IMPROVING THE PERFORMANCE OF TRAJECTORY-BASED MULTIOBJECTIVE OPTIMISERS BY USING RELAXED DOMINANCE

E.K. Burke, J.D. Landa Silva[*]
ekb@cs.nott.ac.uk, jds@cs.nott.ac.uk
Automated Scheduling, Optimisation and Planning Research Group, University of Nottingham, UK.

ABSTRACT

Several recent proposed techniques for multiobjective optimisation use the dominance relation to establish preference among solutions. In this paper, the Pareto archived evolutionary strategy and a population-based annealing algorithm are applied to test instances of a highly constrained combinatorial optimisation problem: academic space allocation. It is shown that the performance of both algorithms is improved by using a relaxed dominance relation and it appears that there is a correlation between this and the existence of constraints in the problem. This paper also discusses why more flexible selection methods may produce better results than the dominance relation in some algorithms and some problem domains.

1. INTRODUCTION

In Pareto optimisation the aim is to find a set of non-dominated solutions that represent a tradeoff among the various conflicting criteria. A number of metaheuristic techniques for Pareto optimisation have been proposed over the years, several of them are extensions of single-objective and single-solution techniques [3,6,8,15]. Recently, the interest for developing multiobjective evolutionary algorithms has increased dramatically [5]. Research in the area has flourished and the number of publications in journals, proceedings (including special sessions and workshops) particularly during the last three years, reflects the growing interest on investigating techniques for effective evolutionary Pareto optimisation, see for example [17]. In addition to developing new approaches, researchers have also reported on extensive experiments for assessing and comparing the performance of Pareto optimisation algorithms [16]. The suitability of evolutionary algorithms for Pareto optimisation has been examined by applying multiobjective evolutionary algorithms to a range of benchmark problems. This has triggered the trend for extending many single-objective methods to create multiobjective variants. On building a much-needed theoretical basis for Pareto optimisation, metrics for assessing the quality of the obtained fronts have also been put forward [12]. It has been noted that since multiobjective evolutionary algorithms have proven to be very successful, it is now interesting to test them in real-world applications including domains such as scheduling and related problems [see 5, page 418].

In this paper we report on our experiments when applying the Pareto archived evolutionary strategy and a population-based annealing algorithm to instances of the space allocation problem. This is a highly constrained combinatorial optimisation problem that can be formulated as a variant of a knapsack problem. These two algorithms are alike in the sense that the evolution of solutions is based solely on self-adaptation with no recombination. The performance of both methods is improved considerably when the dominance relation that measures the attractiveness of candidate solutions is relaxed as proposed by Kokolo et.al. [14]. Moreover, it appears to be a correlation between the above and the existence of constraints in the problem.

Section 2 describes the problem domain and test instances used in our experiments. The two algorithms investigated are outlined in section 3. Section 4 describes the relaxed dominance relation and its use in this paper. Section 5 contains details of our experiments and results while final remarks are presented in section 6.

2. ACADEMIC SPACE ALLOCATION

2.1. Problem Formulation

The space allocation problem in academic institutions refers to the distribution of office space among various resources (staff, postgraduate students, computer rooms, lecture rooms, etc.). Each resource demands a certain amount of space and each room has a limited capacity. There are additional requirements and constraints that restrict the feasibility of solutions. For example, some resources can only be allocated to certain rooms (eg.

[*] JDLS acknowledges support form UACH and PROMEP in México.

lecture rooms where aids are available), or resources may need to be grouped (eg. members of a research group), or resources may need to be adjacent to other resources (eg. secretaries to senior members of staff). Some of those constraints are hard (must be satisfied) while others are soft (desirable to satisfy). The problem is to allocate all resources into the available rooms satisfying all hard constraints and as many soft constraints as possible. In this problem the room space can be wasted or overused but this misuse is penalised. Two objectives can be identified:

1) minimise the misuse of room space and,
2) minimise the violation of soft constraints.

In the real instances of this problem more objectives may exist, for example maximising the functionality of the academic institution, minimising the operation costs, etc. Moreover, constraints can be treated as different objectives since they vary according to the institution and sometimes are conflicting. The problem formulation employs the following notation:

m = number of available rooms.
n = number of resources to allocate.
h = number of hard constraints of the form Z = true.
s = number of soft constraints of the form Z = true.
c_i = capacity of the room i, $i = 1,2,\ldots,m$.
w_j = size of resource j, $j = 1,2,\ldots,n$.
x_{ij} = 1 if resource j is assigned to room i, 0 otherwise.

The aim is to minimise: $F(x) = (F_1(x), F_2(x))$

subject to $\sum_{i=1}^{m} x_{ij} = 1 \quad j = 1,2,\ldots,n$

$Z_k = \text{true} \quad k = 1,2,\ldots,h$

where $F_1(x) = \sum_{i=1}^{m} (WP_i + OP_i)$ and $F_2(x) = \sum_{r=1}^{s} SCP_r$

for each room i, WP_i expresses the penalty if the room capacity is wasted while OP_i expresses the penalty if the capacity is overused. Here, these penalties are given by:

$$WP_i = c_i - \sum_{j=1}^{n} w_j x_{ij} \quad \text{and} \quad OP_i = 2\left(\sum_{j=1}^{n} w_j x_{ij} - c_i\right)$$

SCP_r expresses the penalty due to the violation of the r^{th} soft constraint. The constraints types (soft and hard) and associated penalties are given in the next section. Note that some information regarding proximity between rooms is also needed. In this paper, this is implemented by maintaining (for each room) a list containing those rooms that are adjacent or near to each room.

The problem described above can be seen as a variant of the knapsack problem. Specifically, this is a constrained variant of the bin-packing problem with varying bin capacities. Other constrained variants of knapsack type problems have also been investigated [4,9].

2.2. Test Instances

Test instances of the problem described above have been prepared from real data supplied by three British universities. The three test problems used in this paper are summarised in table 1 below.

	nott1		nott2		trent	
n , m	55 , 55		115 ,115		151 ,73	
constraints	hard	soft	hard	soft	hard	soft
allocated	3	7	12	25	6	13
adjacent	5	4	9	14	--	--
together	--	--	--	--	21	15
not sharing	--	10	10	21	42	103
grouped	1	5	2	5	4	1
total	9	26	33	65	73	132

Constraint	Penalty	Description
allocated	20	resource allocated in a specific room
adjacent	10	resource adjacent to other resource
together	10	resources allocated in same room
not sharing	50	resource not to share a room
grouped	5	resources allocated close each other

Table 1. Characteristics of the test problems used and description of the types of constraints considered in this paper.

3. THE ALGORITHMS

The two algorithms used in the experiments in this paper evolve solutions based on self-adaptation, i.e. the current solution is modified by mutation or local search and no recombination is used. In that sense, algorithms like these are often referred to as trajectory-based methods because the candidate solution is somehow similar to the current one. A description of each algorithm and the justification for using them in this paper are presented next.

3.1. The Pareto Archived Evolutionary Strategy

Several variants of the Pareto Archived Evolutionary Strategy have been proposed but this paper refers to the (1+1)-Pareto archived evolutionary strategy [11]. This algorithm starts with one randomly initialised solution and in each iteration, one candidate solution is generated by means of mutations. An external archive (of limited size) is maintained to collect non-dominated solutions. An adaptive grid that divides the objective space is used to evaluate how crowded the region in which each solution lies is. The candidate solution is discarded if it is dominated by the current solution or any other solution in the external archive. The candidate solution is added to the archive and becomes the current solution if it dominates the current solution. If none of them dominates the other, the decision on which solution becomes the current solution and whether to add or not the candidate solution to the archive is done based on the crowding mechanism.

3.2. Population-Based Annealing Algorithm

The second algorithm is based on the simulated annealing metaheuristic [1]. This algorithm is a population-based annealing method using a common cooling schedule for the whole population. The pseudocode is shown in Fig. 1. Each individual is modified by a local search heuristic H_{LS} that employs three neighbourhood structures and keeps a list of attractive moves and a list of tabu moves. These two lists of moves are shared within the population. Previous experiments with the space allocation problem showed that the efficiency of the local search heuristic is improved when using these lists. An archive is used to maintain a population of non-dominated solutions P_{ND}.

Step 1. Randomly initialise the current population P_C.
Step 2. Copy P_C to the population of best solutions P_B.
Step 3. Initialise P_{ND} with the non-dominated solutions from P_B.
Step 4. Set the acceptance probability, $p \leftarrow 0$, the cooling factor $0<\lambda<1$, the decrement step η (a number of iterations), and the re-heating step φ (a number of iterations).
Step 5. For η iterations, apply the local search heuristic H_{LS} to each individual in P_C.
Step 6. Set $\gamma \leftarrow 1$.
Step 7. For each solution X_C in P_C an its corresponding X_B in P_B.
 Step 7.1. Generate a candidate solution X_C' using H_{LS}.
 Step 7.2. If X_C' dominates X_C, then $X_C \leftarrow X_C'$.
  a) If X_C' dominates X_B, then $X_B \leftarrow X_C'$.
 Step 7.3. If X_C is non-dominated with respect to X_C'.
  a) if $p > 0$ and a random generated number in the normal distribution [0,1] is smaller than p, then make $X_C \leftarrow X_C'$.
  b) if $p = 0$, increment *re-heat iterations* and if (*re-heat iterations mod* φ) = zero, $p \leftarrow 1$.
 Step 7.4. If (iterations *mod* η) = 0, then $p \leftarrow \lambda \cdot p$.
 Step 7.5. If X_C' is non-dominated with respect to P_{ND} then update P_{ND}.
Step 8. Go to Step 9 if no individual has achieved further improvement for η iterations, otherwise go to Step 7.
Step 9. Apply the mutation operator to each individual in P_C.
Step 10. If stopping criterion has not been satisfied, go to Step 7.

Figure 1. The population-based annealing algorithm.

3.3. Implementation

In this paper each solution for the problem formulated in section 2.1 is represented by a vector $x = [\pi_1, \pi_2, \ldots, \pi_n]$ where $\pi_n \in \{1,2,\ldots,m\}$. The infeasibility of solutions in this problem was tackled as follows. In the (1+1)-Pareto archived evolutionary strategy, when a mutated solution is infeasible successive mutations are tried until a feasible solution is generated. This is a very fast operation and it worked well in our experiments. The local search heuristic H_{LS} used in the population-based annealing algorithm also searches until a feasible solution is found. Again, previous work in this problem showed that using various neighbourhood structures works well [2]. Parameters for the population-based annealing algorithm were set as follows: $|P_C| = |P_B| = 20$, $\lambda = 0.8$, $\eta = n$, $\varphi = 10n$. The number of non-dominated solutions in the external archive was limited to 20 in both algorithms.

3.4. Justification

The hybrid metaheuristic described in section 3.2 has been developed as a result of the previous research carried out by the authors on the application of metaheuristics to the space allocation problem [2]. Subsequent experiments showed that the approach was capable of producing good non-dominated fronts in this problem. An interesting observation was that better non-dominated fronts were produced when the relaxed concept of dominance was used instead of the dominance relation (see section 4). For investigating whether this behaviour was due to the algorithm or the problem domain, a well-studied approach had to be implemented.

The (1+1)-Pareto archived evolutionary strategy is a recent technique that is simple to implement, it has been tested across a range of problems and it is considered to be competitive with other modern multiobjective evolutionary algorithms [10]. Multiobjective genetic algorithms have not yet been tested in this problem mainly because previous experience showed that recombination of solutions in this highly constrained problem almost always produces infeasible solutions. Of course that only means that good crossover operators or repairing heuristics would need to be designed and therefore the applicability of such multiobjective evolutionary algorithms to this problem could be considered in the future.

4. DOMINANCE AND α-DOMINANCE

Given the current and the candidate solution(s), any algorithm needs a criterion to assign solution fitness and decide which solution(s) will survive and which ones are to be replaced. Combining all the objectives into a single scalar value is an option for assigning fitness. There are several ways to do this, for example linear weighted aggregation and Tchebycheff functions [5].

When using the dominance relation, a solution x' is preferred over solution x only if x' is at least as good as x in all the objectives and better in at least one of them ($x \preceq x'$). A relaxed form of the dominance relation (called α-dominance) that establishes lower and upper bounds of tradeoffs between the objectives was proposed by Kokolo et.al. [14]. The idea behind α-dominance is that a small detriment in one or perhaps several of the objectives is permitted if an attractive improvement in the other objective(s) is achieved. Note that in some sense, this is similar to establishing preferences among the objectives using weights in an aggregating function (see below).

The common philosophy between α-domination and a simple aggregation of objectives is to allow worsening objective(s) in an attempt to widen the search by accepting not only dominating solutions. This is illustrated in Fig. 2 for a two-objective minimisation problem. Using an aggregated value draws a line that splits the objective space in two regions. Above the line lie the solutions considered worse than *x* and those solutions that are considered to be better are below the line. A line at 45 degrees of inclination is used here for simplicity but different slopes will reflect different preferences. Solutions in B dominate solution *x*. Solutions in B, C and D α-dominate solution *x*. Then in region C for example, β_{uv} represents the maximum detriment permitted in objective *u* given the minimum improvement γ_{vu} in objective *v*. In region D, β_{vu} and γ_{uv} are defined in a similar way.

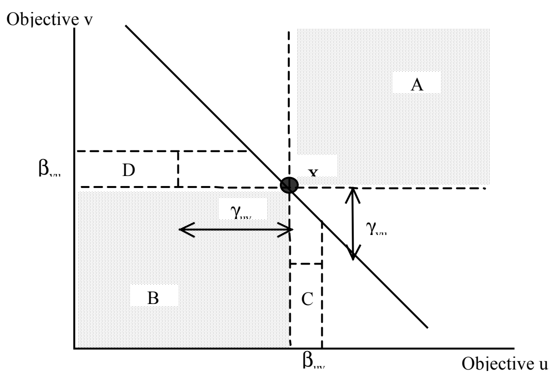

Fig. 2. Aggregating function, dominance and α-dominance.

The different perspectives of “seeing” candidate solutions affects the way in which surviving solutions are selected. An algorithm may find it difficult to discover feasible solutions that dominate the current one(s). This is particularly true in highly constrained combinatorial optimisation problems like the one presented here. Then by accepting α-dominating solutions, it is possible to provide the algorithm with a wider “view” of the potential ways to approach the Pareto optimal front.

In α-dominance, given an optimisation problem with *k* objectives, α_{uv} represents the relation between β_{vu} and γ_{uv} for each pair of objectives $u \neq v$. For example, with respect to Fig. 2 above, α_{uv} expresses the relation between the detriment permitted in the objective *v* and the improvement obtained in the objective *u*. For the formal definition of α-dominance see [14].

5. EXPERIMENTS AND RESULTS

5.1. Experiments

Ten repetitions of the experiments as described next were carried out. Feasible solutions were generated and used as the initial solutions for both algorithms. Each algorithm was executed twice, one run using the standard dominance and one run using the α-dominance. The value α_{uv} = ½ was used for $u \neq v$. The stopping criterion used was a maximum of 10000 candidate solutions visited. Both the offline and online performances of the algorithms were compared. The offline non-dominated sets found by each algorithm when using the α-dominance and standard dominance were collected after 10 repetitions of the above. The online non-dominated sets obtained in each pair of runs with the α-dominance and standard dominance were compared using the following coverage metric introduced by Zitzler et.al. [16]:

$$C(P,T) = \frac{|\{t \in T;\ \exists\ p \in P : p \preceq t\}|}{|T|}$$

where P and T are non-dominated sets. C(P,T) = 1 means that all solutions in T are dominated by at least one solution in P and C(P,T) = 0 means that no solution in T is dominated by a solution in P. Then for each algorithm 10 values of C(dominance,α-dominance) and 10 values of C(α-dominance, dominance) were calculated. These metrics are denoted C(d,α-d) and C(α-d,d) respectively in tables 2 and 3 below.

5.2. Results

For reasons of space only the results for the test instance **trent** are presented here but similar observations were made for the other instances. The offline non-dominated sets found by the algorithms are shown in Fig.3. Table 2 shows the results obtained with respect to the online performance.

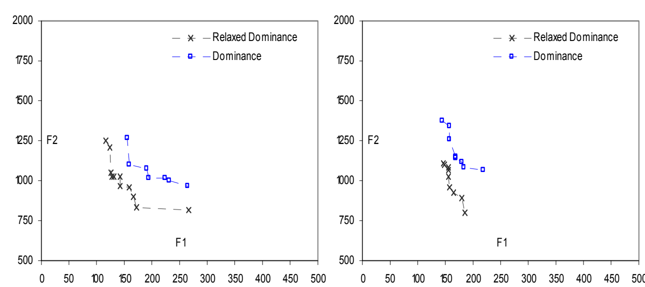

Fig. 3. For test problem **trent**, on the left, offline performance of (1+1)-Pareto archived evolutionary strategy and on the right, offline performance of the population-based annealing algorithm.

Observe that the sets of non-dominated solutions obtained when using the relaxed dominance contain solutions that in general cover the non-dominated sets produced when using the standard dominance relation. The same experiments described above were carried out considering only the soft constraints in the test problems. Of course this eases the restrictions for solutions to be feasible.

Results of the online performance of both algorithms on the trent problem are presented in table 3. Note that the α-dominance does not appear to improve the performance of the algorithms when the hard constraints are not taken into account.

	(1+1)-PAES		PBAA	
	C(d,α-d)	C(α-d,d)	C(d,α-d)	C(α-d,d)
minimum	0	0.82	0	1
average	0.05	0.96	0.01	1
maximum	0.22	1	0.17	1

Table 2. Online performance of the algorithms on problem **trent**.

	(1+1)-PAES		PBAA	
	C(d,α-d)	C(α-d,d)	C(d,α-d)	C(α-d,d)
minimum	0.43	0.21	0.36	0.31
average	0.65	0.24	0.47	0.37
maximum	0.66	0.33	0.53	0.44

Table 3. Online performance of the algorithms when only the soft constraints are considered on problem **trent**.

6. FINAL REMARKS

The use of the dominance relation to establish preference of solutions in multiobjective optimisation deserves attention. According to Knowles et.al., the dominance relation can be beneficial even in single objective optimisation problems [13]. On the other hand, Jaszkiewicz claims that Pareto ranking is not well suited if local search is used [7, page 54]. Kokolo et.al. identified a class of problems that are likely to present serious difficulties to techniques based on dominance selection [14]. The two algorithms implemented here produced better non-dominated fronts when the relaxed dominance was used and this appears to be a consequence of the existence of hard constraints in the problem. Certainly, non-dominated solutions are sought in Pareto optimisation, but under what circumstances (problem domain and algorithms) should the dominance relation be used to identify improvement during the search? When is it more adequate to use the combination of objectives or perhaps a relaxed definition of dominance? The results presented in this paper suggest that it is worthwhile to consider alternative ways for assessing solutions during the search in Pareto optimisation.

7. REFERENCES

[1] Aarts E., Korts J. (Eds.), Simulated Annealing and Boltzman Machines, Wiley, 1998.

[2] Burke E.K., Cowling P., Landa Silva J.D., Petrovic S., Combining Hybrid Metaheuristics and Populations for the Multiobjective Optimisation of Space Allocation Problems, Proceedings of the 2001 Genetic and Evolutionary Computation Conference GECCO 2001, pp. 1252-1259, 2001.

[3] Coello Coello C.A., A Comprehensive Survey of Evolutionary Based Multiobjective Optimization Techniques, Knowledge and Information Systems, Vol. 1, No. 3, pp. 269-308, 1999.

[4] 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, Vol. 4, No. 2, pp.171-186, 2000.

[5] Deb Kalyanmoy, Multi-Objective Optimization Using Evolutionary Algorithms, Wiley, 2001.

[6] Ehrgott M., Gandibleux X., A Survey and Annotated Bibliography of Multiobjective Combinatorial Optimization, OR Spectrum, Vol. 22, No. 4, Springer, pp. 425-460, 2000.

[7] Jaszkiewicz A., Genetic Local Search for Multi-objective Combinatorial Optimization, European Journal of Operational Research, Vol. 137, No. 1, pp.50-71, 2002.

[8] 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, Vol. 137, No. 1, pp. 1-9, 2002.

[9] Kellerer H., Pferschy U., Cardinality Constrained Bin-packing Problems, Annals of Operations Research, Vol. 92, pp. 335-348, 1999.

[10] Knowles J.D., Local-Search and Hybrid Evolutionary Algorithms for Pareto Optimization, PhD thesis, Department of Computer Science, University of Reading, UK, 2001.

[11] Knowles J., Corne D.C., Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy, Evolutionary Computation, Vol. 8, No. 2, pp.149-172, 2000.

[12] Knowles J., Corne D., On Metrics for Comparing Nondominated Sets, Proceedings of the 2002 Congress on Evolutionary Computation CEC2002, Hawaii USA, IEEE Press, pp. 711-716, 2002.

[13] 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, Vol. 1993, pp.269-283, 2001.

[14] Kokolo I., Hajime K., Shigenobu K., Failure of Pareto-based MOEAs: Does Non-dominated Really Mean Near to Optimal?, Proceedings of the CEC2001, Congress on Evolutionary Computation 2001, Seoul, Korea, pp. 957-962, 2001.

[15] Van Valdhuizen D.A., Lamont G.B., Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art, Evolutionary Computation, Vol. 8, No. 2, pp. 125-147, 2000.

[16] Zitzler E., Deb K., Thiele L., Comparison of Multiobjective Evolutionary Algorithms: Empirical Results, Evolutionary Computation, Vol. 8, No. 2, pp.173-195, 2000.

[17] Zitzler E., Deb K., Thiele L., Coello Coello C.A., Corne D., Proceedings of the 1st International Conference on Evolutionary Multi-Criterion Optimization EMO 2001, Lecture Notes in Computer Science, Vol. 1993, Springer, 2001.