1.3 FORMULATING IPS AND BIPS

As in linear programming, translating a problem description into a formulation should be done systematically, and a clear distinction should be made between the data of the problem instance, and the variables (or unknowns) used in the model.

(i) Define what appear to be the necessary variables.

(ii) Use these variables to define a set of constraints so that the feasible points correspond to the feasible solutions of the problem.

(iii) Use these variables to define the objective function.

If difficulties arise, define an additional or alternative set of variables and iterate.

Defining variables and constraints may not always be as easy as in linear programming. Especially for *COPs*, we are often interested in choosing a subset $S \subseteq N$. For this we typically make use of the *incidence vector of S*, which is the n-dimensional 0–1 vector x^S such that $x^S_j = 1$ if $j \in S$, and $x^S_j = 0$ otherwise.

Below we formulate four well-known integer programming problems.

The Assignment Problem

There are n people available to carry out n jobs. Each person is assigned to carry out exactly one job. Some individuals are better suited to particular jobs than others, so there is an estimated cost c_{ij} if person i is assigned to job j. The problem is to find a minimum cost assignment.

Definition of the variables.
$x_{ij} = 1$ if person i does job j, and $x_{ij} = 0$ otherwise.

Definition of the constraints.
Each person i does one job:

$$\sum_{j=1}^{n} x_{ij} = 1 \text{ for } i = 1, \ldots, n.$$

Each job j is done by one person:

$$\sum_{i=1}^{n} x_{ij} = 1 \text{ for } j = 1, \ldots, n.$$

The variables are 0–1:

$$x_{ij} \in \{0, 1\} \text{ for } i = 1, \ldots, n, j = 1, \ldots, n.$$

Definition of the objective function.
The cost of the assignment is minimized:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}.$$

The 0–1 Knapsack Problem

There is a budget b available for investment in projects during the coming year and n projects are under consideration, where a_j is the outlay for project j, and c_j is its expected return. The goal is to choose a set of projects so that the budget is not exceeded and the expected return is maximized.

Definition of the variables.
$x_j = 1$ if project j is selected, and $x_j = 0$ otherwise.

Definition of the constraints.
The budget cannot be exceeded:

$$\sum_{j=1}^{n} a_j x_j \leq b.$$

The variables are 0–1:

$$x_j \in \{0,1\} \text{ for } j = 1, \ldots, n.$$

Definition of the objective function.
The expected return is maximized:

$$\max \sum_{j=1}^{n} c_j x_j.$$

The Set Covering Problem

Given a certain number of regions, the problem is to decide where to install a set of emergency service centers. For each possible center the cost of installing a service center, and which regions it can service are known. For instance, if the centers are fire stations, a station can service those regions for which a fire engine is guaranteed to arrive on the scene of a fire within 8 minutes. The goal is to choose a minimum cost set of service centers so that each region is covered.

First we can formulate it as a more abstract *COP*. Let $M = \{1, \ldots, m\}$ be the set of regions, and $N = \{1, \ldots, n\}$ the set of potential centers. Let $S_j \subseteq M$ be the regions that can be serviced by a center at $j \in N$, and c_j its installation cost. We obtain the problem:

$$\min_{T \subseteq N} \{ \sum_{j \in T} c_j : \cup_{j \in T} S_j = M \}.$$

Now we formulate it as a *BIP*. To facilitate the description, we first construct a 0–1 *incidence matrix* A such that $a_{ij} = 1$ if $i \in S_j$, and $a_{ij} = 0$ otherwise. Note that this is nothing but processing of the data.

Definition of the variables.
$x_j = 1$ if center j is selected, and $x_j = 0$ otherwise.

Definition of the constraints.
At least one center must service region i:

$$\sum_{j=1}^{n} a_{ij} x_j \geq 1 \text{ for } i = 1, \ldots, m.$$

The variables are 0–1:

$$x_j \in \{0, 1\} \text{ for } j = 1, \ldots, n.$$

Definition of the objective function.
The total cost is minimized:

$$\min \sum_{j=1}^{n} c_j x_j.$$

The Traveling Salesman Problem *(TSP)*

This is perhaps the most notorious problem in Operations Research because it is so easy to explain, and so tempting to try and solve. A salesman must visit each of n cities exactly once and then return to his starting point. The time taken to travel from city i to city j is c_{ij}. Find the order in which he should make his tour so as to finish as quickly as possible.

This problem arises in a multitude of forms: a truck driver has a list of clients he must visit on a given day, or a machine must place modules on printed circuit boards, or a stacker crane must pick up and depose crates. Now we formulate it as a *BIP*.

Definition of the variables.
$x_{ij} = 1$ if the salesman goes directly from town i to town j, and $x_{ij} = 0$ otherwise. (x_{ii} is not defined for $i = 1, \ldots, n$.)

Definition of the constraints.
He leaves town i exactly once:

$$\sum_{j: j \neq i} x_{ij} = 1 \text{ for } i = 1, \ldots, n.$$

He arrives at town j exactly once:

$$\sum_{i: i \neq j} x_{ij} = 1 \text{ for } j = 1, \ldots, n.$$

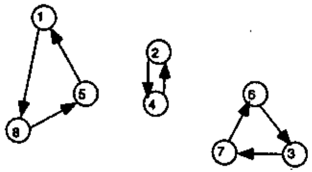

Fig. 1.2 Subtours

So far these are precisely the constraints of the assignment problem. A solution to the assignment problem might give a solution of the form shown in Figure 1.2 (i.e., a set of disconnected subtours). To eliminate these solutions, we need more constraints that guarantee connectivity by imposing that the salesman must pass from one set of cities to another, so-called *cut-set* constraints:

$$\sum_{i \in S} \sum_{j \notin S} x_{ij} \geq 1 \text{ for } S \subset N, S \neq \phi.$$

An alternative is to replace these constraints by *subtour elimination* constraints:

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq \mid S \mid -1 \text{ for } S \subset N, 2 \leq |S| \leq n-1.$$

The variables are 0–1:

$$x_{ij} \in \{0,1\} \text{ for } i = 1, \ldots, n, j = 1, \ldots, n, i \neq j.$$

Definition of the objective function.
The total travel time is minimized:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}.$$

1.4 THE COMBINATORIAL EXPLOSION

The four problems we have looked at so far are all combinatorial in the sense that the optimal solution is some subset of a finite set. Thus in principle these problems can be solved by enumeration . To see for what size of problem instances this is a feasible approach, we need to count the number of possible solutions.

The Assignment Problem. There is a one-to-one correspondence between assignments and permutations of $\{1, \ldots, n\}$. Thus there are $n!$ solutions to compare.

The Knapsack and Covering Problems. In both cases the number of subsets is 2^n. For the knapsack problem with $b = \sum_{j=1}^{n} a_j/2$, at least half of the subsets are feasible, and thus there are at least 2^{n-1} feasible subsets.

The Traveling Salesman Problem. Starting at city 1, the salesman has $n-1$ choices. For the next choice $n-2$ cities are possible, and so on. Thus there are $(n-1)!$ feasible tours.

In Table 1.1 we show how rapidly certain functions grow. Thus a *TSP* with $n = 101$ has approximately 9.33×10^{157} tours.

n	$\log n$	$n^{0.5}$	n^2	2^n	$n!$
10	3.32	3.16	10^2	1.02×10^3	3.6×10^6
100	6.64	10.00	10^4	1.27×10^{30}	9.33×10^{157}
1000	9.97	31.62	10^6	1.07×10^{301}	4.02×10^{2567}

Table 1.1 Some typical functions

The conclusion to be drawn is that using complete enumeration we can only hope to solve such problems for very small values of n. Therefore we have to devise some more intelligent algorithms, otherwise the reader can throw this book out of the window.

1.5 MIXED INTEGER FORMULATIONS

Modeling Fixed Costs

Suppose we wish to model a typical nonlinear fixed charge cost function:

$$h(x) = f + px \text{ if } 0 < x \leq C \text{ and } h(x) = 0 \text{ if } x = 0$$

with $f > 0$ and $p > 0$ (see Figure 1.3).

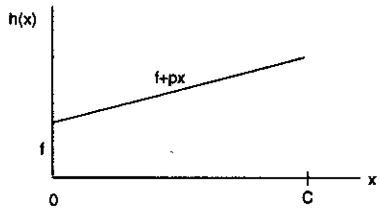

Fig. 1.3 Fixed cost function

Definition of an additional variable.

$$y = 1 \text{ if } x > 0 \text{ and } y = 0 \text{ otherwise.}$$

Definition of the constraints and objective function.
We replace $h(x)$ by $fy + px$, and add the constraints $x \leq Cy, y \in \{0, 1\}$.

Note that this is not a completely satisfactory formulation, because although the costs are correct when $x > 0$, it is possible to have the solution $x = 0, y = 1$. However, as the objective is minimization, this will typically not arise in an optimal solution.

Uncapacitated Facility Location *(UFL)*

Given a set of potential depots $N = \{1, \dots, n\}$ and a set $M = \{1, \dots, m\}$ of clients, suppose there is a fixed cost f_j associated with the use of depot j, and a transportation cost c_{ij} if all of client i's order is delivered from depot j. The problem is to decide which depots to open, and which depot serves each client so as to minimize the sum of the fixed and transportation costs. Note that this problem is similar to the covering problem, except for the addition of the variable transportation costs.

Definition of the variables.
We introduce a fixed cost or depot opening variable $y_j = 1$ if depot j is used, and $y_j = 0$ otherwise.
x_{ij} is the fraction of the demand of client i satisfied from depot j.

Definition of the constraints.
Satisfaction of the demand of client i:

$$\sum_{j=1}^{n} x_{ij} = 1 \text{ for } i = 1, \dots, m.$$

To represent the link between the x_{ij} and the y_j variables, we note that $\sum_{i \in M} x_{ij} \leq m$, and use the fixed cost formulation above to obtain:

$$\sum_{i \in M} x_{ij} \leq m y_j \text{ for } j \in N, y_j \in \{0, 1\} \text{ for } j \in N.$$

Definition of the objective function.
The objective is $\sum_{j \in N} h_j(x_{1j}, \dots, x_{mj})$ where $h_j(x_{1j}, \dots, x_{mj}) = f_j + \sum_{i \in M} c_{ij} x_{ij}$ if $\sum_{i \in M} x_{ij} > 0$, so we obtain

$$\min \sum_{i \in M} \sum_{j \in N} c_{ij} x_{ij} + \sum_{j \in N} f_j y_j.$$

Uncapacitated Lot-Sizing *(ULS)*

The problem is to decide on a production plan for an n-period horizon for a single product. The basic model can be viewed as having data:

f_t is the fixed cost of producing in period t.
p_t is the unit production cost in period t.
h_t is the unit storage cost in period t.
d_t is the demand in period t.

We use the natural (or obvious) *variables*:

x_t is the amount produced in period t.
s_t is the stock at the end of period t.
$y_t = 1$ if production occurs in t, and $y_t = 0$ otherwise.

To handle the fixed costs, we observe that a priori no upper bound is given on x_t. Thus we either must use a very large value $C = M$, or calculate an upper bound based on the problem data.
For *constraints* and *objective* we obtain:

$$\begin{aligned}
\min \sum_{t=1}^{n} p_t x_t + \sum_{t=1}^{n} h_t s_t \quad &+ \quad \sum_{t=1}^{n} f_t y_t \\
s_{t-1} + x_t \quad &= \quad d_t + s_t \text{ for } t = 1, \ldots, n \\
x_t \quad &\leq \quad M y_t \text{ for } t = 1, \ldots, n \\
s_0 = 0, s_t, x_t \quad &\geq \quad 0, y_t \in \{0,1\} \text{ for } t = 1, \ldots, n.
\end{aligned}$$

If we impose that $s_n = 0$, then we can tighten the variable upper bound constraints to $x_t \leq (\sum_{i=t}^{n} d_i) y_t$. Note also that by substituting $s_t = \sum_{i=1}^{t} x_i - \sum_{i=1}^{t} d_i$, the objective function can be rewritten as $\sum_{t=1}^{n} c_t x_t + \sum_{t=1}^{n} f_t y_t - K$ where $c_t = p_t + h_t + \ldots + h_n$ and the constant $K = \sum_{t=1}^{n} h_t (\sum_{i=1}^{t} d_i)$.

Discrete Alternatives or Disjunctions

Suppose $x \in R^n$ satisfies $0 \leq x \leq u$, and either $a^1 x \leq b_1$ or $a^2 x \leq b_2$ (see Figure 1.4). We introduce binary variables y_i for $i = 1, 2$. Then if $M \geq \max\{a^i x - b_i : 0 \leq x \leq u\}$ for $i = 1, 2$, we take as constraints:

$$\begin{gathered}
a^i x - b_i \leq M(1 - y_i) \text{ for } i = 1, 2 \\
y_1 + y_2 = 1, y_i \in \{0,1\} \text{ for } i = 1, 2 \\
0 \leq x \leq u.
\end{gathered}$$

Now if $y_1 = 1$, x satisfies $a^1 x \leq b_1$ whereas $a^2 x \leq b_2$ is inactive, and conversely if $y_2 = 1$.

Such disjunctions arise naturally in scheduling problems. Suppose that two jobs must be processed on the same machine and cannot be processed