Course: Telecommunication Network Design
Teacher: Piotr Chołda piotr.cholda@agh.edu.pl
Studies: Electronics and Telecommunications
Speciality: Networks and Services
Semester: 2^{nd} sem. MSc stud., Fall

Draft of the lecture

Piotr Chołda

December 15, 2017

1 Basics of Complexity Theory and Solution Methods for Discrete Programming Problems

1.1 Fundamentals of complexity theory

1. Landau's notation, $\mathcal{O}$ (big-O).

2. Various types of complexity: polynomial, logarithmic, exponential.

3. Classes of problems:

   - deterministic polynomial $\mathcal{P}$ (polynomially solvable);
   - non-deterministic polynomial $\mathcal{NP}$ (polynomially verifiable);
   - $\mathcal{NP}$-complele ($\mathcal{NPC}$), polynomial reducability, the satisfiability problem, Cook-Levin's theorem;
   - $\mathcal{NP}$-hard;
   - problem $\mathcal{P} \stackrel{?}{=} \mathcal{NP}$.

1.2 Methods for solving discrete programming problems

4. Relaxation of an optimization problem, linear relaxation, properties and mutual relationships between solutions of MIP and its linear relaxation LR.

5. Solution of IP, MIP: branch-and-bound (B&B), usage of linear relaxations:

   - Branch and Bound for Binary variables:

     procedure BBB(N_U, N_0, N_1)
     ▷ We minimize the goal function
       `solve`($N_U, N_0, N_1, \mathbf{x}, \mathbf{u}, F^{LR}\{N_U, N_0, N_1\}$)
     ▷ $\mathbf{x}$ is an element of the solution space ($\mathbf{x} \in \mathbf{X}$)
     ▷ Solve $\mathbf{LR}\{N_U, N_0, N_1\}$
       if $N_U = \emptyset \quad || \quad \forall_{i \in N_U}\colon x_i \in \mathbb{B}$ (is binary) **then**
         if $F^{LR}\{N_U, N_0, N_1\} < F^{\text{best}}$ **then**
           $F^{\text{best}} \leftarrow F^{LR}\{N_U, N_0, N_1\}$
           $\mathbf{x}^{\text{best}} \leftarrow \mathbf{x}$
           $\mathbf{u}^{\text{best}} \leftarrow \mathbf{u}$
     ▷ The best solution so far has just been obtained: store it

        end if
    else
        ▷ In the current solution, there are some non-binary values of variables in N_U
        if $F^{LR}\{N_U, N_0, N_1\} \geq F^{\text{best}}$ **then**
            ▷ Bounding (relaxed solution is worse than the best obtained so far)
            return
        else
            Choose some $e \in N_U$, such that x_i is fractional
            $\text{BBB}(N_U - \{e\}, N_0 \cup \{e\}, N_1)$
            $\text{BBB}(N_U - \{e\}, N_0, N_1 \cup \{e\})$
            ▷ Branching (make some values fixed and then solve two subproblems)
        end if
    end if
end procedure

- General method of Branch and Bound for Integer variables:

procedure BBI(Ω)
    `solve`$(\Omega, F(\Omega), \mathbf{x}'(\Omega), \mathbf{x}''(\Omega))$
    ▷ Solve $\mathbf{P}(\Omega)$
    if `int`$(\mathbf{x}'(\Omega))$ **then**
        ▷ $\mathbf{x}'$ contains only integer components
        if $F(\Omega) < F^{\text{best}}$ **then**
            $F^{\text{best}} \leftarrow F(\Omega)$
            $\mathbf{x}^{\text{best}} \leftarrow (\mathbf{x}'(\Omega), \mathbf{x}''(\Omega))$
        end if
    else
        if $F(\Omega) \geq F^{\text{best}}$ **then**
            ▷ Bounding
            return
        else
            ▷ Branching
            Choose index j of one of the non-integer components of $\mathbf{x}'(\Omega)$
            $\text{BBI}\big((\Omega - \{d_j(\Omega) \leq x_j \leq g_j(\Omega)\}) \cup \{d_j(\Omega) \leq x_j \leq \lfloor x'_j(\Omega) \rfloor\}\big)$
            $\text{BBI}\big((\Omega - \{d_j(\Omega) \leq x_j \leq g_j(\Omega)\}) \cup \{\lceil x'_j(\Omega) \rceil \leq x_j \leq g_j(\Omega)\}\big)$
        end if
    end if
end procedure

6. Knapsack problem. Heuristic solution based on ratios of utility and weight.

1.3 Exercises

1. Present an outline (a short sketch) of an algorithm that finds shortest paths between all pairs of vertices in a weighted digraph (you can base on a known algorithm or use it as a subprocedure). Assess the computational complexity of the presented algorithm. The complexity is expressed with

usage of a variable that describes the size of the problem — here, the number of the digraph's vertices.

2. Solve the following problem, or at least give the range in which the optimal value of z is contained (the length of the range cannot exceed 10% of the optimum value):

   - max $z = 2x_1 + x_2 + 3x_3$
   - s.t.: $3x_1 + x_2 + 2x_3 \leq 12.5$
   - $x_i \in \mathbb{Z} \quad x_i \geq 0 \quad j = 1, 2, 3$

3. Using branch-and-bound (B&B) solve the following mathematical programming problem:

$$\begin{aligned} \text{min} \quad & x_1 - 2x_2 \\ \text{constraints:} \quad & -4x_1 + 6x_2 \leq 9 \\ & x_1 + x_2 \leq 4 \\ & x_1, x_2 \geq 0 \\ & x_1, x_2 \in \mathbb{Z} \end{aligned}$$

4. Using branch-and-bound (B&B) solve the following mathematical programming problem:

$$\begin{aligned} \text{max} \quad & 2x_1 + x_2 \\ \text{constraints:} \quad & -2x_1 - 2x_2 \leq -1 \\ & 2x_1 + 2x_2 \leq 5 \\ & -x_1 + 2x_2 \leq 2 \\ & 2x_1 \leq 3 \\ & x_1, x_2 \in \mathbb{Z}^+ \end{aligned}$$

1.4 Reading

1.4.1 Contents of the lecture

Problems described in this lecture are generally dealt with in the following positions:

- Michał Pióro and Deepankar Medhi. *Routing, Flow and Capacity Design in Communication and Computer Networks.* Morgan Kaufmann Publishers—Elsevier, San Francisco, CA, 2004: chapter 5.2, 6.4, appendix B.

- Poompat Saengudomlert. *Optimization for Communications and Networks.* CRC Press/Science Publishers, Boca Raton, FL, 2012: chapter 4.1-4.2.

1.4.2 Auxiliary references

- Jens Clausen. Branch and Bound Algorithms — Principles and Examples, March 1999. University of Copenhagen Technical Report: a description of various kinds of 'branch and bound' algorithms.

- Fedor V. Fomin and Petteri Kaski. Exact Exponential Algorithms. *Communications of the ACM*, 56(3):80–88, March 2013: solving difficult network problems.
- Eugene L. Lawler and David E. Wood. Branch-and-Bound Methods: A Survey. *Operations Research*, 14(4):699–719, July/August 1966: a seminal paper on algorithms of the 'branch and bound' kind.
- Laurence A. Wolsey. *Integer Programming.* John Wiley & Sons, Inc., New York, NY, 1998: a very good textbook on integer programming.