The University of Oxford

MSc (Mathematics and Foundations of Computer Science)

B6.3 Integer Programming

Michaelmas Term 2023

The steps of (each) mini project are for your guidance; if you wish to take an alternative route to the desired goal, you are free to do so. But, if you follow the suggested route and find yourself unable to carry out any particular step, you may simply assume it so that you can continue with the mini project, but should make this assumption clear in your presentation.

Part 1

Let $\mathcal{P} = \{x \in \mathbb{R}^n : \, Ax \leq b, x \geq 0\}$ be a polyhedron defined by $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$ chosen so that the row vectors of A are linearly independent and $\mathcal{P} \neq \emptyset$. Let (x, s) be a basic feasible solution of the system $Ax + s = b$, $x, s \geq 0$.

Task 1.a) Prove that x is an extreme point of $\mathcal{P}$.

Next, recall that in the course we studied the simplex algorithm for LPs in the form

$$\begin{aligned} \text{(P)} \quad & \max_x c^{\mathrm{T}} x \\ \text{subject to } & Ax = b, \\ & x \geq 0. \end{aligned}$$

Dividing the variables x_i $(i = 1, \ldots, n)$ into a basic block x_B and a non-basic block x_N, the system $Ax = b$ can be rewritten in block form

$$A_B x_B + A_N x_N = b.$$

With this notation the variant of simplex algorithm we use is as follows, where you should pay specific attention to the rules that specify the indices leaving N and entering B:

```
Algorithm (Simplex);
choose a basic feasible solution $(x_B, x_N)$;
while $c_N - A_N^{\mathrm{T}} A_B^{-T} c_B \not\leq 0$ do
    $i := \min\{\ell \in N : c_\ell > A_\ell^{\mathrm{T}} A_B^{-T} c_B\}$;
    // (smallest candidate index, where $A_\ell$ is the $\ell$-th
       column of $A$ indexed by re-ordered indices in $N$)
    if $A_B^{-1} A_i \leq 0$ then
        return '(P) unbounded';
    else
        $j :=$
            $\min\big\{\ell \in B : \ell \in \arg\min\big\{(A_B^{-1}b)_k/(A_B^{-1}A_i)_k : k \in B, (A_B^{-1}A_i)_k > 0\big\}\big\}$;
            // (smallest candidate index, where coefficients
            of $A_B^{-1}b$ indexed by re-ordered indices in $B$)
        $N \leftarrow N \cup \{j\} \setminus \{i\}$;
        $B \leftarrow B \cup \{i\} \setminus \{j\}$;
    end
end
return $(x_B, x_N) = (A_B^{-1}b, 0)$ as optimal basic solution;
```

Now consider the following LPs, where $0 < \epsilon \ll 1$.

$$(\mathrm{P}_n) \quad \max_{x \in \mathbb{R}^n} x_n$$

$$\text{subject to } \epsilon \leq x_1 \leq 1 - \epsilon, \tag{1}$$

$$\epsilon x_{i-1} \leq x_i \leq 1 - \epsilon x_{i-1}, \quad (i = 2, \ldots, n). \tag{2}$$

We introduce slack variables $x_{n+2(i-1)+1}$ and x_{n+2i} to rewrite the constraints as follows

$$\begin{aligned} \epsilon - x_1 + x_{n+1} &= 0, \\ \epsilon + x_1 + x_{n+2} &= 1, \\ \epsilon x_{i-1} - x_i + x_{n+2(i-1)+1} &= 0, \quad (i = 2, \ldots, n), \\ \epsilon x_{i-1} + x_i + x_{n+2i} &= 1, \quad (i = 2, \ldots, n). \end{aligned}$$

We use the basic feasible solution corresponding to the extreme point $x^{(0)} = (\epsilon, \epsilon^2, \ldots, \epsilon^n)$ as a starting point for Algorithm (Simplex) applied to problem (P_n), and we denote the basic feasible solution found after k iterations by

$x^{(k)}$. We also consider doing this for problem

$$(\mathrm{P}_{n-1}) \qquad \max_{y\in\mathbb{R}^{n-1}} y_{n-1}$$
$$\text{subject to } \epsilon y_{i-1} \le y_i \le 1 - \epsilon y_{i-1}, \quad (i = 1, \ldots, n-1),$$

where $y_0 := 1$ is a fixed input parameter, with an analogous indexing of slack variables, starting simplex iterations from the point $y^{(0)} = (\epsilon, \epsilon^2, \ldots, \epsilon^{n-1})$ and denoting the the basic feasible solution found after k iterations by $y^{(k)}$.

Task 1.b) Using induction or otherwise, prove that for $k = 0, \ldots, 2^{n-1} - 1$ the basic feasible solutions we find for the two problems satisfy

$$x_j^{(k)} = \begin{cases} y_j^{(k)} & \text{for } (j = 1, \ldots, n-1), \\ \epsilon y_{n-1}^{(k)} & \text{if } j = n, \end{cases} \tag{3}$$

$$x_j^{(2^{n-1}+k)} = \begin{cases} y_j^{(2^{n-1}-1-k)} & \text{for } (j = 1, \ldots, n-1), \\ 1 - \epsilon y_{n-1}^{(2^{n-1}-1-k)} & \text{if } j = n, \end{cases} \tag{4}$$

Conclude that Algorithm (Simplex) takes $2^n - 1$ simplex iterations to solve problem (P_n).

Task 1.c) (open ended): Consider changing the rule that determines the entering and leaving variables. Give some discussion of how this might affect the worst case complexity of the simplex algorithm.

Part 2

Consider the IP

$$
\begin{aligned}
\max_{x \in \mathbb{R}^2} \; & 2x_1 + x_2 \\
\text{subject to } & 7x_1 + x_2 \leq 28, && (5)\\
& -x_1 + 3x_2 \leq 7, && (6)\\
& -8x_1 - 9x_2 \leq -32, && (7)\\
& x_1, x_2 \geq 0,\\
& x_1, x_2 \in \mathbb{Z}.
\end{aligned}
$$

Task 2.a) Explain how Chvatal-Gomory cuts are generated from linear combinations of constraints (5), (6), (7) with non-negative linear weights $u_1, u_2, u_3 \geq 0$, and show that the choice $u_1 = 1/21, u_2 = 7/22, u_3 = 0$ yields the cutting plane $x_2 \leq 3$.

Task 2.b) Find a choice of $u_1, u_2, u_3 \geq 0$ that yield the cutting plane $-x_1 - x_2 \leq -4$.

Task 2.c) Prove or disprove the existence of $u_1, u_2, u_3 \geq 0$ that yield the valid inequality $-x_1 \leq -2$ as a cutting plane.

Next consider the following IP, where $k \in \mathbb{N}$,

$$
\begin{aligned}
(\mathrm{IP}_k) \quad \max_{x \in \mathbb{R}^2} \; & x_2 \\
\text{subject to } & 2kx_1 + x_2 \leq 2k\\
& -2kx_1 + x_2 \leq 0,\\
& x_1, x_2 \geq 0,\\
& x_1, x_2 \in \mathbb{Z},
\end{aligned}
$$

Task 2.d) Prove that it would take at least k successive Chvatal-Gomory cutting planes to reduce the LP-relaxation bound to the optimal value of problem (IP_k).

Part 3

Consider the 0-1 knapsack problem

$$
\begin{aligned}
\text{(K)} \quad z^* = \max_x & \sum_{j=1}^{n} c_j x_j \\
\text{subject to} & \sum_{j=1}^{n} a_j x_j \leq b, \\
& x_j \in \{0,1\}, \quad (j = 1, \ldots, n),
\end{aligned}
$$

where $a_j, c_j > 0$ $(j = 1, \ldots, n)$ and $b > 0$, and recall that the following greedy heuristic can be used to find primal bounds:

```
Algorithm (Greedy);
re-index variables such that $c_1/a_1 \geq c_2/a_2 \geq \cdots \geq c_n/a_n$;
set $z := 0$, $v := b$;
for $j=1,\ldots,n$ do
    if $a_j \leq v$ then
        $x_j = 1$;
        $v \leftarrow v - a_j$;
        $z = z + c_j$;
    else
        $x_j = 0$;
    end
end
```

We wish to solve problem (K) via LP-based branch-and-bound with fractional branching and greedy primal bounds generated by Algorithm (Greedy).

Task 3 (open ended): Try to construct and analyse problem inputs $a_j, c_j > 0$ $(j = 1, \ldots, n)$ and $b > 0$ which force the branch-and-bound algorithm to generate a number of nodes that approach a worst case upper bound as $n \to \infty$.