8

Multiobjective Combinatorial Optimization

The multicriteria optimization problems we have discussed up to now had a continuous feasible set described by constraints. In the remaining chapters of the book we will be concerned with discrete problems, in which the feasible set is a finite set. These are known as multiobjective discrete and combinatorial optimization problems. They arise naturally in many applications as we shall see in Example 8.1 below when variables are used to model yes/no decisions or objects that are not divisible. In many cases such problems can be understood as optimization problems on some combinatorial structure.

Example 8.1. An important problem in airline operations is the so-called "pairings" or "tours of duty" problem. Let $\{1, \ldots, m\}$ be a set of flights an airline intends to operate. In order to assign the necessary crew to all of these flights this set of flights has to be partitioned into sets of flights that can be operated in sequence by a crew member. Such a sequence of flight has to satisfy quite complex contractual and legal rules. Any sequence of flights that meets all rules is called a pairing or tour of duty. Operating it causes a cost, which consists of pay for the crew member and other costs such as hotel overnights, crew flying as passengers on certain flights, etc. The goal of the airline is to operate all scheduled flights at minimum cost, and it will therefore try to choose the set of pairings that allows to operate the scheduled flights at minimum cost, making sure that each flight is contained in one pairing. In general the number of legal pairings is huge.

However, disruptions caused by bad weather, mechanical problems etc. may occur during operation. These disruptions may result in missed connections and inevitably lead to additional costs caused by delays. An airline will therefore also be interested in selecting pairings that are robust in the sense that they are less susceptible to disruptions. This objective is in conflict with minimization of cost, because it favours longer breaks between flights to make

it possible to compensate earlier delays. Longer breaks, on the other hand, increase cost due to unproductive time.

We can formulate this problem as follows. Let

$$x_i = \begin{cases} 1 & \text{if ToD } i \text{ is selected} \\ 0 & \text{otherwise.} \end{cases}$$

Furthermore, let c_i be a measure for the cost of ToD i and r_i be a measure of the possible delay caused by operating ToD i. We need to solve the bicriterion optimization problem

$$\begin{aligned} \min \sum_{i=1}^{n} c_i x_i \\ \min \sum_{i=1}^{n} r_i x_i \\ \text{subject to } \sum_{i=1}^{n} a_{ji} x_i, = 1 \quad j = 1, \ldots, m \\ x \in \{0,1\}^n. \end{aligned}$$

This is a bicriterion integer programming problem. Note that there are finitely many feasible solutions. More details about this problem can be found in Ehrgott and Ryan (2002). □

In this chapter we will give an introduction to multiobjective discrete optimization, including computational complexity, a general scalarization model for multiobjective integer programs, and solution algorithms for the case that the feasible set is explicitly given.

8.1 Basics

Combinatorial optimization problems have a finite set of feasible solutions. This has significant impact on the way we deal with these problems, both in theory and solution techniques. We shall first introduce combinatorial optimization problems and the multicriteria optimization classes we consider in the following chapters. Some basic observations show that in a multicriteria context combinatorial optimization is quite different from the general or linear optimization framework we have considered in earlier chapters of this text. In particular, we give a brief introduction to the concepts of computational complexity, such as $\mathcal{NP}$-completeness and $\#\mathcal{P}$-completeness. In the subsequent chapters we prove results on computational complexity. These chapters feature some selected combinatorial problems, which are chosen to illustrate one