Multiobjective Evolutionary Algorithms

In this chapter we describe the application of evolutionary techniques to a particular class of problems, namely multiobjective optimisation. We begin by introducing this class of problems and the particularly important notion of Pareto optimality. We then look at some of the current state-of-the-art multiobjective EAs (MOEAs) for this class of problems and examine the ways in which they make use of concepts of different evolutionary spaces and techniques for promoting and preserving diversity within the population.

12.1 Multiobjective Optimisation Problems

In the majority of our discussions in previous chapters we have made free use of analogies such as adaptive landscapes under the assumption that the goal of the EA in an optimisation problem is to find a single solution that maximises a fitness value that is directly related to a single underlying measure of quality. We also discussed a number of modifications to EAs that are aimed at preserving diversity so that a set of solutions is maintained; these represent niches of high fitness, but we have still maintained the conceptual link to an adaptive landscape defined via the assignment of a single quality metric (objective) to each of the set of possible solutions.

We now turn our attention to a class of problems that are currently receiving a lot of interest within the optimisation community, and in practical applications. These are the so-called multiobjective problems (MOPs), where the quality of a solution is defined by its performance in relation to several, possibly conflicting, objectives. In practice it turns out that a great many applications that have traditionally been tackled by defining a single objective function (quality function) have at their heart a multiobjective problem that has been transformed into a single-objective function in order to make optimisation tractable.

To give a simple illustration (inspired by [334]), imagine that we have moved to a new city and are in the process of looking for a house to buy. There are...
a number of factors that we will probably wish to take into account, such as: number of rooms, style of architecture, commuting distance to work and method, provision of local amenities, access to pleasant countryside, and of course, price. Many of these factors work against each other (particularly price), and so the final decision will almost inevitably involve a compromise, based on trading off the house’s rating on different factors.

The example we have just presented is a particularly subjective one, with some factors that are hard to quantify numerically. It does exhibit a feature that is common to multiobjective problems, namely that it is desirable to present the user with a diverse set of possible solutions, representing a range of different trade-offs between objectives.

The alternative is to assign a numerical quality function to each objective, and then combine these scores into a single fitness score using some (usually fixed) weighting. This approach, often called scalarisation, has been used for many years within the operations research and heuristic optimisation communities (see [86, 110] for good reviews), but suffers from a number of drawbacks:

- the use of a weighting function implicitly assumes that we can capture all of the user’s preferences, even before we know what range of possible solutions exist.
- for applications where we are repeatedly solving different instances of the same problem, the use of a weighting function assumes that the user’s preferences remain static, unless we explicitly seek a new weighting every time.

For these reasons optimisation methods that simultaneously find a diverse set of high-quality solutions are attracting increasing interest.

### 12.2 Dominance and Pareto Optimality

The concept of dominance is a simple one: given two solutions, both of which have scores according to some set of objective values (which, without loss of generality, we will assume to be maximised), one solution is said to dominate the other if its score is at least as high for all objectives, and is strictly higher for at least one. We can represent the scores that a solution $A$ gets for $n$ objectives as an $n$-dimensional vector $\mathbf{a}$. Using the $\succeq$ symbol to indicate domination, we can define $A \succeq B$ formally as:

$$A \succeq B \iff \forall i \in \{1, \ldots, n\} \ a_i \geq b_i, \text{ and } \exists i \in \{1, \ldots, n\}, \ a_i > b_i.$$  

For conflicting objectives, there exists no single solution that dominates all others, and we will call a solution nondominated if it is not dominated by any other. All nondominated solutions possess the attribute that their quality cannot be increased with respect to any of the objective functions without