

# Bias, Proxies and Solution Selection in Multiobjective Optimization

Julia Handl and Joshua Knowles

Manchester Interdisciplinary Biocentre, University of Manchester

**Abstract.** This paper provides a summary and additional observations resulting from a review of the use of multiobjective optimization in the fields of bioinformatics and computational biology [13]. This includes the identification of five distinct ‘contexts’ giving rise to multiple objectives, which explain the reasons behind the use of multiobjective optimization in different application areas and may point the way to potential future uses of the technique. The conclusions drawn are believed to generalize to application areas beyond the biochemical domain.

## 1 Introduction

Numerous problems encountered in bioinformatics and computational biology can be formulated as optimisation problems, and thus lend themselves to the application of powerful heuristic search techniques [4, 21]. Traditionally, the optimisation is conducted with respect to a single goal, but the possibility of optimising multiple objectives simultaneously is rapidly becoming more recognised within this field, as in many others.

## 2 Multiobjective optimisation

Multiobjective optimisation (MOO) concerns optimisation problems with multiple objectives (a.k.a. goals or criteria). Typically, the objective functions may estimate very different aspects of the solutions, aspects that are, therefore, incommensurable and often (partially or wholly) in conflict.

A general (unconstrained) multiobjective optimisation problem (MOP) can be defined as:

$$\begin{aligned} \text{‘minimise’ } \mathbf{z} = \mathbf{f}(\mathbf{x}) &= (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})) & (1) \\ \text{with } \mathbf{x} &= (x_1, x_2, \dots, x_n) \in X, \end{aligned}$$

where  $\mathbf{x}$  is an  $n$ -dimensional decision vector or solution, and  $X$  is the decision space, i.e. the set of all expressible solutions. The vector objective function  $\mathbf{f}(\mathbf{x})$  maps  $X$  into  $\mathbb{R}^m$ , where  $m \geq 2$  is the number of objectives. The vector  $\mathbf{z} = \mathbf{f}(\mathbf{x})$  is an objective vector or point. The image of  $X$  in objective space is the set of all attainable points,  $Z$ .

The term ‘minimise’ appears above in quotation marks because its meaning is not yet defined. Alternative minimisation problems exist, including lexicographic

optimisation (e.g. as used in Olympic games medal tables), minimising the maximum of all the objectives (minmax), and minimising a scalarised combination of the objectives (see [10, 25]). However, by far the most frequent understanding of ‘minimise’, above, is in the sense of Pareto optimality. The Pareto optimal set  $X^*$  of solutions consists of all those that it is impossible to improve in any objective without a simultaneous worsening in some other objective:

$$X^* = \{\mathbf{x}^* \in X \mid \nexists \mathbf{x} \in X, \mathbf{f}(\mathbf{x}) \preceq \mathbf{f}(\mathbf{x}^*)\}, \text{ where} \quad (2)$$

$$\mathbf{f}(\mathbf{x}^1) \preceq \mathbf{f}(\mathbf{x}^2) \text{ iff } \forall i \in 1, \dots, m, f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2) \wedge \exists j \in 1, \dots, m, f_j(\mathbf{x}^1) < f_j(\mathbf{x}^2).$$

The points in objective space corresponding to the Pareto optima are termed nondominated and form the Pareto front.

In most cases, the Pareto optimal set contains more than one element because there exist different trade-off solutions to the problem, which offer different compromises of the objectives. Thus, in practice, solving a MOP often means that a human decision-maker (DM) is involved who then chooses a solution that is Pareto optimal (ideally). Methods of decision-making (before, after or interactively during search) have been extensively investigated in a branch of management science/operations research known as multi-criterion decision making (MCDM) [1, 9] and may include advanced methods of visualisation, e.g. [26].

The *a posteriori* and interactive methods of decision-making can be more effective as the decision-maker may be helped by ‘seeing’ what trade-off solutions are possible. This view has led to a burgeoning of methods for generating the whole Pareto set, or an approximation to it.

Several distinct types of methods for generating good approximations to the Pareto set have been developed, e.g. see [16, 22]. Evolutionary algorithm approaches have become particularly popular and good overviews of these can be found in [3, 5, 7, 12]. For the use of multiobjective optimisation techniques in specific application domains, see e.g. [11, 15, 19].

### 3 Five distinct contexts giving rise to multiple objectives

In [13] we provide a detailed review of the application of multiobjective optimisation in bioinformatics and computational biology. The principal aim of this review is to unravel the different reasons underlying the need for multiobjective optimisation in biological applications. To achieve this, a categorisation is introduced, which is based on the different types of contexts in which multiple objectives may arise or be usefully exploited. This categorisation is recapitulated in the following, and a classification of a range of different biological problems with respect to the five contexts introduced is provided at the end of this paper, in Section 4.

#### 3.1 Standard MOO

As a first category, the ‘standard’ context of multiobjective optimisation is identified, where all objectives are clear, measurable goals that one would genuinely

like to optimise. Assuming all important criteria have been included as objectives, one may be unsure about their relative importance but can be certain that the ‘ideal’ solution will be Pareto optimal. Thus, using an approach that generates a Pareto front (approximation), a decision-maker can learn something about the conflicts between the objectives, the space of possible solutions, and may subsequently select a single preferred solution.

A example of this type of problem setting in biology is the optimisation of biochemical processes where trade-offs exist between aspects of product quality and reaction time or throughput.

### 3.2 Counterbalance for bias

The second category is where MOO is used as a tool to counter-balance a measurement bias affecting an objective function. Such a measurement bias is, for example, encountered in alignment problems, where short alignments can be trivially obtained and the number of mismatches automatically increases with the length of the alignment.

Mathematically, this setting can be described as follows, assuming just one (primary) objective to be optimised.

$$f(\mathbf{x}) = f'(\mathbf{x}) + m(g(\mathbf{x})), \quad (3)$$

where  $f'$  is an ideal (i.e., unknown), unbiased measure of the primary objective,  $m(g(\mathbf{x}))$  is a bias term where  $m$  is an unknown but monotone function of a measurable function  $g$ , and  $f$  is the measurable but biased sum of the two. In the example of alignment problems,  $f$  (the scoring function used) gives a (biased) quality estimate,  $g$  is the length of the given alignment,  $m$  is assumed to be a monotone function and  $f'$  is the ideal (but unknown) quality of the alignment.

Given that  $f'$  is thought to provide an objective assessment of the quality of a solution, it would be desirable to minimise  $f'(\mathbf{x})$  as follows:

$$\text{minimise } f'(\mathbf{x}) = f(\mathbf{x}) - m(g(\mathbf{x})), \quad (4)$$

but since  $m$  is unknown it is not possible to formulate the problem in this way. However, the problem may, instead, be formulated as:

$$\begin{aligned} \text{‘minimise’ } & (f(\mathbf{x}), -(g(\mathbf{x}))), \\ \text{with } & \mathbf{x} = (x_1, x_2, \dots, x_n) \in X, \end{aligned} \quad (5)$$

in terms of two measurable objectives. Hence, the framework of MOO is used as a means of introducing an additional objective,  $g$ , to counter-balance the bias of the primary objective.<sup>1</sup>

The set of Pareto optimal solutions will certainly contain the desired solution since each Pareto optimum is the best value of  $f(x)$ , given a fixed value of  $g(x)$ . In this scenario, selection of the best solution does not usually depend on

<sup>1</sup> N.B. the equations above can be generalised to more than one primary objective, where necessary.

preferences, but on the estimation of the biases. In some applications, the biases may be estimated using random control data, and this may help to identify the best solution in the Pareto front.

Examples of this type of problem in biology include unsupervised feature selection, and sequence and structure alignment problems.

### 3.3 Multiple source integration

In the third category, MOO is used to integrate noisy data from multiple sources. Hence, in this setting, it is used as an alternative to an *a priori* or *a posteriori* integration technique. The problems where this approach is used are originally single-objective. However, multiple noisy views of the data need to be integrated, as their combined use may yield better results than the use of data from a single information source.

Mathematically, this setting can be described by a set of objective functions:

$$\begin{aligned} f_1(\mathbf{x}) &= f'_1(\mathbf{x}) + \bar{n}_1 \\ &\vdots \\ f_m(\mathbf{x}) &= f'_m(\mathbf{x}) + \bar{n}_m, \end{aligned} \tag{6}$$

where the function value of each objective function  $f_i$  is equal to the value of an ideal function  $f'_i$  with some unknown random noise  $\bar{n}_i$  on it, for  $i \in 1, \dots, m$ . In some cases, the  $f'$  are all identical, i.e. the ‘views’ of the data arise from the same types of measurement but e.g. taken at different times. By formulating the problem as,

$$\begin{aligned} \text{‘minimise’ } \mathbf{z} = \mathbf{f}(\mathbf{x}) &= (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})) \\ \text{with } \mathbf{x} &= (x_1, x_2, \dots, x_n) \in X, \end{aligned} \tag{7}$$

and finding the Pareto optima, the impact of the noise may be reduced, if it is reasonably uncorrelated with the solution space  $X$ . Nonetheless, note that it is not guaranteed that the desired solution will be among the Pareto optima.

Examples of this type of problem in biology are the inference of phylogenetic trees and data clustering with several dissimilarity matrices.

### 3.4 Performance approximation by proxies

Category four comprises those applications in which the ‘real’, underlying objective of the problem,  $f'(\mathbf{x}, \mathbf{y})$ , is a function of both the solution  $\mathbf{x}$  and some ‘hidden’ variables  $\mathbf{y}$  that are not available during optimisation. For example, in training a supervised classifier,  $\mathbf{y}$  refers to the generalisation ability of the classifier on future data (which may be estimated using a test set after the optimisation, but the classifier must not be trained using these examples).

Since the function  $f'$  is not suitable for use in the optimisation process (because  $\mathbf{y}$  is unavailable), it needs to be replaced by ‘proxy’ objectives  $f_i(\mathbf{x})$ , which

are functions of  $\mathbf{x}$  only. Often, such ‘proxy’ objectives only capture certain aspects of a good solution, and different proxies are complementary with respect to each other. Thus, it should be expected that the desired solution(s) will score relatively highly under all of the ‘proxy’ objectives, and a MOO approach therefore seems useful, although the desired solution cannot be guaranteed to be among the associated set of Pareto optima.

The difference between this context and that of standard MOO (as introduced above) may seem unclear to some readers. However, the distinction is clear: in the case of standard MOO, the objective functions have primacy, i.e. it is they that define the Pareto set, e.g.: the concept of a ‘best car’ does not exist *per se* but, given a search space, a set of ‘best cars’ is induced by the objectives chosen. In contrast, in the context of proxy objectives, it is the solution that has primacy, and the objectives are only a means of orienting the search in order to discover this solution, e.g.: the real structure of a protein exists, and one may try and find it by employing a number of different energy/cost functions.

Examples of this type of problem in biology include supervised classifier training (as explained above) and protein structure prediction.

### 3.5 Multi-objectivisation

The fifth and final category identified is where MOO may be used solely as a way to obtain improved search ‘guidance’ in what is essentially a single-objective problem.<sup>2</sup> Assuming a single objective that is measurable, a problem may still be difficult because of its search landscape. There are at least two difficulties in search landscapes that can potentially be reduced by ‘multi-objectivisation’: (i) where a problem involves frustration (or epistasis), which causes excessive local optima in the search landscape; (ii) where the search landscape contains regions offering little or no objective function gradient. In the first case, decomposition of the primary objective into several different functions (each function either defined over all of the variables or a subset of them) may help to separate out the conflicting aspects of the problem, thus reducing the number of local optima ‘seen’ by a search algorithm [18]. In the second case, the use of extra ‘helper objectives’ in addition to the primary objective may provide helpful guidance in the flat regions of the landscape [14, 18].

Multi-objectivisation may potentially be achieved by any reformulation of the problem that respects the following relation [18]:

$$\forall \mathbf{x}^{opt} \in X \exists \mathbf{x}^* \in X, x^* = x^{opt}, \quad (8)$$

where  $\mathbf{x}^{opt}$  is an optimal solution to the original single-objective problem and  $\mathbf{x}^*$  is a Pareto optimum of the multi-objectivised problem. This ensures that at least one of the true Pareto optimal solutions will be optimal with respect to the original primary objective and will correspond to the best solution.

An example of this type of problem in biology is structure identification from X-ray powder diffraction data.

<sup>2</sup> NB: there is no reason why multi-objectivisation cannot also be generalised to the case where the original problem is multiobjective.

## 4 Discussion and outlook

Our detailed review paper [13] outlines that multiobjective optimisation has widespread applications in computational biology and bioinformatics. It also shows that the performance gains and flexibility afforded by multiobjective optimisation have been illustrated in a range of initial studies, but that in many of these problem domains the full potential of multiobjective approaches in comparison to the current state-of-the-art techniques remains to be explored.

### 4.1 Classification of problems

The reasons underlying the use of multiobjective optimisation differ widely across these application domains, and this aspect may be more interesting and revealing than a distinction between the specific techniques used. Table 1 summarises a classification of the different application areas with respect to the categorisation introduced in the previous section. Note that several of the problems considered can potentially fall into more than one category, dependent on the specific viewpoint taken. In the table, only those categorisations have been indicated that correspond to views taken in the literature in [13], but this categorisation is clearly not final.

### 4.2 Theoretical considerations

Our review in [13] also highlights that the different applications considered do not only differ in the reasons underlying the use of multiobjective optimisation, but may also differ in the way the final set of solutions is treated. In certain applications, such as instrument optimisation or clustering, the decision-maker will be given the approximation set obtained with the main aim of identifying a single preferred solution. In such applications, it may be possible to obtain additional valuable information from the shape of the Pareto front obtained, but, ultimately, one is interested in obtaining a single solution. In other applications, in contrast, it is the entire set of Pareto optimal solutions that is of interest and that will be selected for future use. Examples of this are the use of the concept of Pareto optimality for gene filtering, or its use for the generation of diverse classifiers for the integration in an ensemble classifier.

These differences are interesting, as they affect the conclusions that can be drawn about the quality of the solutions obtained when comparing the solutions of a single- and a multiobjective optimisation problem based on the same objectives. The principles of Pareto optimality guarantee that the optima of the single-objective problems will also form part of the Pareto front for the corresponding multiobjective problem. Hence, the Pareto set is a superset of the set of optima of the individual single-objective problems, and is therefore guaranteed to contain solutions at least as good as those defined by the single-objective problems. Here, the availability of additional solutions (corresponding to trade-offs between the individual objectives) is usually seen as an advantage of multiobjective optimisation, but is difficult to quantify, as all of the solutions in the Pareto

**Table 1.** Categorisation of the main applications discussed. Some categories, such as “Multi-objectivisation” are currently under-represented, but it is expected that further applications of this type will emerge as the use of multiobjective optimisation propagates in the biological domain. The highest number of entries can currently be observed for the category “Proxy”, which contains all those applications in which a ‘gold standard’ (e.g. the best possible classifier over ‘future’ data, the true structure of a protein, the true network of regulatory relationships or the true evolutionary relationship between sequences/structures) exists, which one would like to reach, but does not have direct access to during the optimisation process.

Problem	Standard MOO	Bias	Multiple data sources	Proxies	Multi-objectivisation
ROC curves	—	—	—	✓	—
Rule mining	—	—	—	✓	—
Accuracy vs. complexity	—	—	—	✓	—
Supervised feature selection	—	—	—	✓	—
Ensemble learning	—	—	—	✓	—
Clustering	—	✓	✓	—	—
Unsupervised feature selection	—	✓	—	—	—
Association rule mining	—	—	—	✓	—
Multidimensional scaling	—	—	✓	✓	—
Phylogenetic trees	—	—	✓	—	—
Gene regulatory networks	—	—	✓	✓	—
Sequence alignment	—	✓	✓	✓	—
Structure alignment	—	✓	✓	✓	—
Structure prediction	—	—	✓	✓	✓
Directed evolution	✓	—	—	—	—
Biochemical systems	✓	—	—	—	—
Biochemical processes	✓	—	—	—	—
Experimental design	✓	—	—	—	—

front (including the solutions to the single-objective problems) are incomparable with respect to each other.

Note, however, that this situation changes when moving towards applications, in which sets of solutions are to be identified. If a set of  $N$  solutions is selected based on single-objective optimisation, this will typically be done by selecting the  $N$  solutions scoring highest under an individual objective (e.g. a t-test in gene filtering). Note that only the best solution out of the resulting set is guaranteed to be Pareto optimal, whereas the solutions ranked 2 to  $N$  are not guaranteed to lie on the Pareto front. In contrast, for a number of solutions  $N$  smaller than the number of Pareto optima  $P$ , a multiobjective approach will allow one to select  $N$  Pareto-optimal solutions. Hence, in a comparison of such sets of solutions, a large fraction of the solutions identified by a single-objective approach would be expected to be dominated by those identified by the multiobjective approach. So, in this setting a multiobjective approach may have an even clearer advantage than when only a single solution is sought.

Evidently, in either case, some assumptions are necessary for these theoretical advantages to hold. Firstly, the use of exact optimisation techniques needs to be assumed, that is, both the single- and the multiobjective optimisation technique used must be guaranteed to find the optimal solutions. Secondly, the ranking returned by the objectives in combination with the principle of Pareto optimality must be assumed to reflect the true quality of the solutions. As mentioned previously, if either noisy or vague ‘proxy’ objectives are used, the best solutions cannot be guaranteed to lie in the Pareto front, and, consequently, no guarantees on the relative performance of the single- and multiobjective optimisation techniques can be given.

Despite these limitations in practice, much of the literature surveyed in [13] has explored the use of heuristic techniques for multiobjective optimisation, including its application in the presence of noise and/or when relying on ‘proxy’ objectives. The results obtained in most of this work have been very encouraging and would seem to indicate that the advantages of multiobjective optimisation do hold in practice.

### 4.3 Evaluation function development

Our review [13] also observes that the computational complexity of multiobjective optimisation may prove prohibitive in some practical settings, such as sequence alignment or protein structure prediction for large molecules. However, despite such limitations regarding its potential application within standard processing or search tools, multiobjective optimisation may prove a valuable tool in the design and development of improved and efficient software for these bioinformatics tasks. In particular, it may allow one to learn about the trade-offs in a problem and to identify recurring patterns in the Pareto fronts, the knowledge and understanding of which may help in the formulation of novel and better single-objective problem formulations.

### 4.4 Visualisation and solution identification

The large majority of MOPs identified in our review have been tackled by generating a whole Pareto front and by then applying (or hoping to apply) some form of decision-making process afterwards to choose a single solution. This strategy defers decision-making until ‘all the information is in’ (a good thing when little is known about the possible trade-offs) but the problem remains how to identify/select a single best solution. The really successful application of this mode of MOO thus calls for advanced methods for the visualisation of the Pareto front and for the support of the decision-maker in selecting solutions from it.

Evidently, straightforward visualisations of the Pareto front are only possible in two or three dimensions, and a representation of the solutions obtained and the relationships between them becomes much more intricate for higher dimensions. To date, only few methods for effective visualisation have been introduced that can deal with the truly multidimensional case (one of the main examples is a parallel axis plot [23]), and visualisation remains a major topic for future research.

Automatic identification of promising solutions from Pareto front approximations has been investigated in several recent works [2, 6, 8, 20, 24]. However, these papers have generally dealt with methods for steering/focusing the search towards the (potentially) more important areas without the need for additional preference information from the decision-maker (usually by searching more strongly in regions of the Pareto front that have highest local curvature). An alternative approach is to first obtain the most complete Pareto front approximation set possible and then to, *a posteriori*, reduce this set to a single solution by some automated process, taking account of the whole Pareto front shape and other information.

More broadly, other approaches to support decision-making in MOO exist, too. Where expert knowledge on how to balance conflicting measures/goals is available, this can be extracted by using preference articulation techniques [9, 1, 17], whereby a series of concrete questions about preferences are asked to the DM. The answers then determine if it is possible to build one or other type of consistent model of the DM's internal utility function; if so, then an automated procedure can potentially be developed for solution evaluation/selection. Note that far more complicated types of model exist than a simple weighted sum over the objectives. These preference articulation techniques are so far used mainly in operations management and have yet to be transferred to biological applications, but there is plenty of potential for future successes in this area.

## 5 Summary

Our detailed review in [13] outlines the wide applicability of multiobjective optimisation in biological problem domains, and illustrates its potential with references to existing results from the literature, where available. Rather than differentiating between differences in the optimisation techniques used, differences in the reasons underlying the attractiveness of multiobjective approaches in different problem domains are emphasised, and these have been recapitulated in the present paper. It is hoped that this viewpoint will help to provide additional insight into the advantages afforded by multiobjective optimisation with regard to the applications listed and/or additional problems encountered in and beyond the field of bioinformatics.

### Acknowledgments

Julia Handl acknowledges support of a doctoral scholarship from the German Academic Exchange Service (DAAD) and the Gottlieb Daimler- and Karl Benz-Foundation, Germany. Joshua Knowles is supported by a David Phillips Fellowship from the Biotechnology and Biological Sciences Research Council (BBSRC), UK.

### References

1. V. Belton and T. J. Stewart. *Multiple Criteria Decision Analysis: an Integrated Approach*. Springer-Verlag, Berlin, Germany, 2002.

2. J. Branke, K. Deb, H. Dierolf, and M. Osswald. Finding knees in multi-objective optimization. In *Proceedings of the Eighth International Conference on Parallel Problem Solving from Nature*, pages 722–731. Springer-Verlag, Berlin, Germany, 2004.
3. C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York, NY, 2002.
4. J. Cohen. Bioinformatics — an introduction for computer scientists. *ACM Computing Surveys*, 36(2):122–158, 2004.
5. D. W. Corne, K. Deb, P. J. Fleming, and J. D. Knowles. The good of the many outweighs the good of the one: evolutionary multi-objective optimization. *Connections. The Newsletter of the IEEE Neural Networks Society*, 1(1):9–13, 2003.
6. I. Das. On characterizing the ‘knee’ of the Pareto curve based on normal-boundary intersection. *Structural Optimization*, 18:107–115, 1999.
7. K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley and Son Ltd, Chichester, UK, 2001.
8. K. Deb. Multi-objective evolutionary algorithms: introducing bias among Pareto-optimal solutions. In *Advances in Evolutionary Computing: Theory and Applications*, pages 263–292. Springer-Verlag, London, UK, 2003.
9. J. S. Dyer, P. C. Fishburn, R. E. Steuer, J. Wallenius, and S. Zionts. Multiple criteria decision making, multiattribute utility theory: the next ten years. *Management Science*, 38(5):645–654, 1992.
10. M. Ehrgott. *Multicriteria Optimization*. Springer-Verlag, Berlin, Germany, 2000.
11. M. Ehrgott and X. Gandibleux. A survey and annotated bibliography of multiobjective combinatorial optimization. *OR Spectrum*, 22(4):425–460, 2000.
12. C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423. Morgan Kaufman Publishers, San Francisco, CA, 1993.
13. J. Handl, J. Knowles, and D. Kell. Multiobjective optimization in computational biology and bioinformatics. *IEEE Transactions on computational biology and bioinformatics*, 2006. In press.
14. M. T. Jensen. Guiding single-objective optimization using multi-objective methods. In *Applications of Evolutionary Computation*, pages 268–279. Springer-Verlag, Berlin, Germany, 2003.
15. Y. Jin, editor. *Multi-Objective Machine Learning*, volume 16 of *Studies in Computational Intelligence*. Springer-Verlag, Berlin, Germany, 2006.
16. D. F. Jones, S. K. Mirrazavi, and M. Tamiz. Multi-objective meta-heuristics: an overview of the current state-of-the-art. *European Journal of Operational Research*, 137(1):1–9, 2002.
17. R. L. Keeney and H. Raiffa. *Decisions with Multiple Objectives: Preferences and Value Trade-Offs*. Cambridge University Press, Cambridge, UK, 1993.
18. J. D. Knowles, R. A. Watson, and D. W. Corne. Reducing local optima in single-objective problems by multi-objectivization. In *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*, pages 269–283. Springer-Verlag, Berlin, Germany, 2001.
19. R. T. Marler and J. S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26(6):369–395, 2004.
20. C. A. Mattson, A. A. Mullur, and A. Messac. Smart Pareto filter: obtaining a minimal representation of multiobjective design space. *Engineering Optimization*, 36:721–740, 2004.

21. S. Mitra. Computational intelligence in bioinformatics. *Transactions on Rough Sets*, pages 134–152, 2005.
22. S. Ruzika and M. M. Wiecek. Approximation methods in multiobjective programming. *Journal of Optimization Theory and Applications*, 126(3):473–501, 2005.
23. K. J. Shaw, A. L. Nortcliffe, M. Thompson, J. Love, P. J. Fleming, and C. M. Fonseca. Assessing the performance of multiobjective genetic algorithms for optimization of a batch process scheduling problem. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 37–45, 1999.
24. G. Stehr, H. Graeb, and K. Antreich. Performance trade-off analysis of analog circuits by normal-boundary intersection. In *40th Design Automation Conference*, pages 958–963. IEEE Press, Anaheim, CA, 2003.
25. R. Steuer. *Multiple Criteria Optimization, Theory, Computation and Applications*. John Wiley and Son Ltd, New York, NY, 1986.
26. H. Wright, K. Brodlie, and T. David. Navigating high-dimensional spaces to support design steering. In *Proceedings of the 11th IEEE Visualization Conference*, pages 291–296. IEEE Press, Anaheim, CA, 2000.