Chapter 1

A PRIMER IN COLUMN GENERATION

Jacques Desrosiers
Marco E. Lübbecke

Abstract We give a didactic introduction to the use of the column generation technique in linear and in particular in integer programming. We touch on both, the relevant basic theory and more advanced ideas which help in solving large scale practical problems. Our discussion includes embedding Dantzig-Wolfe decomposition and Lagrangian relaxation within a branch-and-bound framework, deriving natural branching and cutting rules by means of a so-called compact formulation, and understanding and influencing the behavior of the dual variables during column generation. Most concepts are illustrated via a small example. We close with a discussion of the classical cutting stock problem and some suggestions for further reading.

1. Hands-on experience

Let us start right away by solving a constrained shortest path problem. Consider the network depicted in Figure 1.1. Besides a cost c_{ij} there is a resource consumption t_{ij} attributed to each arc $(i, j) \in A$, say a traversal time. Our goal is to find a shortest path from node 1 to node 6 such that the total traversal time of the path does not exceed 14 time units.

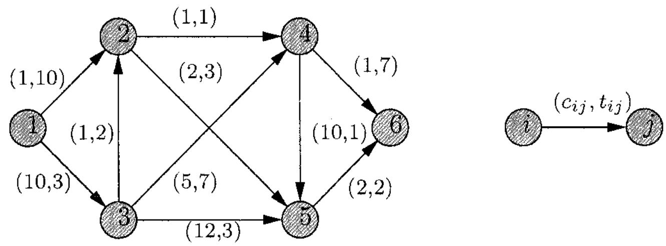

Figure 1.1. Time constrained shortest path problem, (p. 599 Ahuja et al., 1993).

One way to state this particular network flow problem is as the integer program (1.1)–(1.6). One unit of flow has to leave the source (1.2) and has to enter the sink (1.4), while flow conservation (1.3) holds at all other nodes. The time resource constraint appears as (1.5).

$$z^\star := \min \sum_{(i,j)\in A} c_{ij}x_{ij} \tag{1.1}$$

$$\text{subject to} \quad \sum_{j:\,(1,j)\in A} x_{1j} = 1 \tag{1.2}$$

$$\sum_{j:\,(i,j)\in A} x_{ij} - \sum_{j:\,(j,i)\in A} x_{ji} = 0 \quad i = 2,3,4,5 \tag{1.3}$$

$$\sum_{i:\,(i,6)\in A} x_{i6} = 1 \tag{1.4}$$

$$\sum_{(i,j)\in A} t_{ij}x_{ij} \leq 14 \tag{1.5}$$

$$x_{ij} = 0 \text{ or } 1 \quad (i,j) \in A \tag{1.6}$$

An inspection shows that there are nine possible paths, three of which consume too much time. The optimal integer solution is path 13246 of cost 13 with a traversal time of 13. How would we find this out? First note that the resource constraint (1.5) prevents us from solving our problem with a classical shortest path algorithm. In fact, no polynomial time algorithm is likely to exist since the resource constrained shortest path problem is $\mathcal{NP}$-hard. However, since the problem is *almost* a shortest path problem, we would like to exploit this embedded well-studied structure algorithmically.

1.1 An equivalent reformulation: Arcs vs. paths

If we ignore the complicating constraint (1.5), the easily tractable remainder is $X = \{x_{ij} = 0 \text{ or } 1 \mid (1.2)\text{–}(1.4)\}$. It is a well-known result in network flow theory that an extreme point $\mathbf{x}_p = (x_{pij})$ of the polytope defined by the convex hull of X corresponds to a path $p \in P$ in the network. This enables us to express any arc flow as a convex combination of path flows:

$$x_{ij} = \sum_{p\in P} x_{pij}\lambda_p \quad (i,j) \in A \tag{1.7}$$

$$\sum_{p\in P} \lambda_p = 1 \tag{1.8}$$

$$\lambda_p \geq 0 \quad p \in P. \tag{1.9}$$