Automated Solution Selection in Multi-Objective Optimisation

Andrew Lewis and David Ireland

***Abstract*—This paper proposes an approach to the solution of multi-objective optimisation problems that delivers a single, preferred solution. A conventional, population-based, multi-objective optimisation method is used to provide a set of solutions approximating the Pareto front. As the set of solutions evolves, an approximation to the Pareto front is derived using a Kriging method. This approximate surface is traversed using a single objective optimisation method, driven by a simple, aggregated objective function that expresses design preferences. The approach is demonstrated using a combination of multi-objective particle swarm optimisation (MOPSO) and the Simplex method of Nelder and Mead, applied to several, standard, multi-objective test problems. Good, compromise solutions meeting user-defined design preferences are delivered without manual intervention.**

I. Introduction

INCREASINGLY, in the engineering design process, the prototyping stage is being replaced by computer modelling. This is normally cheaper, allows a systematic exploration of a wider range of scenarios and the possibility of optimisation of the design. For example, mechanical engineering design problems might require that components meet functional specifications and are also optimal in the sense of giving maximal fatigue life [14]. Such problems can be solved by computing the fatigue life, involving a finite element analysis of the stress field followed by computation of the growth rate of hypothetical pre-existing cracks under a given load regime.

In most real-world problems, an "optimal" solution involves simultaneously satisfying several objectives. Minimizing cost and risk while maximizing performance, maximizing strength of a component while minimizing its weight, the goals are often conflicting, and what is "optimal" subject to interpretation. Several approaches have been suggested to solve these *multi-objective optimisation* problems.

Drawing on the existing depth of experience and algorithms to solve optimisation problems with a single objective, the design problem can be re-formulated as the problem of optimizing one of the objectives, perhaps that chosen as the most critical, while the remaining objectives are applied as constraints on the design parameters. While this may produce an adequate compromise in some situations, where two or more objectives are defined by the ultimate performance of the model and the range of their possible values initially unknown it can be difficult to formulate. It also becomes awkward to readily find optimal values for several parameters simultaneously.

If all the objectives are to be manipulated simultaneously, three approaches can be identified [3]:

- *a priori* specification of preferences relating to the objectives;
- *a posteriori* selection of a particular solution from a set, derived by the optimisation method, which defines a trade-off surface or Pareto front;
- *progressive* specification of preferences, by interactive dialogue with the optimisation method.

Of these approaches, the first found early, widespread adoption because it again drew on the existing competence in the solution of problems with single objectives. Typically, some algebraic aggregation is made of the multiple objectives, with weights specifying the design preferences, to form an aggregate objective function (AOF) which is then solved using a method suitable for single objective optimisation. The simplest formulation of an AOF is a simple, weighted sum. However, this has been demonstrated to have significant shortcomings and, in fact, an inability to find solutions in regions of the Pareto front that are non-convex. Koski regarded this difficulty to be "well known" [7] and decades of research have been devoted to its amelioration (see, for example, [9], [10], [15].) The authors have experienced this difficulty first-hand when trialling the approach for solution of a problem in the design of dual-band antennas, where "middling" gain performance at two frequencies is preferable to excellent performance at only one, but impossible to find.

The third of the approaches outlined, that of an interactive optimisation method, can be quite effective. It particularly lends itself to problems in which the reduction of objectives to numerical expression is difficult. However, it can be time-consuming and impractical, in the sense of requiring the frequent availability of a Decision Maker to supply preference information over what may be long periods of execution.

Remaining is the method of populating, by some appropriate method, the set of solutions defining the Pareto front. As it places no extraneous demands on the formulation of the multiple objective functions it has been widely adopted as efficient methods for deriving the set of solutions have been developed. It still has the drawback that it does not provide a single, "optimal" solution. Instead it "half solves" the problem, leaving the task of choosing a preferred solution, from what may be a large number of proffered possibilities, to the external Decision Maker. This is not an easy task. Indeed, efforts have been made to simplify it by reducing the number of solutions provided [8]. But if the problem involves a large number of objectives, finding a single solution on what may be a large and complex hypersurface becomes a significant problem in itself: an optimisation problem.

Figure 1 shows the attainment surface obtained from optimisation of a problem in antenna design with three objectives.

Andrew Lewis is with the Institute for Integrated and Intelligent Systems, and David Ireland, the School of Microelectronic Engineering, Griffith University, Queensland, Australia (email: {a.lewis,d.ireland}@griffith.edu.au).

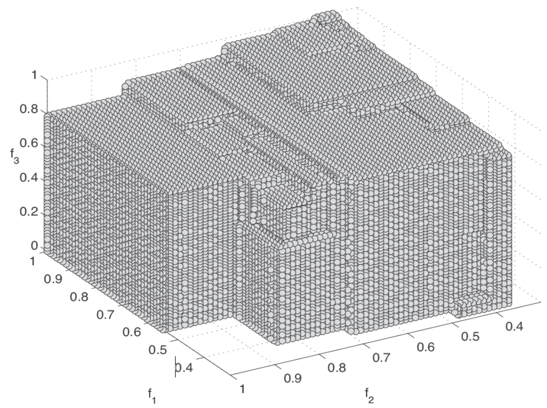

Fig. 1. Attainment surface for three-objective problem in antenna design

This is a relatively simple surface, but demonstrates the complexity of real-world problems compared to the artificially simple test cases often considered. In the problem shown, f_1 is to be minimised, while f_2 and f_3 are maximised. Finding the particular solution corresponding to a desired trade-off, or set of design preferences, is non-trivial.

This paper proposes an approach to the solution of this difficulty, and demonstrates its use on several, standard, multi-objective test problems. In essence, a conventional, population-based, multi-objective optimisation method is used to provide a set of solutions approximating the Pareto front; for the purposes of this experiment a multi-objective particle swarm optimisation (MOPSO) method [2] has been used. As the set of solutions evolves, an approximation to the Pareto front is derived using a Kriging method [5]. This approximate surface is then traversed using a single objective optimisation method, in this case the Simplex method of Nelder and Mead [13], driven by a simple, aggregated objective function (AOF), and finally a single, preferred solution is found.

II. Approximating the Pareto Front

The first step is to provide an approximation to the Pareto front. Population-based methods have proved very successful at this task, and various forms have been developed. Their operation is generally to sample the objective functions and derive from the population a non-dominated set of Pareto-optimal solutions which are then stored in an archive. Where the number of objectives is M, this set of discrete points define an $(M-1)$-dimensional hypersurface, the attainment surface, that approximates the Pareto front. The various methods then use different means to progressively improve the set stored in the archive.

For the MOPSO algorithm used, solution formulation begins with the definition of a population, or swarm, of decision vectors, denoted P^t where t represents the generation. Each i particle in the swarm has a position and velocity defined in parameter space at time t as $\mathbf{x} = [x_1, x_2, ..., x_n]^T$ and $\mathbf{v} = [v_1, v_2, ..., v_n]^T$ respectively. After each generation these vectors are routinely updated using,

$$\mathbf{v}_i^{t+1} = w\mathbf{v}_i^t + c_1 R_1(\mathbf{p}_i^t - \mathbf{x}_i^t) + c_2 R_2(\mathbf{p}_g^t - \mathbf{x}_i^t) \quad (1)$$

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^t \quad (2)$$

where w is generally a positive, constant, inertial term to dampen the effects of the following accelerative terms, c_1 and c_2 are positive weighting coefficients and R_1 and R_2 are random numbers $\in [0,1]$. In common with most population-based methods, the PSO relies on an archive-like database of pareto-optimal solutions which is updated after each generation in order to maintain a pure set of non-dominated solutions. From this archive the $\mathbf{p}_g^t$ particle is chosen while $\mathbf{p}_i^t$ is generally some solution taken from the memory of the i-th particle. Both positions contribute to determining the direction and velocity of the particular particle. For a more detailed explanation of the MOPSO algorithm the reader is referred to [11], [12].

The discrete points stored in the archive can provide only a sampling of the current attainment surface. In the next step, a single-objective optimisation method is used to traverse this surface, and it may require objective function values for points intermediate to those stored in the archive in the course of its search. In order to reduce the computational overhead of the use of the second optimisation method, surrogate approximations are used for these interpolated points.

A. Interpolation of the Pareto Front

This method is not intended to be two, sequential steps, as that would increase the time required to provide a solution. Instead, the search for a single solution satisfying design preferences is intended to proceed *during* execution of the MOPSO algorithm. The passage of time allows a multi-objective optimisation algorithm to build a progressively fuller and more evenly distributed set of Pareto-optimal solutions. As the set evolves, the search for the final, single solution will also continue to refine its trial solution.

However, during execution the set available for use may be incomplete or patchy, necessitating some form of interpolation. Interpolation of unknown regions of the Pareto-optimal hypersurface can be done using a variety of methods. Most commonly encountered are low-ordered polynomial equations, Radial Basis Neural Networks and Kriging methods (see, for example, [16]). In this instance a Kriging formulation was chosen. it was considered the most suitable method for approximating deterministic sampled points as a Kriging output will provide an exact interpolation at known points and a smooth response through unknown regions. Mathematically expressed, Kriging postulates a global model $f(\mathbf{x})$ superimposed by a realised, spatially-correlated, stochastic process $Z(\mathbf{x})$ with zero mean and a variance, s^2, given by,

$$y(\mathbf{x}) = f(\mathbf{x}) + Z(\mathbf{x}) \quad (3)$$

Intuitively, $f(\mathbf{x})$ approximates the parameter space while $Z(\mathbf{x})$ creates local deviations smoothing the interpolated data across unknown regions of the Pareto-Front. In the work presented here a constant global trend was assumed, implying $f(\mathbf{x}) = \beta$. This was estimated, along with the correlated stochastic process, using a maximum likelihood heuristic. For a full description and derivation of the Kriging formulation the reader is referred to [5].

III. Searching the Pareto Front

Once some set of approximately Pareto-optimal solutions and a method for interpolation of unknown values is available, it is possible to traverse the generated hypersurface seeking a solution that satisfies user-specified design preferences. It is proposed this be performed using a secondary optimisation search with a single objective. This single objective should provide some expression of the user-specified design preferences. A simple, weighted sum was considered sufficient, contrary to the objections of Wilson et al. [16], since it need only serve to express the design preferences. It is not called upon to find all points necessary to define the Pareto front, as that task is handled by the MOPSO algorithm. Therefore, once the archive had been populated, and an approximation to the Pareto-Front derived using the Kriging model, a single-objective optimisation algorithm, the Simplex method, was used to find the minimum of an (M-1)-Dimensional space transformed using the augmented Tchebycheff function:

$$F = max(\omega_1 f_1(\mathbf{x}), ..., \omega_M f_M(\mathbf{x})) + \rho \sum_{i=1}^{M} \omega_i f_i(\mathbf{x}) \tag{4}$$

where ω_i is the user defined scalar weight of the i-th objective value, M was the number of objectives and ρ is a positive constant, here set to 0.05. This function was chosen as the non-linear component allows the traversing of non-convex regions of the Pareto-Front [6]. The initial starting point of the Simplex was chosen as the best known minimum of F among the archive points. The individual objective functions were normalised with respect to the maximum objective function values found in the archive to prevent biassing of the objectives.

When a stationary point was identified by the Simplex optimisation algorithm, its location in objective space was verified by evaluation of the original, multiple objective function values using the coordinates of the point in parameter space. The operation of the Simplex algorithm implicitly assumes piecewise linearity and continuity of the Pareto front, and a piecewise linear mapping between the hypersurface of the Pareto front in objective space and the corresponding smooth surface through the solution locations in parameter space.

IV. Test Functions

Three well known test functions were used in this work, taken from [17]. All had the form:

$$f_1(\mathbf{x}) = x_1$$
$$f_2(\mathbf{x}) = g(\mathbf{x}) \cdot h(f_1, g)$$

where the choice of the h and g functions dictate the shape of the Pareto-Front. The dimension of parameter space, N, was 5 for all test functions.

A. Convex

A Convex shaped Pareto front was derived when

$$g(\mathbf{x}) = 1 + \frac{9}{N-1} \sum_{n=2}^{N} x_n \tag{5}$$

$$h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}} \tag{6}$$

B. Non-Convex

Conversely, a non-convex shaped Pareto Front was derived when

$$g(\mathbf{x}) = 1 + \frac{9}{N-1} \sum_{n=2}^{N} x_n \tag{7}$$

$$h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^2 \tag{8}$$

C. Discontinuity

Discontinuity in objective space, but not in parameter space, was created using

$$g(\mathbf{x}) = 1 + \frac{9}{N-1} \sum_{n=2}^{N} x_n \tag{9}$$

$$h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}} - \frac{f_1}{g} sin(10\pi f_1) \tag{10}$$

V. Computational Results

Following are graphical representations of the results of computational experiments using each of the test functions. Figures 2, 6 and 10 show the attainment surfaces, the experimentally achieved approximations to the Pareto fronts, for convex, non-convex and discontinuous test functions, respectively. Following each of these figures are three contour maps, one for each, individually-weighted AOF, showing the AOF on a two-dimensional slice through the parameter space. The contour maps are marked with a "•" at the location of the preferred solution found by the single-objective optimisation pass, demonstrating the degree to which this search was able to find a minimum of the weighted AOF.

An alternative to the proposed hybrid optimisation method for *a priori* specification of design preferences, as outlined in the introduction, is to use a single-objective optimisation method directly with the specified AOF. As noted, this has been found to experience difficulties but, as a point of comparison, a single objective particle swarm optimisation (PSO) algorithm was run for each of the test cases, using equal preference weights, i.e. $\omega_1 = \omega_2 = 0.5$. The PSO algorithm used 10 particles, was allowed 100 iterations, and five different runs were performed using different random seeds, for each of the test cases. The best result obtained for each test case is marked with a "$\oplus$" and the median result of the five runs with a "$\circ$", in each of Figures 4, 8 and 12. A summary of the comparison between the hybrid and *a priori* results is presented in Table I

TABLE I
COMPARISON OF RESULTS OF HYBRID AND *a priori* METHODS
($\omega_1 = \omega_2 = 0.5$)

	Global Best	Hybrid Best	PSO Best	PSO Median
Convex	0.210	0.261	0.287	0.825
Non-convex	0.340	0.502	0.531	0.652
Discontinuous	0.219	0.341	0.413	0.561

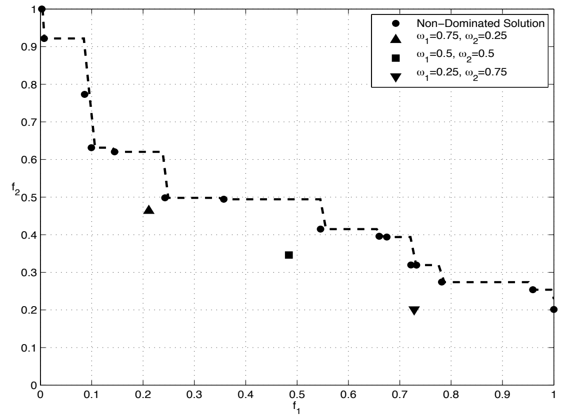

Fig. 2. Attainment surface for the convex test function

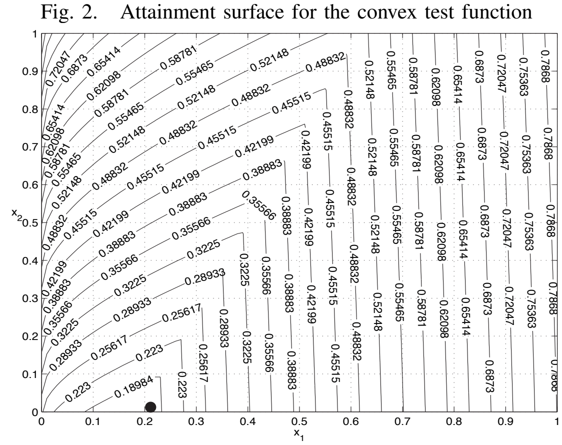

Fig. 3. AOF for convex test function $\omega_1 = 0.75, \omega_2 = 0.25$

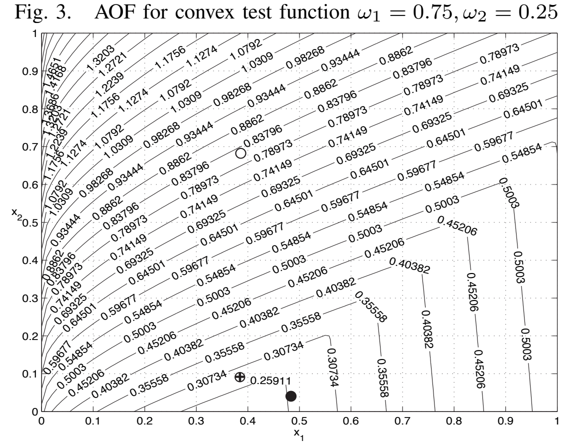

Fig. 4. AOF for convex test function $\omega_1 = 0.5, \omega_2 = 0.5$ (• = hybrid, $\oplus$ = PSO best, $\circ$ = PSO median)

In Figure 2 the final solution points for each AOF can be observed, the equal-weighted solution being near the centre of the front, and the biassed-weight solutions being dislocated to either side. It is interesting to note that, by virtue of the final, validation step, the final solutions found all lie in advance of the approximation to the Pareto front that the (only partially converged) MOPSO search has found. In each of Figures 3, 4 and 5 it can be seen that a near-optimal solution of each corresponding AOF has been found.

In Figures 6, 7, 8 and 9, similar results for the non-convex test function can be seen to those for the convex test function. The solutions found lie closer to the attainment surface, and the quality of the validated solutions is slightly less than for the convex test case. An hypothesis for this behaviour is that the tangent to the concave attainment surface that would form the search direction for the Simplex algorithm is generally directed toward the region of dominated solutions behind the advancing approximation to the Pareto front that the points stored in the archive represent. It would be unlikely for a point predicted by the Simplex search to be significantly better than the current approximations. As the MOPSO algorithm did not proceed to full convergence in these experiments, the resulting solutions found are also observably of lower quality.

In Figure 10 the attainment surface for the discontinuous test function is shown. The final solutions derived from the Simplex search are also indicated. For one AOF, it can be seen that the predicted solution falls well behind the attainment surface, and the sub-optimality of the solution can be confirmed by reference to Figure 13. This particular solution can be seen to be in one of the discontinuous regions of the Pareto front. The Simplex algorithm, with its assumption of continuity of the Pareto front, has predicted a minimum in this region that, when the evaluation has been made to validate it, has been shown to be false. Figure 10 also shows the minimum values of the test function in this region – the "false minimum" can be seen to lie close to this line: a reasonable approximation to the minimum value of the test function, but not a Pareto-optimal solution.

It would be possible to modify the algorithm, for practical application, so that if the value predicted at the endpoint and the actual value differed by more than some reasonable tolerance, the solution presented reverted to the nearest archive member. For purpose of illustration, this point is shown in Figure 13, as the open circle, "$\circ$", showing how this course of action could improve the quality of solution delivered.

When the results using the *a priori* PSO method are inspected, in Figures 4, 8 and 12 and Table I, it can be seen

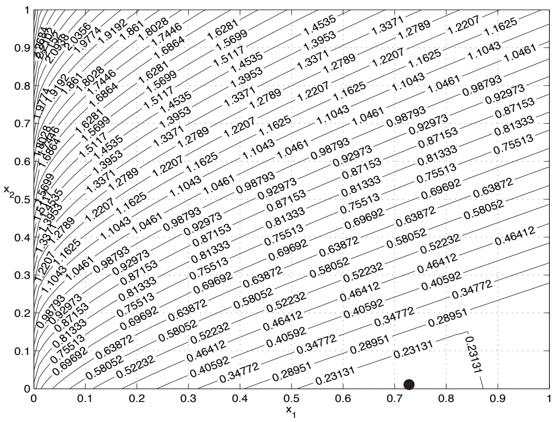

Fig. 5. AOF for convex test function $\omega_1 = 0.25, \omega_2 = 0.75$

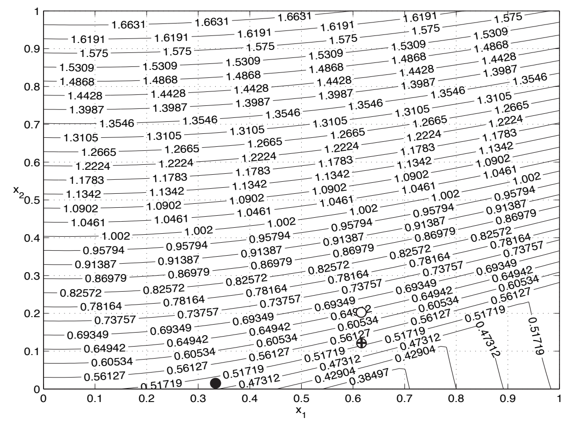

Fig. 8. AOF for non-convex test function $\omega_1 = 0.5, \omega_2 = 0.5$ (• = hybrid, $\oplus$ = PSO best, ∘ = PSO median)

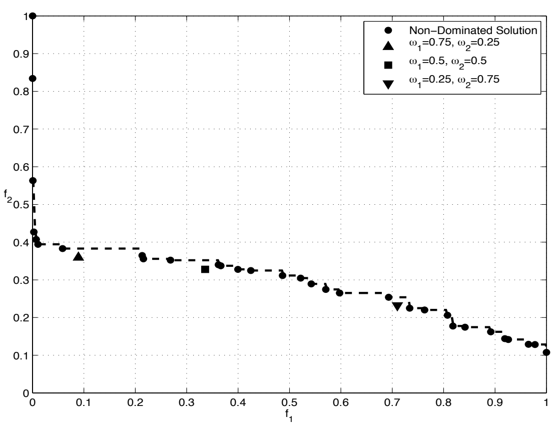

Fig. 6. Attainment surface for the non-convex test function

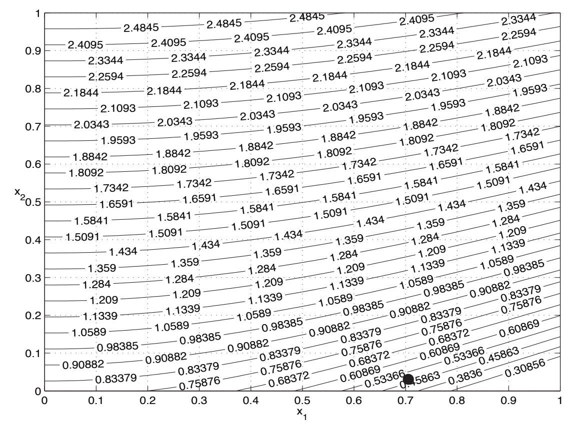

Fig. 9. AOF for non-convex test function $\omega_1 = 0.25, \omega_2 = 0.75$

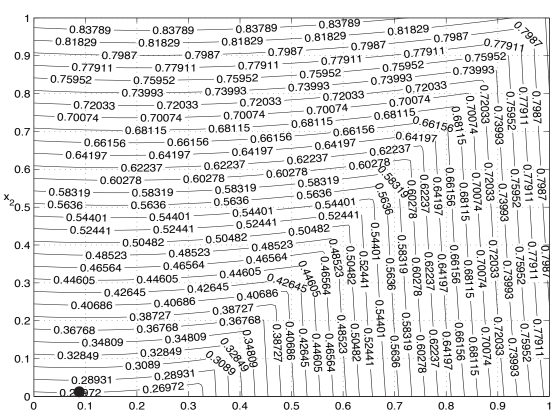

Fig. 7. AOF for non-convex test function $\omega_1 = 0.75, \omega_2 = 0.25$

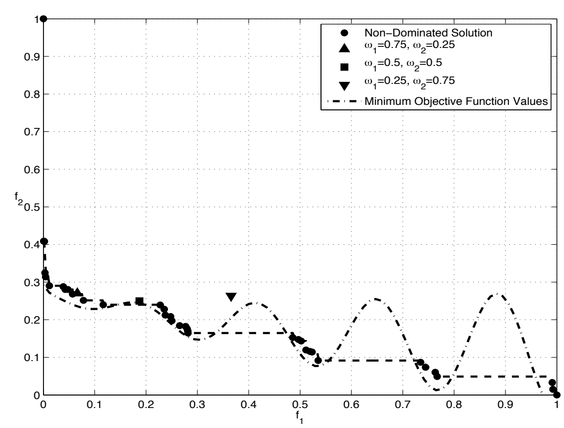

Fig. 10. Attainment surface for the discontinuous test function

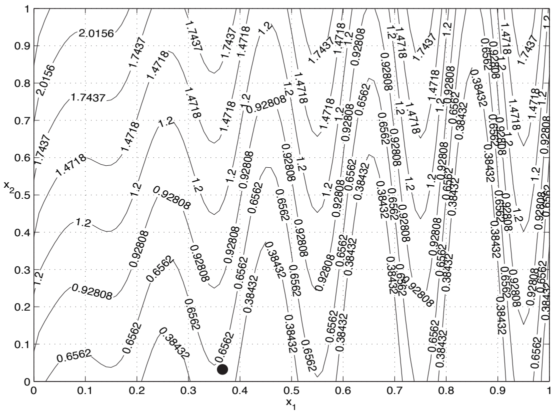

Fig. 11. AOF for discontinuous test function $\omega_1 = 0.75, \omega_2 = 0.25$

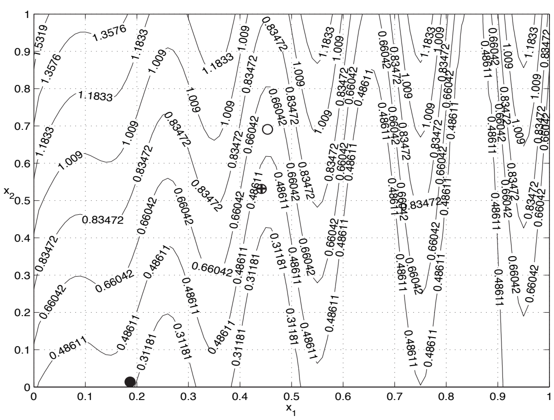

Fig. 12. AOF for discontinuous test function $\omega_1 = 0.5, \omega_2 = 0.5$ ($\bullet$ = hybrid, $\oplus$ = PSO best, $\circ$ = PSO median)

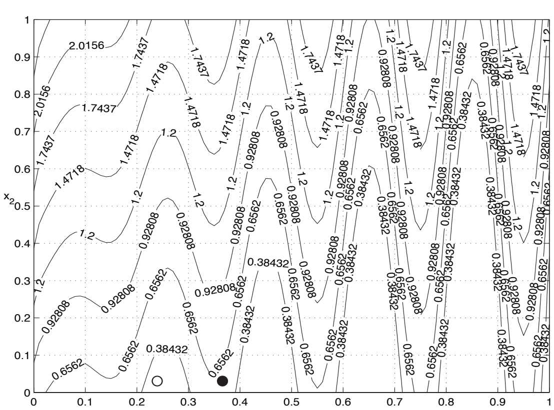

Fig. 13. AOF for discontinuous test function $\omega_1 = 0.25, \omega_2 = 0.75$

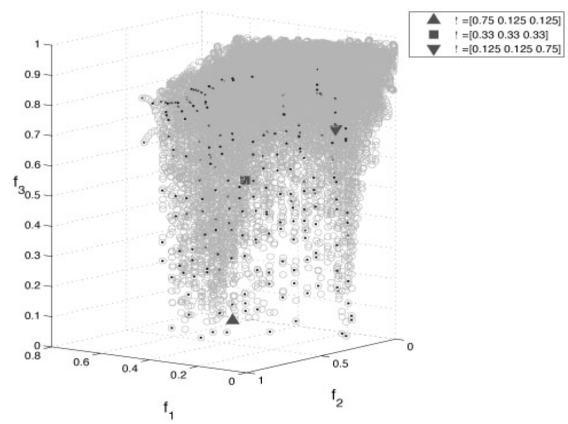

Fig. 14. Location of preferred solutions in an antenna design problem

that the hybrid method has found better solutions in all cases. The *a priori* method achieved a reasonably good result for the convex test case, as might be expected, but a surprisingly large degree of variability, as can be seen from the very poor median value of the five trials. This variability can also be seen for the other test cases. In all cases, the hybrid approach was more robust in providing better results.

VI. Case Study

To confirm the applicability of the method to real-world applications, it was tested on the antenna design problem illustrated in Figure 1. The problem was solved using the MOPSO algorithm, with the "second-phase" search being performed using the Simplex method of Nelder and Mead. Interpolation was again implemented using the same Kriging method. To build the Kriging model, it is necessary to optimise model parameters over a correlation matrix whose size is $O(n)$ where n is the number of points in the data set. The optimisation process involves repeated matrix inversion operations, making this a time-consuming task. For this reason a K-means clustering technique was used to reduce the data set. Figure 14 shows a representation of the attainment surface from Figure 1 with the points of the reduced data set superimposed.

Three different sets of user preferences for each of the three objectives were applied: one in which all objectives were equally weighted, and two others in which objective 1 and objective 2 were preferentially weighted, respectively. The final points found by the "second-phase" search for each set of user preferences are also shown in Figure 14. It may be seen that the equally weighted case has successfully found a compromise solution. The case for which objective 1 was favoured has been displaced toward a low value in that objective, which coincidentally yields a low value for objective 3, but a high value for objective 2. For the case favouring objective 2, it can be seen that a low value was obtained for that objective, at the expense of the other two objectives.

The major computational cost of performing the "second-phase" search lies in construction of the Kriging model. For the reduced data set of 200 points and three models, one for each objective, this task required approximately 3 minutes execution time on the computational cluster used for the experimental tests. Evaluation of the computational electromagnetics simulation that provides objective function values required approximately 10 minutes execution time on the same platform and, since there were sufficient nodes available in the cluster to concurrently evaluate all particles, this was also the time required for a single iteration of the MOPSO algorithm. Since the "second-phase" search is intended to be carried out between iterations of the MOPSO algorithm, i.e. *at the same time as objective function evaluation*, the technique can be applied with little or no additional overhead. At most it may require additional time for one pass after MOPSO has completed. Since the MOPSO algorithm may take tens or hundreds of iterations, even this overhead is negligible.

VII. Conclusion

An approach of hybridising a multi-objective optimisation method and subsequent single-objective search has been proposed as a means to automate the process of solution selection from the set of Pareto-optimal solutions typically delivered. A Kriging method was used to provide surrogate approximations to reduce the computational overhead of the interpolated "second-phase" search. In this way, the search for a single, preferred solution can proceed without requiring *any* additional function evaluations, except for a single, final evaluation to confirm the feasibility of the chosen solution.

Demonstrated using the conventional Multi-Objective Particle Swarm optimisation (MOPSO) algorithm and the Simplex method of Nelder and Mead, and applied to a number of standard test problems, the approach showed an ability to deliver close approximations to user-preferred solutions without manual intervention. The proposed approach was also demonstrated to yield superior results to an alternative, *a priori* aggregation of weighted objectives and use of a single objective optimisation method.

All experiments were performed using standard test cases. Sampling real-world applications of multi-objective optimisation as reported in the literature, on average the maximum number of objectives considered was three. For example, in a review paper across a range of problems in manufacturing [4], a majority of cases considered three objectives, many reduced problems to single objectives using weighted-sum approaches and, of over 20 applications reported, only one considered a maximum of five objectives. Coincidentally, this figure lies near the limit of what may conveniently be graphically represented to allow determination of preferred solutions by visual inspection. It remains an open question whether:

- there are not sufficient real-world problems of interest with more than three or four objectives that cannot adequately be solved by aggregation of objectives,

or

- practitioners do not generally attempt to solve problems where the number of objectives required precludes using visual inspection to determine a preferred solution.

If the latter is true, the method described in this paper offers the potential for a wider range of problems to become tractable.

The method was also demonstrated to yield practical results on a realistic problem in antenna design with insignificant overhead.

Further work will be directed at testing the method with different component algorithms, and application to problems of higher dimensionality in both objectives and parameters. A refined implementation will be integrated in a comprehensive optimisation framework [1] for use as a complete design optimisation methodology.

References

[1] D. Abramson, A. Lewis, and T. Peachey. Nimrod/O: A tool for automatic design optimization. In *Proceedings of the 4th International Conference on Algorithms & Architectures for Parallel Processing (ICA3PP 2000)*, Hong Kong, China, 2000.

[2] C. Coello and M. Lechunga. MOPSO: A proposal for multiple objective particle swarm optimization. In *Computational Intelligence*, Hawaii, 2002. IEEE Press.

[3] Y. Collette and P.Siarry. *Multiobjective optimization*. Springer-Verlag, 2003.

[4] C. Dimopoulos. A Review of Evolutionary Multiobjective Optimization Applications in the Area of Production Research In *Proceedings of the Congress on Evolutionary Computation (CEC 04)*, 2:1487-1494, Oregon, USA, June 2004.

[5] M. Emmerich, K. Giannakoglou, and B. Naujoks. Single- and multi-objective evolutionary optimization assisted by gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4):421–439, August 2006.

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

[7] J. Koski. Defectiveness of weighting method in multicriterion optimization of structures. *Communications in Applied Numerical Methods*, 1:333–337, 1985.

[8] C. Mattson, A. Mullur, and A. Messac. Smart pareto filter: Obtaining a minimal representation of multiobjective design space. *Engineering optimisation*, 36(6):721–740, 2004.

[9] A. Messac, Physical programming: Effective Optimization for Computational Design *AIAA Journal*, 34(1):149-158, 1996.

[10] A. Messac, G. Sundararaj, R. Tappeta, and J. Renaud. Ability of objective functions to generate points on non-convex pareto frontiers. *AIIA Journal*, 38(6):1084–1091, 2000.

[11] J. Moore and R. Chapman. Application of particle swarm to multiobjective optimization. Technical report, Department of Computer Science and Software Engineering, Auburn University, 1999.

[12] S. Mostaghim and J. Teich. Strategies for finding good local guides in multi-objective particle swarm optimization. In *Proceedings of the 2003 Swarm Intelligence Symposium, SIS '03*, pages 26–33, 2003.

[13] J. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

[14] T. Peachey, D. Abramson, A. Lewis, D. Kurniawan, and R. Jones. Optimization using Nimrod/O and its application to robust mechanical design. In *Parallel Processing and Applied Mathematics*, pages 730–737, Czestochowa, Poland, 2003. Springer-Verlag. Lecture Notes in Computer Science, Volume 3019/2004.

[15] M. Scott and E. Antonsson. Compensation and weights for trade-offs in engineering design: Beyond the weighted sum. *Journal of Mechanical Design*, 127(6):1045–1055, 2005.

[16] B. Wilson, D. Cappelleri, T. Simpson, and M. Frecker. Efficient pareto frontier exploration using surrogate approximations. *Optimization and Engineering*, 2:31–50, 2001.

[17] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms on test functions of different difficulty. In *Proceedings of the 1999 Genetic and Evolutionary Computation Conference. Workshop Program*, pages 121–122, Orlando, Florida, 1999.