Algorithmic Aspects of Communication Networks

Chapter 3
General Optimization Methods for Network Design

Part 2

- **The idea of branch and bound methods**
  - Consider a (hard to solve) optimization problem

$$\text{minimize } f(x) \qquad \text{subject to } x \in M \qquad (1)$$

  - Associate a *relaxed* optimization problem

$$\text{minimize } g(x) \qquad \text{subject to } x \in R \qquad (2)$$

  such that

    - $R \supseteq M$,
    - if $x \in R$, then $g(x) = f(x)$, and
    - (2) can be solved efficiently.

Branch and Bound – the basic algorithm

- Recall:

An *optimal solution* for (1) is an element x^* of M such that $f(x) \geq f(x^*)$ for all x in M (likewise for (2)).

- Claim 1:

(a) If y^* is an optimal solution for (2), then $f(x) \geq f(y^*)$ for all x in M.
(b) If y^* is an optimal solution for (2) and $y^* \in M$, then y^* is an optimal solution for (1) too.

- Claim 2:

Let $M = M_1 \dot{\cup} M_2$ be a partition of M, and let x^*_k be an optimal solution for the OP

$$\text{minimize } f(x) \quad \text{subject to} \quad x \in M_k \qquad (1_k)$$

($k=1,2$). Then $\operatorname{argmin}_x \{f(x^*_1), f(x^*_2)\}$ is an optimal solution for (1).

Branch and Bound – the basic algorithm

- Let R' be a nonempty subset of R, and consider the optimization problem

$$\text{minimize } g(y) \quad \text{subject to} \quad y \in R' \qquad\qquad (R').$$

  - If (R') has a solution, then SOLVE(R') returns a pair $(y^*, g(y^*))$ consisting of an optimal solution y^* of (R') and the corresponding value $g(y^*)$, otherwise SOLVE(R') returns (n,n).

  - If SOLVE(R') yields an optimal solution $y^* \notin M$, then BRANCH(R', y^*) returns two disjoint subsets R'_1, R'_2 of R' such that

$$M \cap R' = (M \cap R'_1) \cup (M \cap R'_2).$$

Branch and Bound – the basic algorithm

The idea of the algorithm

- Initialization $\mathbf{L} \leftarrow \{R\}$; $best \leftarrow \infty$;
- While $\mathbf{L} \neq \emptyset$ do

```
begin
  choose B ∈ L;                                   % according to a specific rule %
  (y*, g(y*)) ← SOLVE(B);
  if y* ∈ M and g(y*) < best then
      begin
        best ← g(y*);
        y_best ← y*;
        remove B from L;
      end;
  if g(y*) ≥ best then remove B from L;           % bounding %
  else
      begin                                       % branching %
        (B1, B2) ← BRANCH(B, y*);
              L ← L ∪ {B1, B2};
      end;
end
```

Branch and Bound – the basic algorithm

A more concrete recursive realization of Branch and Bound

- Initialization $A \leftarrow M$, $B \leftarrow R$, $best \leftarrow \infty$

- procedure BB(A,B,f,g)

```
begin
   (y*, g(y*)) ← SOLVE(B)
   if y* ∈ A then
      if g(y*) < best then
        begin best ← g(y*); return (y*, g(y*))
      else
       if  g(y*) ≥ best  then  return        % bounding  %
       else                                  % branching  %
         begin
            (B1, B2) ← BRANCH(B, y*);
            BB(A, B1,f,g);
            BB(A, B2,f,g);
          end;
 end
```

Branch and Bound – the basic algorithm

- Remarks:
  - Branch and Bound yields an (exact) optimal solution provided there is a constant K (depending on the input) such that all subproblems obtained after at most K repetitions of BRANCH either have no optimal solution or their optimal solutions are in M.
  - Branch and Bound is not (necessarily) efficient.

Branch and Bound for MIP

- Consider the MIP (1)

  minimize $f(\boldsymbol{x}) = \boldsymbol{c}\,\boldsymbol{x}$

  subject to $\boldsymbol{A}\boldsymbol{x} = \mathbf{b}$,
  $\boldsymbol{x} \geq \boldsymbol{0}$, and
  x_j is integer for all $k \in I$

  where $\boldsymbol{x} = (x_{1,\ldots,}x_n)$, $\boldsymbol{b} \geq \boldsymbol{0}$, $\boldsymbol{A}$ has rank $m \leq n$, and $I \subseteq \{1, .., n\}$

- Choose as relaxed problem the LOP (2) (called *LP-relaxation*)

  minimize $f(\boldsymbol{x}) = \boldsymbol{c}\,\boldsymbol{x}$

  subject to $\boldsymbol{A}\boldsymbol{x} = \mathbf{b}$,
  $\boldsymbol{x} \geq \boldsymbol{0}$

  where $\boldsymbol{x} = (x_{1,\ldots,}x_n)$, $\boldsymbol{b} \geq \boldsymbol{0}$, and $\boldsymbol{A}$ has rank $m \leq n$

Branch and Bound for MIP

- SOLVE can be any method to solve LOPs (e.g. the simplex algortihm).

- If $(y^*, g(y^*))$ is an optimal solution where y^*_k is not an integer, then BRANCH(B, y^*) 'adds' new inequalitities to B, i.e.

$$\text{BRANCH}(B, y^*) = (B_1, B_2) \text{ where}$$
$$B_1 = \{y \in B \mid y_k \leq [y^*_k]\},$$
$$B_2 = \{y \in B \mid y_k \geq \{y^*_k\}\}.$$

(Notation: $[y^*_k]$ is the smallest integer not smaller than y^*_k and $\{y^*_k\}$ is the greatest integer not greater than y^*_k.)

Cutting planes for MIP

- Consider the MIP (1)

  minimize $\quad f(\boldsymbol{x}) = \boldsymbol{c}\,\boldsymbol{x}$

  subject to $\quad \boldsymbol{x} \in M = \{\boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x} = \mathbf{b},\ \boldsymbol{x} \geq \boldsymbol{0},$ and x_j is integer for all $k \in I\}$

  where $\quad \boldsymbol{x} = (x_{1,\dots,}x_n)$ and $I \subseteq \{1,..,n\}$,

and its LP-relaxation (2)

  minimize $\quad f(\boldsymbol{x}) = \boldsymbol{c}\,\boldsymbol{x}$

  subject to $\quad \boldsymbol{x} \in R = \{\boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x} = \mathbf{b},\ \boldsymbol{x} \geq \boldsymbol{0}\}$.

Cutting planes for MIP

- A *valid cut* for (1) is an equality $\mathbf{d}\,\mathbf{x} \geq q$ such that
  - $\{\boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x} = \mathbf{b}, \boldsymbol{x} \geq \boldsymbol{0}, \mathbf{d}\,\mathbf{x} \geq q\} \neq R$ and
  - $\{\boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x} = \mathbf{b}, \boldsymbol{x} \geq \boldsymbol{0}$, and x_j is integer for all $k \in I$, $\mathbf{d}\,\mathbf{x} \geq q\} = M$.

- Outline of a cutting plane algorithm

<u>begin</u>
 $B \leftarrow R$;
 $(y^*, g(y^*)) \leftarrow \text{SOLVE}(B)$;
 <u>while</u> $y^* \notin M$ <u>do</u>
 <u>begin</u>
  compute a valid cut $\mathbf{d}\,\mathbf{x} \geq q$ for (B);
  $B \leftarrow \{\mathbf{x} \in B \mid \mathbf{d}\,\mathbf{x} \geq q\}$;
  $(y^*, g(y^*)) \leftarrow \text{SOLVE}(B)$;
 <u>end;</u>
<u>end</u>

Cutting planes for MIP

- There are two kinds of cuts:
  - problem specific ones (e.g. for the knap sack problem), and
  - general purpose ones (Gomory cuts).

- Gomory cuts
  - Let $\boldsymbol{x} = (x_{1,\ldots,}x_n)$ be a basic solution of the LP-relaxation such that x_i is not an integer.
  - Then x_i is a basic variable and from the simplex tableau we know that

$$x_i + \sum a_{ij} x_j = b_i \qquad (a)$$

  - Equation (a) implies

$$x_i + \sum [a_{ij}] x_j - [b_i] = b_i - [b_i] - \sum (a_{ij} - [a_{ij}]) x_j \qquad (b)$$

Cutting planes for MIP

- For any point $\mathbf{x} \in M$ the right hand site of (b) is less than 1, and the left hand site is an integer.
- Consequently,

$$b_i - [b_i] - \sum (a_{ij} - [a_{ij}])\, x_i \leq 0 \qquad \text{(c)}$$

  for any point $\mathbf{x} \in M$.

- Furthermore, for the original basic solution $\boldsymbol{x} = (x_{1,\ldots,}x_n)$ is

$$b_i - [b_i] - \sum (a_{ij} - [a_{ij}])\, x_i = b_i - [b_i]$$

  (because all non basic variables =0) not an integer, and therefore

$$b_i - [b_i] - \sum (a_{ij} - [a_{ij}])\, x_i = b_i - [b_i] > 0,$$

  i.e. (c) excludes $\mathbf{x}$.

- Hence (c) is a valid cut. The inequality (c) is called a Gomory cut.