Chapter 3

Modelling in Combinatorial Optimization

Combinatorial optimization usually deals with the optimization of an objective function over a *finite domain*. In most of the cases the variables are restricted to *integers or natural numbers* and there are restrictions that allow only finitely many *feasible solutions*. Although in principle possible, exhaustive search through all finitely many feasible solutions is practically not an option because of their huge number. Probably the most famous combinatorial optimization problem is the *travelling salesman problem*, other examples are the *minimum spanning tree problem*, the *knapsack problem*, or the *bin packing problem*.

3.1 An Introductory Example

Suppose we run an online shop and we deliver our goods in boxes of maximum capacity C. We have a concrete order with n items of sizes $s_1, \dots, s_n$, respectively. The size of an item could be for instance the weight or the volume and the capacity would then be the maximum weight allowed or the maximum space available in the box. The question is how to pack the items into a minimal number of boxes such that all capacity restrictions are still satisfied. In case the capacity is interpreted as volume we neglect the geometrical problem of fitting items of a certain shape into the boxes having a certain shape, i.e. if we have items with a total volume of x then we assume these items fit into a box with volume $V \geq x$. It is clear that in practice, by the geometry of the box and the items, this might lead to inappropriate solutions. As a simple example, consider a sphere of volume 1, which has a diameter of ≈ 1.24. Clearly, the spere dos not fit into a unit cube with side lenghts 1, although the volume restriction would be satisfied.

3.2 Modelling the Packing Problem

The problem described in Section 3.1 is known in literature as the *bin packing problem*.

Problem 3.1: Bin Packing Problem

Given: Positive numbers $a_1, \dots, a_n, A$.

Find: $k \in \mathbb{N}$ and $p\colon \mathbb{N}_{1,n} \to \mathbb{N}_{1,k}$ such that

$$\forall_{1 \leq j \leq k} \sum_{\substack{i \\ p(i)=j}} a_i \leq A.$$

We consider n items $I_1, \dots, I_n$ and need to find a number k of bins $B_1, \dots, B_k$ such that each I_i goes into one of the B_j. The packing function p assigns each item index i a bin

index j, and $p(i) = j$ means that item I_i is packed into bin B_j. If k and p satisfy the above conditions we call the pair (k, p) a *feasible solution* (for the bin packing problem with respect to $a_1, \ldots, a_n, A$).

Often, in particular in the context of complexity theory, problems are stated as decision problems. For the bin packing problem, this variant is the following.

PROBLEM 3.2: BIN PACKING DECISION PROBLEM

Given: Positive numbers $a_1, \ldots, a_n, A$ and $k \in \mathbb{N}$.

Question: Does there exist a function $p\colon \mathbb{N}_{1,n} \to \mathbb{N}_{1,k}$ such that (k, p) a feasible solution for the bin packing problem with respect to $a_1, \ldots, a_n, A$?

If there is a feasible solution (k, p) for the bin packing problem then of course (m, q) is again a feasible solution for all $m > k$ and $q\colon \mathbb{N}_{1,n} \to \mathbb{N}_{1,m}$ such that $q(i) = p(i)$ for all $1 \leq i \leq n$, hence there are infinitely many solutions. It is then pretty natural to ask for the best possible solution.

PROBLEM 3.3: OPTIMAL BIN PACKING PROBLEM

Given: Positive numbers $a_1, \ldots, a_n, A$.

Find: $k \in \mathbb{N}$ and $p\colon \mathbb{N}_{1,n} \to \mathbb{N}_{1,k}$ such that

1. (k, p) a feasible solution for the bin packing problem w.r.t. $a_1, \ldots, a_n, A$ and
2. k is minimal, i.e.

$$\forall_{m<k} \; \forall_{q\colon \mathbb{N}_{1,n} \to \mathbb{N}_{1,m}} \; (m, q) \text{ is not a feasible solution for the bin packing problem with respect to } a_1, \ldots, a_n, A.$$

3.2.1 Bin Packing as a Linear Optimization Problem

A first approach for solving the optimal bin packing problem is to re-formulate the problem as a linear programming (linear optimization) problem. First we observe that $k \leq n$. We introduce *binary decision variables* x_{ij} for $1 \leq i, j \leq n$ and y_j for $1 \leq j \leq n$ with the following meaning:

$$x_{ij} := \begin{cases} 1 & \text{item } I_i \text{ goes into bin } B_j \\ 0 & \text{otherwise} \end{cases} \qquad y_j := \begin{cases} 1 & \text{bin } B_j \text{ will be occupied} \\ 0 & \text{otherwise} \end{cases}$$

The feasibility of a solution for the bin packing problem can then be described by

$$\forall_{1 \leq j \leq n} \sum_{i=1}^{n} x_{ij} a_i \leq A y_j \tag{3.1}$$

$$\forall_{1 \leq i \leq n} \sum_{j=1}^{n} x_{ij} = 1, \tag{3.2}$$

where (3.1) captures the capacity restrictions for each bin and (3.2) enforces that each article goes into exactly one bin. The number of bins used is obviously $\sum_{j=1}^{n} y_j$, hence, (3.1) and (3.2) together with

$$\sum_{j=1}^{n} y_j \longrightarrow \text{Min} \tag{3.3}$$

form an *integer linear programming problem* with the integer variables

$$x_{ij} \in \{0,1\} \text{ for } 1 \leq i,j \leq n \quad \text{and} \quad y_j \in \{0,1\} \text{ for } 1 \leq j \leq n.$$

We write *BPLP*(a, A) for this problem. It is easy to reconstruct a solution for the optimal bin packing problem from a solution (x, y) of *BPLP*(a, A). For this, let $k := \sum_{j=1}^{n} y_j$ and let $\pi\colon \mathbb{N}_{1,n} \to \mathbb{N}_{1,n}$ be a permutation of $\mathbb{N}_{1,n}$ such that

$$\bigwedge_{1 \leq i \leq k}^{\forall} y_{\pi(i)} = 1 \wedge \bigwedge_{k < i \leq n}^{\forall} y_{\pi(i)} = 0. \tag{3.4}$$

π permutes the bins in such a way that the first k bins $y_{\pi(1)}, \ldots, y_{\pi(k)}$ will be occupied and the remaining $n-k$ bins $y_{\pi(k+1)}, \ldots, y_{\pi(n)}$ are empty. From (3.2) it follows immediately that for every $1 \leq i \leq n$ there is a unique $1 \leq j \leq n$ with $x_{ij} = 1$, thus the function

$$p\colon \mathbb{N}_{1,n} \to \mathbb{N}_{1,k}, i \mapsto \text{the unique } j \text{ with } x_{i\pi(j)} = 1$$

is well-defined, since π just permutes the columns of (x_{ij}) in such a way that zero-columns are moved to the end. In order to see that (k, p) is feasible, we take $1 \leq j \leq k$ arbitrary but fixed and now

$$\sum_{\substack{i \\ p(i)=j}} a_i = \sum_{\substack{i \\ x_{i\pi(j)}=1}} a_i = \sum_{i=1}^{n} x_{i\pi(j)} a_i \overset{(3.1)}{\leq} A y_{\pi(j)} \overset{(3.4)}{=} A.$$

The minimality of k follows easily from the definition of k together with (3.3).

EXAMPLE 3.4

Consider 7 articles with weights

$$a_1 = 0.2 \quad a_2 = 0.5 \quad a_3 = 0.4 \quad a_4 = 0.7 \quad a_5 = 0.1 \quad a_6 = 0.3 \quad a_7 = 0.8$$

and bins with maximum capacity of 1. The formulation as a linear programming problem now assumes at most $n = 7$ bins and introduces 49 variables $x_{11}, \ldots, x_{77}$ plus 7 variables $y_1, \ldots, y_7$, hence, a total of 56 variables. We have 7 capacity restrictions (3.1)

$$0.2x_{1j} + 0.5x_{2j} + 0.4x_{3j} + 0.7x_{4j} + 0.1x_{5j} + 0.3x_{6j} + 0.8x_{7j} \leq y_j \quad \text{for } 1 \leq j \leq 7$$

and 7 unicity restrictions (3.2)

$$x_{i1} + x_{i2} + x_{i3} + x_{i4} + x_{i5} + x_{i6} + x_{i7} = 1 \quad \text{for } 1 \leq i \leq 7,$$

which results in a solution

$$(1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,1,1,0,0,0,0),$$

which translates into $k =$ and a mapping p for the articles

$$p(1) = 1 \quad p(2) = 2 \quad p(3) = 2 \quad p(4) = 3 \quad p(5) = 2 \quad p(6) = 3 \quad p(7) = 1.$$

3.2.2 Heuristic Approximation Algorithms for the Bin Packing Problem

For big instances of bin packing problems the integer linear programming problem can be computationally very expensive. Therefore, approximation algorithms based on heuristics are very popular for solving huge bin packing problems.

In the following, we write P for a problem and $P(x)$ for an instance of problem P with input x. If A is an algorithm, then $A(x)$ denotes the result of algorithm A applied to input x.

DEFINITION 3.5: APPROXIMATION ALGORITHM

Let P be an optimization problem and $\bar{P}$ the relaxed problem ignoring optimality. We call A an *approximation algorithm* for problem P if and only if every $y = A(x)$ is still a solution of $\bar{P}(x)$ but not necessarily a solution of $P(x)$.

DEFINITION 3.6: APPROXIMATION QUALITY

Let P be an optimization problem and $y(x)$ a solution for $P(x)$. Let furthermore $k \geq 1$. We call A a *k-approximation algorithm* for problem P if and only if for all admissible inputs x of P

$$A(x) \begin{cases} \leq ky(x) & \text{if } P \text{ is a minimization problem} \\ \geq \frac{1}{k}y(x) & \text{if } P \text{ is a maximization problem} \end{cases}$$

THEOREM 3.7

If $P \neq NP$[a], then there is no k-approximation algorithm for the optimal bin packing problem with $k < \frac{3}{2}$.

[a] we do not go into details

Data: Positive numbers $a_1, \ldots, a_n, A$.
Result: k, p such that (k, p) is a solution for the bin packing problem

$k = 0$
sort a into decreasing order
for $j = 1, \ldots, n$ **do**
  $c_j = A$
end
for $i = 1, \ldots, n$ **do**
  $m = 0$
  for $j = 1, \ldots, k$ **do**
    if $a_i \leq c_j$ **then**
      $m = j$
      $p(i) = j$
      $c_j = c_j - a_i$
    end
  end
  if $m = 0$ **then**
    $k = k + 1$
    $p(i) = k$
    $c_k = c_k - a_i$
  end
end

Algorithm 2: First Fit Decreasing Heuristic (FFD)

THEOREM 3.8

FFD is a $\frac{3}{2}$-approximation algorithm for the optimal bin packing problem.

THEOREM 3.9

For all instances x of the bin packing problem with solution $y(x)$ we have

$$FFD(x) \leq \frac{11}{9} y(x) + \frac{2}{3}.$$

EXAMPLE 3.10

Consider again 7 articles with weights

$$a_1 = 0.2 \quad a_2 = 0.5 \quad a_3 = 0.4 \quad a_4 = 0.7 \quad a_5 = 0.1 \quad a_6 = 0.3 \quad a_7 = 0.8$$

and bins with maximum capacity of 1. Running the FFD-algorithm on this instance of the bin packing problem gives the same solution as shown in Example 3.4.