Chapter 7
Multi-objective Optimization Using Surrogates

Ivan Voutchkov and Andy Keane

Abstract. Until recently, optimization was regarded as a discipline of rather theoretical interest, with limited real-life applicability due to the computational or experimental expense involved. Practical multiobjective optimization was considered almost as an utopia even in academic studies due to the multiplication of this expense. This paper discusses the idea of using surrogate models for multiobjective optimization. With recent advances in grid and parallel computing more companies are buying inexpensive computing clusters that can work in parallel. This allows, for example, efficient fusion of surrogates and finite element models into a multiobjective optimization cycle. The research presented here demonstrates this idea using several response surface methods on a pre-selected set of test functions. We aim to show that there are number of techniques which can be used to tackle difficult problems and we also demonstrate that a careful choice of response surface methods is important when carrying out surrogate assisted multiobjective search.

7.1 Introduction

In the world of real engineering design, there are often multiple targets which manufacturers are trying to achieve. For instance in the aerospace industry, a general problem is to minimize weight, cost and fuel consumption while keeping performance and safety at a maximum. Each of these targets might be easy to achieve individually. An airplane made of balsa wood would be very light and will have low fuel consumption, however it will not be structurally strong enough to perform at high speeds or carry useful payload. Also such an airplane might not be very safe,
i.e., robust to various weather and operational conditions. On the other hand, a solid body and a very powerful engine will make the aircraft structurally sound and able to fly at high speeds, but its cost and fuel consumption will increase enormously. So engineers are continuously making trade-offs and producing designs that will satisfy as many requirements as possible, while industrial, commercial and ecological standards are at the same time getting ever tighter.

Multiobjective optimization (MO) is a tool that aids engineers in choosing the best design in a world where many targets need to be satisfied. Unlike conventional optimization, MO will not produce a single solution, but rather a set of solutions, most commonly referred to as Pareto front (PF) \[12\]. By definition it will contain only non-dominated solutions\[1\]. It is up to the engineer to select a final design by examining this front.

Over the past few decades with the rapid growth of computational power, the focus in optimization algorithms in general has shifted from local approaches that find the optimal value with the minimal number of function evaluations to more global strategies which are not necessarily as efficient as local searches but (some more than the others) promise to converge to global solutions, the main players being various strands of genetic and evolutionary algorithms. At the same time, computing power has essentially stopped growing in terms of flops per CPU core. Instead parallel processing is an integral part of any modern computer system. Computing clusters are ever more accessible through various techniques and interfaces such as multi-threading, multi-core, Windows HPC, Condor, Globus, etc.

Parallel processing means that several function evaluations can be obtained at the same time, which perfectly suits the ideology behind genetic and evolutionary algorithms. For example Genetic algorithms are based on the idea borrowed from biological reproduction, where the offspring of two parents copy the best genes of their parents but also introduce some mutation to allow diversity. The entire generation of offspring produced by parents in a generation represent designs that can be evaluated in parallel. The fittest individuals survive and are copied into the next generation, whilst weak designs are given some random chance with low probability to survive. Such parallel search methods are conveniently applicable to multiobjective optimization problems, where the fitness of an individual is measured by how close to the Pareto front this designs is. All individuals are ranked, those that are part of the Pareto front get the lowest (best) rank, the next best have higher rank and so on. Thus the multiobjective optimization is reduced to single objective minimization of the rank of the individuals. This is idea has been developed by Deb and implemented in NSGA2 \[5\].

In the context of this paper, the aim of MO is to produce a well spread out set of optimal designs, with as few function evaluations as possible. There are number of methods published and widely used to do this – MOGA, SPEA, PAES, VEGA, NSGA2, etc. Some are better than others - generally the most popular in the literature are NSGA2 (Deb) and SPEA2 (Zitzler), because they are found to achieve good results for most problems \[2, 3, 4, 5, 6\]. The first is based on genetic algorithms and

---

1 Non-dominated designs are those where to improve performance in any particular goal performance in at least one other goal must be made worse.