MULTIOBJECTIVE GENETIC ALGORITHMS

C. M. Fonseca and P. J. Fleming[1]

Abstract

Multiobjective Genetic Algorithms (MOGAs) are introduced as a modification of the standard genetic algorithm at the selection level. Rank-based fitness assignment and the implementation of sharing in the objective value domain are two of the important aspects of this class of algorithms. The ability of the decision maker (DM) to progressively articulate its preferences while learning about the problem under consideration is one of their most attractive features.

Illustrative results of how the DM can interact with the genetic algorithm are presented. They also show the ability of the MOGA to uniformly sample regions of the trade-off surface.

1 Introduction

There are many real world problems which cannot satisfactorily be characterized by a single performance measure. In the design of an engineering system, for example, factors such as quality, economy and actual performance may have to be compromised. An optimization approach to such problems conventionally requires an expert, the decision maker (DM) to specify how important those measures, or objectives, are relatively to one another, previous to the the optimization process.

The solution to a multiobjective optimization problem, however, is generally not a single point. It consists of a family of points, the Pareto-optimal set, which describes the trade-offs available in the problem. Each point in this set is such that no improvement can be achieved in any one objective without degradation occurring in at least one of the remaining. Therefore, Pareto-optimal points are also called non-dominated, or non-inferior, solutions to the MO problem.

[1] *C.Fonseca@shef.ac.uk, P.Fleming@shef.ac.uk*
University of Sheffield, Department of Automatic Control and Systems Engineering,
P.O. Box 600, Mappin Street, Sheffield, S1 4DU, U.K.

Non-inferior solutions have been obtained one at a time by solving appropriately formulated non-linear programming problems. Methods such as the weighted sum approach, the ε-constraint method and goal programming have been used, of which the goal attainment method, a particular case of goal programming, has shown to be particularly useful in Computer Aided Control System Design (CACSD) [1, 2, 3]. Still, an *a priori* articulation of preferences, made through the precise expression of usually not well known weights and/or priorities, obscures the interplay between objectives and denies the DM a better understanding of the problem.

Genetic algorithms maintain a population of solutions and, therefore, can search for many non-inferior solutions in parallel. Their ability to produce a set of solutions in a single run, without recourse to strong domain-specific assumptions and heuristics, confers an immediate benefit over conventional MO methods. Earlier work on genetic algorithms for multiobjective optimization has been carried out by Schaffer [4], Wienke *et al.* [5], Hajela and Lin [6]. Kursawe [7] used evolution strategies.

2 Multiobjective selection approaches

The main difference between a conventional GA and a MOGA resides in the assignment of fitness. Once fitness has been assigned to individuals, selection can be performed and genetic operators applied as usual.

Methods that rely on the direct combination of performance measures can easily be combined with a GA. However, the GA can be used to address broader formulations where objectives are kept separate during the optimization process.

2.1 Vector Evaluated Genetic Algorithms

In the Vector Evaluated GA (VEGA) approach [4], the population is divided into as many subpopula-

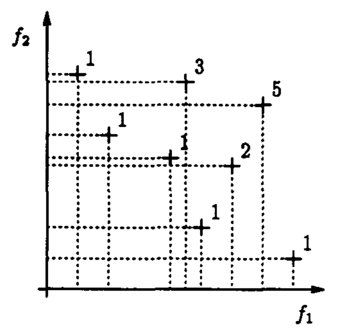

Figure 1: Multiobjective ranking

tions as there are objectives. Individuals in each subpopulation are selected according to their performance, measured in terms of the corresponding objective function. Finally, all subpopulations are shuffled together and the algorithm continues with the application of the genetic operators.

The VEGA approach is in fact equivalent to linearly combining all the objectives in a single fitness measure and, therefore, cannot be expected to perform well in the case of a concave trade-off surface. The fact that individuals tend to split into different species, each of them particularly strong in one of the objectives, was called *speciation* and is due to the proportional selection approach.

2.2 Multiobjective ranking according to dominance

The use of the concept of dominance in a ranking approach to multiobjective selection was proposed by Goldberg [8, p. 201]. Ranking completely ignores how objectives are scaled and can guarantee that, at each selection step, all currently non-dominated individuals are given the same preference.

A suitable MO ranking scheme is described in [9] and illustrated in Figure 1. Individuals are assigned a rank which corresponds to how many individuals in the current population dominate them. In this way, non-dominated individuals are always assigned the same rank, independently of the shape of the trade-off surface.

Individual fitness can be derived in the following way:

1. Sort population according to rank.

2. Assign fitnesses to individuals by interpolating from the best to the worst in the usual way, according to some function, usually linear but not necessarily.

3. Average the fitnesses of individuals with the same rank, so that all of them will be sampled at the same rate. This procedure keeps the global population fitness constant while maintaining appropriate selective pressure, as defined by the function used.

3 Niche-formation methods for MOGAs

When used on a multimodal function, the genetic algorithm is expected to converge to its global optimum. In the case where there are two or more equivalent optima, the GA is known to drift towards one of them in a long term perspective. This phenomenon of *genetic drift* has been well observed in nature and is due to the populations being finite. It becomes more and more important as the populations get smaller.

Niche formation methods such as sharing [10, 11] prevent genetic drift by penalizing individuals which are too close to one another. The population then tends to distribute itself around the existing optima, balancing performance with relative distance, and forming stable sub-populations, or *niches*.

Sharing has conventionally been applied in the decision variable and in the genotypic domains. In the case of multiobjective optimization, it can also be applied in the objective value domain, in order to promote the uniform sampling of the trade-off surface.

Due to the definition of non-dominance, an upper bound for the hyper-area of a bounded trade-of surface can be calculated, which enables one to sensibly estimate the sharing parameter σ_{share}. Mating restriction can also be implemented, by defining the corresponding parameter σ_{mating}.

4 Progressive articulation of preferences with the MOGA

A simple Pareto-based ranking scheme, as described earlier, would make the GA try to evolve a discretized version of the whole trade-off surface of a given problem. However, the GA works with a finite population and the full solution to the MO problem may simply be too large to be sampled accurately by a reduced number of individuals.

Since the designer is usually looking for a single compromise solution to the MO problem, a very precise knowledge of areas of the trade-off surface which, according to some higher level knowledge, express bad

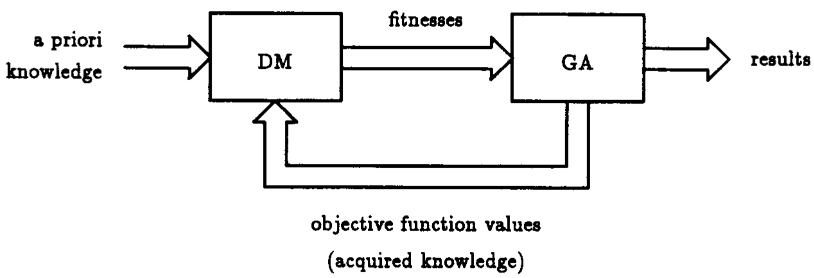

Figure 2: A General Multiobjective Genetic Optimizer

compromises is of no utility. When such higher-level information exists and can be expressed, for example in terms of goals to be attained, it should be used to influence selection.

In this case, the *utility* (or fitness, in GA terminology) of a candidate solution is no longer necessarily constant across the whole Pareto set. It is now an, often subjective, value which depends on the DM. When nothing is known about what a good compromise is, the concept of Pareto optimality allows for an objective rating of individual performance. As different points are evaluated, the DM acquires knowledge about the trade-offs present in the problem and is in a better position to refine its requirements. The interaction between the DM and the GA is illustrated in Figure 2.

4.1 Goal-directed multiobjective ranking

The multiobjective ranking scheme presented earlier can be extended to take goal information into account, by including this information when comparing pairs of individuals. The main idea is that degradation in those objectives which satisfy their goals is acceptable provided it results in real improvement in the remaining objectives and it does not go beyond the goal values.

This concept can be implemented by modifying the relational operator used to compare individuals, as proposed in [9]. Individual ranks are still computed as one plus the number of individuals in the current population which are preferable to the individual under consideration.

This approach allows the DM to direct the search to the compromise region of interest without imposing any restrictions on the search space. Decisions are made solely at the objective value level and only reflected in the decision variable domain via the process of evolution.

5 Some experimental results

The MOGA, as briefly presented here, is currently being applied to the step response optimization of a Pegasus gas turbine engine. A full non-linear model of the engine [12], implemented in SIMULINK [13], is used to simulate the system, given a number of initial conditions and the controller parameter settings. The GA is implemented in MATLAB [14, 15], which means that all the code actually runs in the same computation environment.

The logarithm of each controller parameter was Gray encoded as a 14-bit string, leading to 70-bit long chromosomes. A random initial population of size 80 and standard two-point reduced surrogate crossover and binary mutation were used. The initial goal values were set according to a number of performance requirements for the engine. Four objectives were used:

- t_r The time taken to reach 70% of the final output change. Goal: $t_r \leq 0.59$s.
- t_s The time taken to settle within $\pm 10\%$ of the final output change. Goal: $t_s \leq 1.08$s.
- os Overshoot, measured relatively to the final output change. Goal: $os \leq 10\%$.
- err A measure of the output error 4 seconds after the step, relative to the final output change. Goal: $err \leq 10\%$.

A typical trade-off graph, obtained after 40 generations with the initial goals, is presented in Figure 3 and represents the accumulated set of satisfactory non-dominated points. At this stage, the setting of a much tighter goal for the output error ($err \leq 0.1\%$) reveals the graph in Figure 4, which contains a subset of the points in Figure 3. Continuing to run the GA, more definition can be obtained in this area (Figure 5). Figure 6 presents an alternative view of these solutions, illustrating the arising step responses.

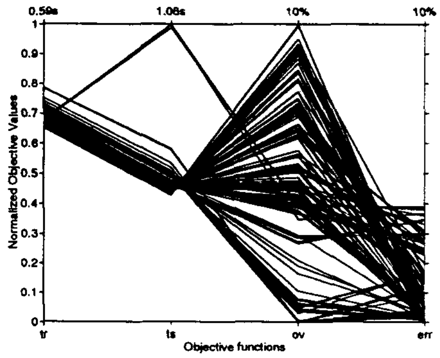

Figure 3: Trade-off Graph for the Pegasus Gas Turbine Engine after 40 Generations (Initial Goals)

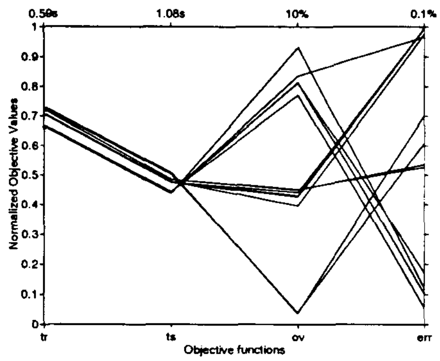

Figure 4: Trade-off Graph for the Pegasus Gas Turbine Engine after 40 Generations (New Goals)

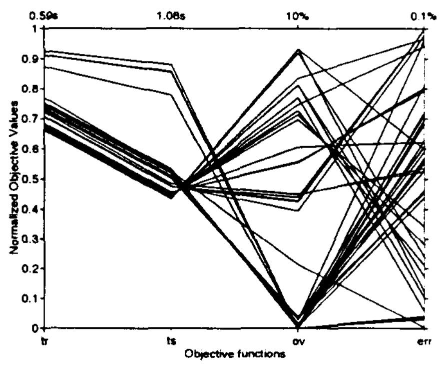

Figure 5: Trade-off Graph for the Pegasus Gas Turbine Engine after 60 Generations (New Goals)

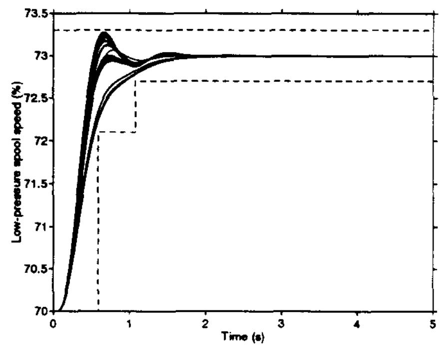

Figure 6: Satisfactory Step Responses after 60 Generations (New Goals)

6 Concluding remarks

Genetic algorithms are a powerful optimization tool which is particularly appropriate to multiobjective optimization. The ability to sample trade-off surfaces in a global, efficient and directed way is very important for the extra knowledge it provides. An illustrative example which shows how the DM can interactively guide the GA by specifying a goal vector was given.

At this stage, the development of a front-end and suitable visualization tools which cater for easy human-machine interaction is necessary. Its development under MATLAB is now even more attractive, given the new graphic capabilities of version 4.

Acknowledgements

The first author gratefully acknowledges support by Programa CIENCIA, Junta Nacional de Investigação Científica e Tecnológica, Portugal.

References

[1] P. J. Fleming, "Computer aided design of regulators using multiobjective optimization," in *Proceedings of the 5th IFAC Workshop on Control Applications of Nonlinear Programming and Optimization*, (Capri), pp. 47–52, Pergamon Press, 1985.

[2] R. Farshadnia, *CACSD using Multi-Objective Optimization*. PhD thesis, University of Wales, Bangor, UK, 1991.

[3] P. J. Fleming, T. P. Crummey, and A. J. Chipperfield, "Computer assisted control system design and multiobjective optimization," in *Proceedings of the ISA Conference on Industrial Automation*, (Montreal, Canada), pp. 7.23–7.26, 1992.

[4] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Genetic Algorithms and Their Applications: Proceedings of the First International Conference on Genetic Algorithms* (J. J. Grefenstette, ed.), pp. 93–100, Lawrence Erlbaum, 1985.

[5] D. Wienke, C. Lucasius, and G. Kateman, "Multicriteria target vector optimization of analytical procedures using a genetic algorithm. Part I. Theory, numerical simulations and application to atomic emission spectroscopy," *Analytica Chimica Acta*, vol. 265, no. 2, pp. 211–225, 1992.

[6] P. Hajela and C.-Y. Lin, "Genetic search strategies in multicriterion optimal design," *Structural Optimization*, vol. 4, pp. 99–107, 1992.

[7] F. Kursawe, "A variant of evolution strategies for vector optimization," in *Parallel Problem Solving from Nature, 1st Workshop, Proceedings* (H.-P. Schwefel and R. Männer, eds.), pp. 193–197, Berlin: Springer-Verlag, 1991.

[8] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, Massachusetts: Addison-Wesley, 1989.

[9] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization," in *Genetic Algorithms: Proceedings of the Fifth International Conference* (S. Forrest, ed.), Morgan Kaufmann, July 1993.

[10] D. E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," in *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms* (J. J. Grefenstette, ed.), pp. 41–49, Lawrence Erlbaum, 1987.

[11] K. Deb and D. E. Goldberg, "An investigation of niche and species formation in genetic function optimization," in *Proceedings of the Third International Conference on Genetic Algorithms* (J. D. Schaffer, ed.), pp. 42–50, Morgan Kaufmann, 1989.

[12] S. D. Hancock, *Gas Turbine Engine Controller Design Using Multi-Objective Optimization Techniques*. PhD thesis, University of Wales, Bangor, UK, 1992.

[13] The MathWorks, Inc., SIMULINK *User's Guide*, Mar. 1992.

[14] The MathWorks, Inc., MATLAB *Reference Guide*, Aug. 1992.

[15] P. J. Fleming, C. M. Fonseca, and T. P. Crummey, "MATLAB: Its toolboxes and open structure," in *CAD for Control Systems* (D. A. Linkens, ed.), ch. 11, pp. 271–286, Marcel-Dekker, 1993.