If the similarity between x_A and x_B (measured with a distance metric) is smaller than a predefined value, an amount of energy is transferred from agent A to agent B. The amount of energy transferred depends on the degree of similarity between x_A and x_B. This is similar to fitness sharing but here, one agent receives and the other provides.

3.4. *Restricted Mating*

Restricted mating is a mechanism that prevents the recombination of individuals that do not satisfy a predefined criterion. Most of the times, this criterion is that mating individuals should not be too close to each other in the objective space or in the solution space. In this sense, restricted mating can be regarded as a mechanism that is similar to crowding/clustering. An example of restricted mating is the strategy implemented by Kumar and Rockett[21] in their Pareto converging genetic algorithm. That algorithm is an island based approach in which the genetic operations are restricted to individuals within the same island. There is no migration between islands and no cross-fertilization between individuals in two different islands. However, two islands can be merged into one island in order to test convergence during the search. Their algorithm is a steady-state approach that produces only one offspring in each iteration. Kumar and Rocket argue that the steady-state nature of the algorithm helps maintain diversity because genetic drift, which is inherent in generational genetic algorithms, is less likely to occur. Other examples of restricted mating mechanisms are used in the approaches implemented by Lu and Yen[29,30] and the cellular genetic algorithm of Murata *et al.*[35].

3.5. *Relaxed Forms of Dominance*

Another strategy to encourage diversity that has been explored recently by several researchers, is to use relaxed forms of the dominance relation to assess the fitness of individuals. As described in Sec. 2, in the standard dominance relation a solution x is considered better than another solution y only if x is not worse than y in all the objectives and x is better that y in at least one of the objectives. In the relaxed forms of dominance, the basic idea is to consider a solution x as better that a solution y even if x is worse that y in some objective(s). Usually, the condition is that such deterioration must be compensated by a good improvement in the value of other objective(s). The idea is that by using relaxed forms of dominance, the algorithm will be capable of exploring more of solutions and hence, to maintain a better diversity. For example, Laumanns *et al.*[28] proposed the use of ϵ-dominance to implement archiving/selection strategies that permit to achieve a better convergence and distribution of the approximation non-dominated set. Burke and Landa Silva[3] used a variant of α-dominance, which is also a relaxed form of dominance, to improve the converge ability of two multi-objective search algorithms. Mostaghim and Teich[34] compared the performance of a multi-objective optimization algorithm when using a clustering technique and when using the ϵ-dominance method. They observed in their experiments that using ϵ-dominance to update the archive of non-dominated solutions, was beneficial because it helped to reduce the computation time and it also helped to achieve a better convergence and comparable diversity.

Another interesting aspect of using relaxed forms of the dominance relation is that it can help to identify those solutions that are more attractive to the decision-makers out of the set of solutions in the trade-off surface, which can be of considerable size. As it was pointed out by Farina and Amato[13], the number of solutions that can be considerable equal or incomparable (based on standard dominance) to the current solution, increases considerably with the number of objectives. They developed the notion of k-dominance in which they proposed to take also into consideration the number of incomparable or equal objectives in the new solution and the normalized size of improvement achieved in the other objectives. In k-dominance v_1 k-dominates v_2 if and only if:

$$n_e < M \text{ and } n_b \geq \frac{M-n_e}{k+1} \text{ where } 0 \geq k \leq 1$$

In the above, n_b is the number of objectives in which v_1 is better than v_2, and n_e is the number of objectives in which v_1 and v_2 are equal. Farina and Amato also extended k-dominance by evaluating the number of n_b, n_e, in a fuzzy way instead of a crisp way by introducing a tolerance on the i_{th} objective, that is the interval at which an improvement on objective i is meaningless. Jin and Wong[17] also investigated archiving techniques based on their concept of relaxed dominance, called E-dominance. The main feature of their archiving mechanism is that it adapts according to the solutions that have been found. It also includes the concept of hyper-rectangles to enclose the search space even considering unseen solutions. This gives their technique the advantage of not requiring prior knowledge of the objective space (objective values).

3.6. *Helper Objectives*

The specification of helper objectives is a strategy that has been used to aid the search not only in multi-objective optimization but also in single-objective optimization. For example, this mechanism can be used to handle constraints by treating each constraint as additional objective to be optimised. In single-objective optimization the aim of helper objectives is help on maintaining diversity and escaping from local optima. For example, Jensen[16] and Knowles *et al.*[20] proposed the 'multi-objectivization' of single-objective optimization problems which is decomposing the single-objective problem into subcomponents by considering multiple objectives. In this way, 'multi-objectivization' can help to remove local optima because for the search process to be stuck it is required that all objectives are stuck. The helper objectives should be chosen so that they are in conflict with the main objective, at least partially.

3.7. *Objective Oriented Heuristic Selection*

Another idea that has been proposed to help maintaining diversity in multi-objective optimization is to adapt the local search strategy according to the current distribution of solutions in the objective and/or the solution space. For example, Knowles and Corne[23] proposed to adapt the focus of the search on exploration or exploitation when approximating the Pareto front, by selecting the most adequate between three search strategies: 1)use a population-based method that tries to improve in all objectives at once in order to approach the Pareto front from all directions, 2)generate a weighted vector which is used to specify a specific search direction, or 3)use a single-solution local search method that tries to move along the Pareto front by perturbing one solution and obtain a nearby point in the front.

The selected strategy depends on the correlation between distance in the solution space and distance in the objective space. This strategy was also investigated by Jin and Sendhoff[18] for some continuous test problems. Adapting the local search heuristic according to the value of the objectives in the solutions has also been proposed as a mechanism to maintain diversity while converging to the Pareto front. For example, Burke *et al.*[5] implemented an approach that has been termed 'hyper-heuristic'. The idea is to use a guiding/learning method that choses the most promising heuristic in order to push solutions towards the desired area in the objectives of interest. This technique takes into consideration the localization of the solution in the objective space and the ability of each local search heuristic to achieve improvements on each objective. The idea is to try improving poor objectives while maintaining the rich ones. Adapting the heuristic local search is interesting when using hybrid approaches that use local search in an efficient way. Then, the analysis or pre-sampling of the fitness landscape can be useful to design a good hybrid[32].

3.8. *Using Diversity to Guide the Search*

Various evolutionary algorithms for multi-objective optimization use estimators of density in the objective space to bias the selection operator in order to maintain diversity in the population. Laumanns *et al.*[27] noted that the accuracy of the density estimator used has a strong effect on the performance of the selection strategy and hence, the density estimator should be good for the diversity maintenance strategy to be effective. Also, Knowles *et al.*[25] proposed a bounded archiving technique that attempts to maximize the hypervolume covered by the approximation set. They compared the performance of their archiving technique against other methods and obtained promising results. However, they pointed out that the computational cost was considerable for more than three objectives.

In single-objective optimization some researchers have also made some efforts towards designing evolutionary algorithms that maintain diversity in an adaptive fashion by using diversity measures to guide the search. For example, Ursem[44] proposed a diversity-guided evolutionary algorithm that alternates between phases of exploration and exploitation according to a measure of the diversity in the population given by the distance to the average point. If the diversity is below a threshold d_{low}, the algorithm uses selection and recombination in an exploration mode. If the diversity is above a threshold d_{high}, the algorithm uses mutation in an exploitation mode. Another approach that uses diversity measures to guide the search is the diversity-control-oriented genetic algorithm of Shimodaira[38] in which the probability of individuals to survive depends on the Hamming distance between the individual and the best individual in the population.

4. The Two-Objective Space Allocation Problem

The management of physical space in universities is an important and difficult issue as it was discussed by Burke and Varley[6]. With the continuous increase in the number of students and staff, it must be ensured that the available estate is used as efficiently as possible while simultaneously satisfying a considerable number of constraints. The allocation of office space to staff, postgraduate students, teaching rooms, computers rooms, *etc.* is carried out manually is most universities. This is a process that takes a considerable amount of time and effort from the space administrators. More importantly, this manual distribution usually provokes an inefficient utilization of the available estate.

4.1. *Problem Description*

The space allocation problem can be briefly described as follows. Given a set of n entities (people, teaching rooms, computer rooms, *etc.*) and a set of m available rooms, the problem is to allocate all the n entities into the m rooms in such a way that the office space is used as efficiently as possible and the additional constraints are satisfied. Each entity requires a certain amount of space[a] according to university regulations and each room has a given capacity. It is very unlikely that the capacity of a room matches exactly the amount of space required by the entities allocated to the room. Let c_i be the capacity of the i^{th} room and let s_i be the space required by all the entities allocated to the room. Then, if $c_i > s_i$, space is said to be wasted, while if $c_i < s_i$, space is said to be overused. It is less desirable to overuse space than to waste it. The overall space utilization efficiency is measured by the amount of space that is being misused, i.e. space wasted plus space overused for all rooms (space misuse is represented by F_1). In addition to this, space administrators should ensure that certain constraints are satisfied. Some constraints are *hard*, i.e. they must be satisfied while other constraints are *soft*, i.e. their violation should be minimized. The number of different types of constraints varies considerably between problem instances but in general, the constraints limit the ways in which the entities can be allocated to rooms. For example, two professors must not share a room, a computer room should be allocated in the ground floor and adjacent to a seminar room, teaching rooms must be away from noisy areas, postgraduate students in the same research group should be grouped together, *etc.* The penalty applied when a constraint is violated depends on the type of constraint and it may also vary from one problem instance to another (soft constraints violation is represented by F_2). A solution or allocation is represented here by a vector $\Pi = [\pi_1, \pi_2, \ldots, \pi_n]$ where each $\pi_j \in \{1, 2, \ldots, m\}$ for $j = 1, 2, \ldots, n$ indicates the room to which the j^{th} entity has been allocated.

In a multi-criteria optimization problem, the criteria can be conflict-

[a]Note that here, space is the floor area usually measured in m^2.

ing, harmonious or independent and this has an influence on the difficulty to achieve a good approximation to the Pareto front as it was discussed by Purshouse and Fleming[37]. The existence of conflicting criteria makes more difficult to achieve a good convergence. If the criteria are harmonious, convergence is not affected but achieving a good diversity may be more difficult because it is very probable that solutions will have similar values in the harmonious criteria. If the criteria are independent, it is possible to decompose the problem and then to use a divide and conquer strategy to solve it. An investigation into the conflicting nature of the criteria in the space allocation problem was carried out by Landa Silva[26]. In that investigation it was found that in general, the minimization of space wastage is not in conflict with the minimization of space overuse and that the satisfaction of different types of soft constraints is not in conflict with each other. However, it was also found that the minimization of space misuse (overuse and wastage) is in strong conflict with the minimization of soft constraints violation. Therefore, we consider two objectives in the space allocation problem:

(1) Minimization of space misuse, i.e. minimization of F_1.
(2) Minimization of soft constraints violation, i.e. minimization of F_2.

In this problem, space administrators often know of additional constraints which are not (or cannot for political reasons) be explicitly built into the objectives. For example, when two members of staff have a personality clash and cannot be allocated in the same room. Another common example is when people have a preference for certain type of rooms. Therefore, in this context, the aim of a multi-objective optimization algorithm is to aid the space administrators by finding a set of alternative high-quality allocations. Space administrators usually want to see a set of allocations which are very similar in certain aspects while being very different in other aspects. For example, administrators may want to see two or more alternative solutions in which the teaching areas are allocated to the same rooms in each of the allocations but with different ways of distributing offices to people. Another example is when the allocation needs to be re-organized and the space administrators want to explore alternative non-dominated solutions that are very similar to the existing distribution in order to avoid major disruptions. Then, in the space allocation problem it is important to take into consideration the diversity of the set of allocations with respect to the solution space.

Besides its practical interest, the space allocation problem as described here is of scientific importance because it can be formulated as a variant of the multiple knapsack problem which is an important problem in combinatorial optimization (see Dawande *et al.*[9] and Martello and Toth[31]).

4.2. *Measuring Diversity of Non-dominated Sets*

There are various papers in the literature that propose, compare and discuss indicators to assess the performance of multi-objective optimization algorithms. These include those by Knowles and Corne[24] Ang *et al.*[1], Farhang-Mehr and Azarm[12], Tan *et al.*[42], Okabe *et al.*[36] and others. Assessing the diversity (in the solution space or in the objective space) of a non-dominated set is a difficult task because, as it was discussed in Sec. 2, the diversity should be measured in terms of the distribution and the spread of solutions in the set. Some of the indicators proposed in the literature seek to evaluate the quality of the spread and the distribution of solutions. For example, the S metric of Zitzler and Thiele[48] calculates the hypervolume of the k-dimensional region covered by the approximation set. But a reference point must be given in order to compute the hypervolume and the location of this reference point may have an influence on how two or more non-dominated sets compare. Deb *et al.*[10] proposed a spacing metric designed to measure how evenly points are distributed. That metric is based on computing the Euclidean distance between each pair of non-dominated solutions and it also requires the boundary solutions. Another spacing metric which is also based on the Euclidean distance between pairs of non-dominated solutions is the one described by Van Veldheuzien and Lamont[46]. Other metrics that have been proposed to estimate the diversity of a population of solutions are based on entropy as proposed by Farhang-Mehr and Azarm[11]. These metrics require the division of the objective space into a cellular structure. A high entropy value indicates a better distribution of solutions across the trade-off surface because it measures the flatness of the distribution of solutions or points.

In this paper, diversity in the objective space is measured using a population metric proposed by Morrison and De Jong[33]. We have selected this metric because it does not require reference solutions and it is also related to the Hamming and Euclidean distances between solutions. The metric by Morrison and De Jong is inspired on concepts of mechanical engineering, specifically on the moment of inertia which measures mass distribution of an object. The centroid of a set of p points in a k-dimensional space has coordinates given by eq. 1, where $x_{i,j}$ is the value of the i^{th} dimension in the j^{th} point. The measure of diversity for the population of p points, based on their moment of inertia is given by eq. 2. The higher the value of I, the higher the diversity of the set of p points.

$$c_i = \frac{\sum_{j=1}^{p} x_{i,j}}{p} \quad \text{for } i = 1, 2, \ldots k \tag{1}$$

$$I = \sum_{i=1}^{k} \sum_{j=1}^{p} (x_{i,j} - c_i)^2 \tag{2}$$

To measure diversity in the solution space, the metric used should provide a meaningful way to express the similarity between solutions for the problem at hand. Therefore, we have designed a specific way of measuring diversity in the solution space for the space allocation problem. Equation 3 gives the percentage of non-similarity or variety used here as a measure of diversity for a set of allocations, where $D(j)$ is the number of different values in the j^{th} position for all the p vectors representing the solutions. Figure 2 illustrates how the percentage of variety is calculated for a set of $p = 5$ allocations.

$$V = \frac{\sum_{j=1}^{n} \frac{D(j)-1}{p-1}}{n} \cdot 100 \tag{3}$$

	Five strings representing allocations						
	A	A	A	A	A	A	A
	A	A	B	B	A	B	B
	A	B	B	C	B	C	C
	A	B	B	C	B	D	D
	A	B	B	C	C	D	E
D(j)	1	2	2	3	3	4	5
(*D(j)* – 1) / (*p* – 1)	0	0.25	0.25	0.50	0.50	0.75	1
V(p) = (3.25 / 7) x 100 = 46.42 %							

Fig. 2. Calculation of the percentage of variety V for a set of $p = 5$ allocations. The number of entities is $n = 7$ and the number of rooms is $m = 5$.

5. Using Diversity to Guide the Search

In this Sec. we describe the strategies that we implemented in order to obtain approximation sets that better satisfy the requirements from the decision makers in the space allocation problem. The diversity indicators I (eq. 2) and V (eq. 3) described above are used to guide the search and find sets of non-dominated solutions that are diverse with respect to both the solution space and the objective space.

5.1. *Diversity as a Helper Objective*

We use the diversity in the objective space as a helper objective in order to decide when a candidate solution is considered attractive. Let P be a population of solutions from which a solution x is used to generate a candidate solution x'. Then, I (eq. 2) indicates the diversity of the set P while I' indicates the diversity of the set P' in which x is replaced by x'. We use the expression u $dominates(c_1, c_2, ...)$ v to indicate that the criteria $c_1, c_2, ...$ are used to determine dominance between vectors u and v. Then, a candidate solution x' is considered attractive if x' $dominates(F_T, I)$ x where $F_T = F_1 + F_2$. That is, x' is considered better than x if $F_T(x') < F_T(x)$ and $I' \geq I$ or if $F_T(x') \leq F_T(x)$ and $I' > I$. Note that we use the aggregated value F_T instead the individual criteria F_1 and F_2. This is because in our previous research we have observed that the aggregation method was more beneficial than the dominance relation[b] for the overall performance of our algorithm over all set of instances (see Burke and Landa Silva[4]). Then, a candidate solution is accepted if it has better fitness (F_T) without worsening the diversity in the objective space (I) or if it has the same fitness value but it improves the diversity in the objective space.

5.2. *Diversity to Control Exploration and Exploitation*

We use the diversity measure in the solution space to alternate between the phases of exploration and exploitation in our algorithm. This is similar to the strategy implemented by Ursem[44] in single-objective optimization. As it was discussed above, the measure V (eq. 3) is an indication of how diverse a set of allocations is considered by the space administrators. The value of $V(P_{ND})$ is used to control the algorithm search strategy, where

[b] We also found that using relaxed forms of dominance (see Sec. 3.5) seems to improve the performance of our algorithm but only in some problem instances.

P_{ND} is the current set of non-dominated solutions. First, two threshold values are set, V_{good} is the diversity that is considered as 'good' in the obtained set of non-dominated solutions and V_{min} is the minimum diversity that is accepted in the obtained set of non-dominated solutions. Then, when $V(P_{ND}) \geq V_{good}$ the algorithm is in exploitation mode and when $V(P_{ND}) < V_{min}$ the algorithm enters the exploration mode. In exploitation mode, the algorithm attempts to find better solutions by using local search only. In exploration mode, the algorithm uses local search and a specialized mutation operator in order to increase the diversity $V(P_{ND})$ of the current set of non-dominated solutions. Based on our previous experience[26] with the space allocation problem, we set $V_{good} = 70\%$ and $V_{min} = 30\%$ in our experiments.

5.3. *The Population-based Hybrid Annealing Algorithm*

Our algorithm is a population-based approach in which each individual is evolved by means of local search and a specialized mutation operator. The algorithm is shown in pseudocode 1 and is a modified version of our previous approach described elsewhere[4]. The modification consists on adding the mechanisms described above to guide the search based on the diversity measures. The population P_C contains the current solution for each individual. The population P_B contains the best solution (in terms of F_T) found by each individual so far. The population P_{ND} is the external archive of non-dominated solutions. A common annealing schedule is used to control the evolution process of the whole population by means of the global acceptance probability ρ (steps 6.3 and 6.4). The local search heuristic H_{LS} selects the type of move from *relocate*, *swap*, and *interchange* if all the n entities are allocated. If there are unallocated entities (this occurs when the specialized mutation operator is applied as described below), then H_{LS} employs the *allocate* move. *Relocate* moves an entity from one area to another, *swap* exchanges the assigned areas between two entities, *interchange* exchanges all the allocated entities between two areas, and *allocate* finds a suitable area to allocate an unallocated entity. The local search heuristic H_{LS} incorporates a *cooperation mechanism* to encourage information sharing between individuals in the population. This *cooperation mechanism* maintains two matrices M_T and M_A of size $n \times m$ in which the cell (j, i) indicates the allocation of the j^{th} entity to the i^{th} area. M_T stores pairs (entity,area) that are considered tabu for a number of iterations while M_A stores those that are considered attractive during the search.

Pseudocode 1. The Population-based Hybrid Annealing Algorithm.

1.Generate the initial current population of solutions P_C
2.Copy P_C to the population of best solutions P_B
3.Set acceptance probability $\rho \leftarrow 0$, cooling factor $0 < \alpha < 1$, decrement step η, re-heating step φ, and re-heating counter $\tau \leftarrow 0$ (η, φ and τ are a number of iterations)
4.For η iterations, apply the local search heuristic H_{LS} to each individual in P_C
5.Set $\rho \leftarrow 1$, $mode =$ exploitation
6.For each X_C in P_C and its corresponding X_B in P_B,
  6.1.Generate a candidate solution X'_C using H_{LS}
  6.2.If X'_C $dominates(F_T, I)$ X_C, then $X_C \leftarrow X'_C$
    a)If X'_C $dominates(F_T, I)$ X_B, then $X_B \leftarrow X'_C$
  6.3.If X'_C $dominates(F_T, I)$ X_C is false, then
    a)if $\rho > 0$ and a random generated number in the normal distribution [0,1] is smaller than ρ, then $X_C \leftarrow X'_C$
    b)if $\rho \approx 0$ (in our setting, if $\rho < 0.0001$), then $\tau \leftarrow \tau + 1$ and if $\tau \geq \varphi$, then $\rho \leftarrow 1$ and $\tau = 0$
  6.4.If ($iterations \bmod \eta$) = 0, then $\rho \leftarrow \alpha \cdot \gamma$
  6.5.If no solution in P_{ND} $dominates(F_1, F_2)$ X'_C, update P_{ND}
7.If $mode =$ exploitation and $V(P_{ND}) < V_{minimum}$, then $mode \leftarrow$ exploration
8.if $mode =$ exploration and $V(P_{ND}) \geq V_{good}$, then $mode \leftarrow$ exploitation
9.If $mode =$ exploration, then apply the specialized mutation operator to each individual in P_C
10.If stopping criterion has not been satisfied, go to Step 6

When a move produces a detriment in the fitness of the solution, M_T is updated as $M_T(j, i) = \text{iterations} + tenure$ which indicates that moves involving that pair are considered tabu for $tenure \approx n$ iterations. When a move produces an improvement in the fitness of the solutions, M_A is updated as $M_A = M_A + 1$ to indicate that the higher the value of the cell, the more attractive the moves involving that pair are considered. The purpose of the specialized *mutation operator* is to disturb solutions in a controlled way in order to promote exploration. This operator unallocates a maximum of $n/5$ entities from their assigned area of space. The entities to be unallocated are selected in decreasing order of their associated penalty (violation of soft constraints associated to the entity). The entities that are unallocated by the mutation operator are re-allocated by the heuristic H_{LS}.

In the algorithm presented in pseudocode 1, the diversity I is used as a helper objective in steps 6.2 and 6.3 while the diversity $V(P_{ND})$ is used to guide the search in steps 7-9. In our previous approach[4], the preference of the candidate solution X'_C over X_C and X_B in steps 6.2 and 6.3, is based solely on the value of F_T. The other difference in our previous implementation is that the specialized mutation operator (steps 7-9) is applied when no individual in P_B has achieved an improvement for η iterations instead of being controlled by the diversity in the solution space as proposed here.

6. Experiments and Results

The purpose of our experiments was to investigate if the mechanisms described above to guide the search based on the diversity measures I (eq. 2) and V (eq. 3) help our algorithm to find better sets on non-dominated solutions. Here, we are interested in finding sets of non-dominated allocations that have a good spread and distribution in the objective space but also have high diversity in the solution space. We compared the performance of the algorithm presented in pseudocode 1 to our previous implementation using the same real-world data sets **nott1**, **nott1b** and **trent1** described in that paper[4] (these test instances are available from `http://www.cs.nott.ac.uk/~jds/research/spacedata.html`).

6.1. *Experimental Setting*

For each test instance, we executed 10 runs of our algorithm described in pseudocode 1 and 10 runs of the previous implementation. In each pair of runs, the same initial set of solutions was used for the two algorithms. In each run, the stopping criterion was a maximum number of solution evaluations set to 100000, 80000 and 50000 for **nott1**, **nott1b** and **trent1** respectively. The parameters for the algorithm were set as in our previous paper[4]: $|P_C| = |P_B| = 20$, $\alpha = 0.95$, $\eta = n$ and $\varphi = 10 \cdot n$. We compared the two algorithm implementations with respect to the online and the offline performance. For the online performance, we directly compare the P_{ND} sets obtained by the algorithms in each run. For the offline performance, an overall non-dominated set is obtained for each algorithm by merging all the 10 P_{ND} sets produced. A visual comparison of two non-dominated sets found by the two algorithms was not possible because no evident difference was observed in the bi-dimensional graph with axis F_1 and F_2. Therefore, we used four criteria to compare two sets of non-dominated solutions: the diversity in the objective space I (eq. 2), the diversity in the solution space

V (eq. 3), the number of non-dominated solutions found $|P_{ND}|$ and the C metric of Zitzler *et al.*[49] which is given by eq. 4. If $C(A, B) = 1$, all solutions in set B are dominated by at least one solution in set A. If $C(A, B) = 0$, no solution in set B is dominated by a solution in set A. We used the C metric because it directly compares the quality of two non-dominated sets, it is simple to compute and it does not require knowledge of the Pareto optimal front.

$$C(A, B) = \frac{|\{b \in B; \exists a \in A : a \preceq b\}|}{|B|} \tag{4}$$

We carried out our experiments on a PC with a 3.0GHz processor, 768MB of memory and running on Windows XP. The algorithms were coded on MS Visual C++ version 6.0.

6.2. *Discussion of Obtained Results*

The results of the experiments described above are shown in tables 1 to 3. Each table presents the results obtained for one test instance. DGPBAA refers to the implementation described in pseudocode 1 (with the diversity control mechanisms) and PBAA refers to the previous version (without the diversity control mechanisms). The values in the columns I, V are computed for the set P_{ND}. For the values in the column $C(A, B)$, A represents the non-dominated set obtained by DGPBAA and B represents the non-dominated set obtained by the PBAA.

It can be observed that the use of the diversity control mechanisms helps to improve the performance of the search algorithm. For example, for the **nott1** instance we can see in table 1 that in each of the 10 runs the non-dominated set obtained by DGPBAA is better than the non-dominated set obtained by PBAA. That is, the approximation sets obtained when the diversity measures are used to guide the search have higher diversity in the objective space (I), higher diversity in the solution space (V), more non-dominated solutions (size) and also compares (slightly) better when using the C metric. Similar observations can be made for the test problems **nott1b** and **trent1** in tables 2 and 3 respectively. It is important to highlight that in each single run, the diversity in the solution space of the non-dominated set obtained when using the diversity control mechanisms is greater than V_{good}. On the contrary, when the mechanisms to control diversity are not used, the diversity in the solutions space of the obtained non-dominated set is below V_{good}, except for a few runs in the test instance **trent1** as shown in table 3. When using the C metric it is not clear whether the DGPBAA implementation finds better non-dominated sets. However, we should emphasize that the main contribution of the implemented mechanisms appears to be that they help the algorithm to maintain diversity in both the objective and the solution space. This is precisely the aim in the space allocation problem tackled here, to provide a set of non-dominated solutions that better satisfies the requirements of the space administrators.

Table 1. Results for the test instance **nott1**.

	DGPBAA				PBAA			
run	I	V	size	$C(A,B)$	I	V	size	$C(B,A)$
1	4.70	76.3	23	0.71	3.45	61.6	21	0.46
2	4.95	74.7	28	0.63	3.83	61.4	16	0.37
3	4.56	79.3	25	0.69	3.39	56.2	18	0.42
4	4.87	81.6	24	0.57	3.47	49.4	15	0.47
5	4.91	76.1	27	0.60	3.76	62.1	20	0.47
6	4.52	75.9	29	0.73	3.51	56.3	18	0.37
7	4.59	73.6	28	0.64	3.36	59.2	15	0.39
8	5.03	77.4	22	0.62	3.28	53.7	19	0.41
9	4.86	80.2	26	0.66	3.52	58.3	17	0.49
10	4.77	81.6	25	0.64	3.41	52.5	19	0.43
offline	6.43	73.2	37	0.62	5.12	47.4	24	0.37

Table 2. Results for the test instance **nott1b**.

	DGPBAA				PBAA			
run	I	V	size	$C(A,B)$	I	V	size	$C(B,A)$
1	4.31	72.5	21	0.59	3.13	65.2	20	0.46
2	4.48	74.2	18	0.61	3.51	61.6	15	0.51
3	4.87	75.6	19	0.57	3.04	62.7	18	0.48
4	4.22	71.8	22	0.46	3.76	60.5	16	0.48
5	4.95	74.3	17	0.62	3.28	58.4	19	0.56
6	5.04	75.1	24	0.59	3.16	57.3	18	0.49
7	4.69	73.5	18	0.63	2.94	61.3	17	0.44
8	4.27	71.6	19	0.71	3.45	55.7	21	0.31
9	4.63	74.8	22	0.66	3.31	57.3	14	0.48
10	4.91	73.5	21	0.57	3.34	61.6	20	0.41
offline	5.63	67.2	34	0.72	4.12	41.4	21	0.38

Table 3. Results for the test instance **trent1**.

	DGPBAA				PBAA			
run	I	V	size	$C(A,B)$	I	V	size	$C(B,A)$
1	5.45	82.6	25	0.64	4.02	61.2	21	0.56
2	5.51	75.4	23	0.53	4.62	63.6	16	0.48
3	5.34	80.2	27	0.48	3.56	71.2	18	0.37
4	5.16	77.5	22	0.51	4.23	64.9	16	0.41
5	5.46	74.9	25	0.47	4.56	69.5	14	0.39
6	5.62	79.4	29	0.40	4.18	62.7	22	0.36
7	5.39	81.0	31	0.59	4.39	64.6	23	0.46
8	5.26	75.8	24	0.57	4.40	61.1	19	0.51
9	5.11	79.4	21	0.46	4.04	73.7	16	0.39
10	5.74	82.6	25	0.49	3.87	64.2	20	0.35
offline	6.76	72.4	43	0.68	5.12	52.4	28	0.37

7. Summary

In this paper we have shown that diversity measures can be used to guide the search in multi-objective optimization in order to achieve sets of non-dominated solutions that better satisfy the requirements of the decision-makers. We carried out experiments for a real-world problem with two objectives, the problem of space allocation in academic institutions. In this problem, the decision-makers are interested in obtaining a good approximation set that is also diverse with respect to the solution space. We used the moment of inertia to measure diversity in the objective space and a problem-specific indicator to measure diversity in the solution space. The algorithm used in our experiments is a population-based approach in which each individual in the population in improved by local search and a specialized mutation operator is used to disturb a solution in a controlled fashion. Two diversity control mechanisms were incorporated to the algorithm, one based on diversity in the objective space and another based on diversity in the solution space. In the first mechanism, the diversity in the objective space is used as a helper objective in order to determine if candidate solutions generated by local search are accepted or not. In the second mechanism, the diversity in the solution space is used to alternate between the phases of exploitation and exploration. During exploitation, the algorithm employs local search only. During exploration, the specialized mutation operator is also applied in addition to local search. In order to assess the contribution of the diversity control mechanisms, we carried out experiments on three real-world test instances of the space allocation problem in academic institutions. The results obtained in our experiments show that the algorithm produces better sets of non-dominated solutions when the diversity control mechanisms are used to guide the search. In particular, these non-dominated sets have higher diversity in the solution space which is a common requirement by space administrators.

References

1. Ang K.H., Chong G., Li Y., Preliminary statement on the current progress of multi-objective evolutionary algorithm performance measurement, *Proceedings of the 2002 congress on evolutionary computation (CEC 2002)*, IEEE press, 1139-1144, (2002).
2. Belton V., Stewart T.J., Multiple criteria decision analysis - an integrated approach, Kluwer academic publishers, 2002.
3. Burke E.K., Landa Silva J.D., Improving the performance of multiobjective optimizers by using relaxed dominance, *Proceedings of the 4th asia-pacific conference on simulated evolution and learning (SEAL 2002)*, 203-207, (2002).
4. Burke E.K., Landa Silva J.D., The effect of the fitness evaluation method on the performance of multiobjective search algorithms, *to appear in European Journal of Operational Research*, (2004).
5. Burke E.K., Landa Silva J.D., Soubeiga E., Hyperheuristic approaches for multiobjective optimisation, *Proceedings of the 5th metaheuristics international conference (MIC 2003)*, (2003). Extended version available from the authors.
6. Burke E.K., Varley D.B., Space Allocation: an analysis of higher education requirements. The Practice and theory of automated timetabling II: Selected papers from the 2nd international conference on the practice and theory of automated timetabling (PATAT 97), *Lecture notes in computer science*, **1408**, Springer , 20-33, (1998).
7. Coello Coello C.A., Van Veldhuizen D.A., Lamont G.B., Evolutionary algorithms for solving multi-objective problems, Kluwer academic publishers, (2002).
8. Czyzak P., Jaszkiewicz A., Pareto simulated annealing - a metaheuristic for multiple-objective combinatorial optimization, *Journal of multicriteria decision analysis*, **7(1)**, 34-47, (1998).
9. Dawande M., Kalagnanam J., Keskinocak P., Ravi R., Salman F.S., Approximation algorithms for the multiple knapsack problem with assignment restrictions, *Journal of combinatorial optimization*, **4(2)**, 171-186, (2000).
10. Deb K., Multi-objective optimization using evolutionary algorithms, Wiley, (2001).
11. Farhang-Mehr A., Azarm S., Diversity assessment of pareto optimal solution sets: an entropy approach, *Proceedings of the 2002 congress on evolutionary computation (CEC 2002)*, IEEE press, 723-728, (2002).
12. Farhang-Mehr A., Azarm S., Minimal sets of quality metrics, Proceedings of the 2nd international conference on evolutionary multi-criterion opti-

mization (EMO 2003), *Lecture notes in computer science*, **2632**, Springer, 405-417, (2003).

13. Farina M., Amato P., Fuzzy optimality and evolutionary multiobjective optimization, Proceedings of the 2nd international conference on evolutionary multi-criterion optimization (EMO 2003), *Lecture notes in computer science*, **2362**, Springer, 58-72, (2003).
14. Gandibleux X., Morita H., Katoh N., The supported solutions used as a genetic information in a population heuristics, Proceedings of the 1st international conference on evolutionary multi-criterion optimization (EMO 2001), *Lecture notes in computer science*, **1993**, Springer, 429-442, (2001).
15. Ishibuchi H., Yoshida T., Murata T., Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling, *IEEE transactions on evolutionary computation*, **7(2)**, IEEE press, 204-223, (2003).
16. Jensen M.T., Guiding single-objective optimization using multi-objective methods, Applications of evolutionary computing, Proceedings of the EvoWorkshops 2002, *Lecture notes in computer science*, **2611**, Springer, 268-279, (2003).
17. Jin H., Wong M.L., Adaptive diversity maintenance and convergence guarantee in multiobjective evolutionary algorithms, *Proceedings of the 2003 congress on evolutionary computation (CEC 2003)*, IEEE press, 2498-2505, (2003).
18. Jin Y., Sendhoff B., Connectedness, regularity and the success of local search in evolutionary multi-objective optimization, *Proceedings of the 2003 congress on evolutionary computation (CEC 2003)*, IEEE press, 1910-1917 (2003).
19. Jones D.F., Mirrazavi S.K., Tamiz M., Multiobjective meta-heuristics: an overview of the current state-of-the-art, *European journal of operational research*, **137(1)**, 1-9, (2001).
20. Knowles J.D., Watson R.A., Corne D.W., Reducing local optima in single-objective problems by multi-objectivization. Proceedings of the 1st international conference on evolutionary multi-criterion optimization (EMO 2001), *Lecture notes in computer science*, **1993**, Springer, 269-283, (2001).
21. Kumar R., Rockett P., Improved sampling of the pareto-front in multiobjective genetic optimization by steady-state evolution: a pareto converging genetic algorithm, *Evolutionary computation*, **10(3)**, 283-314, (2002).
22. Knowles J., Corne D.C., Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary computation*, **8(2)**, MIT press, 149-172, (2000).
23. Knowles J.D., Corne D.W., Towards landscape analyses to inform the design of a hybrid local search for the multiobjective quadratic assignment problem, *Soft computing systems: design, management and applications*, IOS Press, 271-279, (2002).
24. Knowles J., Corne D., On metrics for comparing nondominated sets, *Proceedings of the 2002 congress on evolutionary computation (CEC 2002)*, IEEE press, 711-716, (2002).

25. Knowles J.D., Corne D.W., Fleischer M., Bounded archiving using the lebesgue measure, *Proceedings of the 2003 congress on evolutionary computation (CEC 2003)*, IEEE press, 2490-2497, (2003).
26. Landa Silva J.D., Metaheuristics and multiobjective approaches for space allocation. *PhD thesis*, School of computer science and information technology, University of Nottingham, UK, (2003).
27. Laumanns M., Zitzler E., Thiele L., On the effects of archiving, elitism, and density based selection in evolutionary multi-objective optimization, Proceedings of the 1st international conference on evolutionary multi-criterion optimization (EMO 2001), *Lecture notes in computer science*, **1993**, Springer, 181-196, (2001).
28. Laumanns M., Thiele L., Deb K., Zitzler E., Combining convergence and diversity in evolutionary multiobjective optimization, *Evolutionary computation*, **10(3)**, 263-282, (2002).
29. Lu H., Yen G.G., Dynamic population size in multiobjective evolutionary algorithms, *Proceedings of the 2002 congress on evolutionary computation (CEC 2002)*, IEEE press, 1648-1653, (2002).
30. Lu H., Yen G.G., Rank-density based multiobjective genetic algorithm, *Proceedings of the 2002 congress on evolutionary computation (CEC 2002)*, IEEE press, 944-949, (2002).
31. Martello S., Toth P., Knapsack problems - algorithms and computer implementations, Wiley, (1990).
32. Merz P, Freisleben B. Fitness landscape and memetic algorithm design, in: Corne D., Dorigo M., Glover F. (eds.), *New ideas in optimisation*, McGraw Hill, 245-260, (1999).
33. Morrison R.W., De Jong K.A., Measurement of population diversity, Artificial Evolution: Selected Papers of the 5th International Conference on Artificial Evolution (EA 2001), *Lecture notes in computer science*, **2310**, Springer, 31-41, (2001).
34. Mostaghim S., Teich J., The role of ϵ-dominance in multi-objective particle swarm optimization methods, *Proceedings of the 2003 congress on evolutionary computation (CEC 2003)*, IEEE press, 1764-1771, (2003).
35. Murata T., Ishibuchi H., Gen M., Specification of genetic search directions in cellular multi-objective genetic algorithms, Proceedings of the 1st international conference on evolutionary multi-criterion optimization (EMO 2001), *Lecture notes in computer science*, **1993**, Springer, 82-95, (2001).
36. Okabe T., Jin Y., Sendhoff B. A critical survey of performance indices for multi-objective optimisation. *Proceedings of the 2003 congress on evolutionary computation (CEC 2003)*, IEEE press, 862-869, (2003).
37. Purshouse R.C., Fleming P.J., Conflict, harmony, and independence: relationships in evolutionary multi-criterion optimisation, Proceedings of the 2nd international conference on evolutionary multi-criterion optimization (EMO 2003), *Lecture notes in computer science*, **2632**, Springer, 16-30, (2003).
38. Shimodaira H., A diversity control oriented genetic algorithm (DCGA): development and experimental results, *Proceedings of the 1999 genetic and*

evolutionary computation conference (GECCO 1999), Morgan kaufmann, 603-611, 1999.
39. Steuer Ralph E., Multiple criteria optimization: theory, computation and application, Wiley, (1986).
40. Socha K., Kisiel-Dorohinicki M., Agent-based evolutionary multiobjective optimization, *Proceedings of the 2002 congress on evolutionary computation (CEC 2002)*, IEEE press, 109-114, (2002).
41. Talbi E.G., Rahoudal M., Mabed M.H., Dhaenens C., A hybrid evolutionary approach for multicriteria optimization problems: application to the flow shop, Proceedings of the 1st international conference on evolutionary multi-criterion optimization (EMO 2001), *Lecture notes in computer science*, **1993**, Springer, 416-428, (2001).
42. Tan K.C., Lee T.H., Khor E.F., Evolutionary algorithms for multi-objective optimization: performance assessments and comparisons, *Artificial intelligence review*, **17**, 253-290, (2002).
43. Ulungu E.L., Teghem J. Fortemps P.H., Tuyttens D., MOSA method: a tool for solving multiobjective combinatorial optimization problems, *Journal of multicriteria decision analysis*, **8**, 221-236, (1999).
44. Ursem R.K., Diversity-guided evolutionary algorithms, Proceedings of the 7th parallel problem solving from nature (PPSN VII), *Lecture notes in computer science*, 2439, Springer, 462-471, (2002).
45. Van Veldhuizen D.A., Lamont G.B., Multiobjective evolutionary algorithms: analyzing the state-of-the-art, *Evolutionary computation*, **8(2)**, 125-147, (2000).
46. Van Veldhuizen D.A., Lamont G.B., On measuring multiobjective evolutionary algorithms performance, *Proceedings of the 2000 congress on evolutionary computation (CEC 2000)*, IEEE press, 204-211, (2000).
47. Zhu Z.Y., Leung K.S., Asynchronous self-adjustable island genetic algorithm for multi-objective optimization problems, *Proceedings of the 2002 congress on evolutionary computation (CEC 2002)*, IEEE press, 837-842, (2002).
48. Zitzler E., Thiele L., Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach, *IEEE transactions on evolutionary computation*, 3(4), 257-271, (1999).
49. Zitzler E., Deb K., Thiele L. Comparison of multiobjective evolutionary algorithms: empirical results. *Evolutionary computation*, **8(2)**, 173-195, (2000).