B6.3 Integer Programming
Problem Sheet 3

Prof. Raphael Hauser

Section A

Problem A.1: Solve the following IP by LP-based branch & bound,

$$\begin{aligned} \max\ & 9x_1 + 5x_2 \\ \text{s.t.}\ & 4x_1 + 9x_2 \leq 35 \\ & x_1 \leq 6 \\ & x_1 - 3x_2 \geq 1 \\ & 3x_1 + 2x_2 \leq 19 \\ & x \in \mathbb{Z}^2_+. \end{aligned}$$

Problem A.2: You are in the process of solving an integer linear maximisation problem in the variables x_1, x_2, x_3 via LP based branch-and-bound. The current value of the global primal bound is $\underline{z} = -\infty$, and the list of active subproblems is the following, where $\mathscr{P}_j$ is the formulation of the subproblem

$$(\mathcal{F}_j) \quad \max_{x \in \mathbb{R}^n} c^{\mathrm{T}} x$$
$$\text{subject to } x \in \mathscr{P}_j \cap \mathbb{Z},$$

$x_i^{[j]}$ are the optimal decision values of the LP-relaxation of the subproblem, and $\overline{z}^{[j]}$ is the objective value at $x^{[j]}$,

$\mathscr{P}_j$	$\overline{z}^{[j]}$	$x_1^{[j]}$	$x_2^{[j]}$	$x_3^{[j]}$
$\mathscr{P} \cap \{x_1 \geq 6,\ x_2 \leq 3\}$	90.5	6	3	0.5
$\mathscr{P} \cap \{x_1 \leq 5,\ x_2 \leq 13\}$	165.25	5	13	5.75
$\mathscr{P} \cap \{x_1 \leq 5,\ x_2 \geq 14,\ x_3 \geq 1\}$	138	4.25	16	1
$\mathscr{P} \cap \{x_1 \leq 5,\ x_2 \geq 14,\ x_3 \leq 0\}$	121.25	3.75	15.25	0

What is the current value of the global dual bound $\overline{z}$? Have we pruned any subproblem by optimality, by bound or by infeasibility yet? Explain your answer in each case.

Section B

Problem B.1: Consider the 0-1 knapsack problem $\max\{\sum_{j=1}^{n} c_j x_j \,:\, \sum_{j=1}^{n} a_j x_j \leq b,\, x \in \{0,1\}^n\}$, where $a_j, c_j > 0$ for $j = 1, \dots, n$.

i) Show that if $\frac{c_1}{a_1} \geq \cdots \geq \frac{c_n}{a_n} > 0$, $\sum_{j=1}^{r-1} a_j \leq b$ and $\sum_{j=1}^{r} a_j > b$, the solution of the LP relaxation is $x_j = 1$ for $j = 1, \dots, r-1$, $x_r = (b - \sum_{j=1}^{r-1} a_j)/a_r$ and $x_j = 0$ for $j > r$. [*Hint: First assume $x_r > 0$ and use complementary slackness to generate a certificate of optimality. Then extend the proof to the case $x_r = 0$.*]

ii) Solve the following instance by branch-and-bound,

$$\begin{aligned} &\max 17x_1 + 10x_2 + 25x_3 + 17x_4 \\ &\text{s.t. } 5x_1 + 3x_2 + 8x_3 + 7x_4 \leq 12 \\ &\quad x \in \{0,1\}^4. \end{aligned}$$

Problem B.2:

i) A relaxation of an integer programming problem (IP) $z = \max\{c^{\mathrm{T}}x \,:\, x \in \mathscr{F}\}$ is any optimisation problem (R) $w = \max\{g(x) \,:\, x \in \mathscr{R}\}$ with feasible set $\mathscr{R} \supseteq \mathscr{F}$ and an objective function $g(x)$ that satisfies $g(x) \geq c^{\mathrm{T}}x$ for all $x \in \mathscr{F}$. Show that if (R) is a relaxation of (IP), then $w \geq z$.

ii) Recall the symmetric travelling saleseman problem from Lecture 1,

$$\begin{aligned} \text{(TSP)} \quad z = \min &\sum_{e \in E} c_e x_e \\ \text{s.t.} &\sum_{e \in \delta(i)} x_e = 2 \quad (i \in V) \\ &\sum_{e \in E(S)} x_e \leq |S| - 1 \quad (S \subset V \text{ s.t. } 2 \leq |S| \leq |V| - 1) \\ &x \in \{0,1\}^{|E|}, \end{aligned}$$

where $G = (V, E)$ is an undirected graph (not necessarily assumed to be complete), $\delta(i) \subseteq E$ is the set of edges incident to node i, and $E(S)$ is the set of edges in E that are incident to two nodes from S. A 1-*tree* in $G = (V, E)$ is a subgraph that consists of the union of two edges adjacent to node 1 and a spanning tree on the remaining nodes. Show that the following minimum weight 1-tree problem is a relaxation of (TSP),

$$\begin{aligned} \text{(MW1T)} \quad w = \min &\sum_{e \in E} c_e x_e \\ \text{s.t. } &E(x) := \{e \in E \,:\, x_e = 1\} \text{ is a 1-tree,} \\ &x_e \in \{0,1\} \quad \forall\, e \in E. \end{aligned}$$

iii) Design an algorithm to solve (MW1T) and prove its correctness (prove that its output must be a minimum weight 1-tree of G.

iv) Design a branching rule that can be used in a minimum 1-tree relaxation based branch-and-bound algorithm to solve (TSP). [*Hint: remember that the subproblems must all be of the same type, i.e. TSPs that live in subgraphs of G.*]

Problem B.3 [B&P, long] An airline operates flights on a network of airports (nodes) $V = \{1, \dots, 4\}$. We focus on a set of five flight legs with following passenger capacities:

leg	connecting nodes	capacity
ℓ_1	$1 \to 2$	100
ℓ_2	$1 \to 3$	200
ℓ_3	$2 \to 4$	150
ℓ_4	$3 \to 2$	100
ℓ_5	$3 \to 4$	100

The departure and arrival times of the five legs are compatible with enabling passengers to take any of the following routes r_k specified below. Each route r_k is sold as two different products p_k^1, p_k^2 (economy and business class) at two different prices c_k^1, c_k^2:

route r_k	using legs	connecting nodes	c_k^1	c_k^2	d_k^1	d_k^2	d_k^3
r_1	ℓ_1	$1 \to 2$	250	600	40	20	15
r_2	ℓ_1, ℓ_3	$1 \to 2 \to 4$	300	620	45	25	20
r_3	ℓ_2	$1 \to 3$	130	400	40	15	15
r_4	ℓ_2, ℓ_5	$1 \to 3 \to 4$	170	520	50	20	30
r_5	ℓ_2, ℓ_4	$1 \to 3 \to 2$	150	300	45	15	15
r_6	ℓ_3	$2 \to 4$	180	250	50	30	10
r_7	ℓ_5	$3 \to 4$	160	500	40	10	35
r_8	ℓ_4, ℓ_3	$3 \to 2 \to 4$	160	400	45	20	20

For each k the demand of products p_k^1 and p_k^2 are governed by three integers $d_1, d_2, d_3 \in \mathbb{N}$, whereby d_1 is the unconstrained demand (the maximum number of tickets that could be sold if all flight legs had unlimited capacity) of the group of customers who are only interested in an economy ticket, d_2 is the unconstrained demand of a group of passengers who consider buying either economy or business but would prefer the cheaper ticket if on offer, and d_3 is the unconstrained demand of travellers who only consider business class travel. The uncapacitated demand of product p_k^1 is thus $d_1 + d_2$, while the demand $d_3 + d_2 \times (1 - x_k^1/(d_1 + d_2))$ of product p_k^2 depends on the number x_k^1 of tickets of product p_k^1 sold, assuming that a fraction $d_2/(d_1 + d_2)$ of the economy tickets are bought up by Group 2 (who cause demand displacement). The objective of the airline is to decide how many tickets x_k^i of each product p_k^i to sell so as to satisfy the capacity constraints on each flight leg and to maximise the total revenue.

i) Let $x_k = (x_k^1, x_k^2)$, $(k = 1, \dots, 8)$. Identify the blocks A_k in the joint constraints

$$\begin{bmatrix} A_1 & \dots & A_8 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_8 \end{bmatrix} \leq \begin{bmatrix} 100 \\ 200 \\ 150 \\ 100 \\ 100 \end{bmatrix}$$

given by the capacity constraints on $\ell_1, \dots, \ell_5$. Give a formulation of the sets $\mathscr{X}_k$ that contain the block vectors x_k that are compatible with the demand constraints. Formulate the revenue maximisation problem (RMP) of the airline as an IP in block-angular form.

ii) Formulate the Dantzig-Wolfe reformulation (IPM) of (RMP), the associated LP master problem (LPM) and its dual (DM). Discuss how you would solve the associated column generation subproblems (CGIP$_k$).

iii) Suppose we were instead interested in solving the LP relaxation (P) of (RMP). Discuss how to set up a Dantzig-Wolfe reformulation of (P) and the associated LP master problem (PM). Discuss the relationship between (PM) and problem (LPM) from part ii).

Section C

Problem C.1: Consider Example (Cutting Stock Problem) from the slides of Lecture 11, where we derived a dual bound of the root problem by delayed column generation and then used inspection to obtain an optimal solution by an ad hoc rounding procedure. Instead of using this inspection trick, complete the process of solving the root problem by branch & price.

Problem C.2: A relaxation of an integer programming problem (IP) $z = \max\{c^{\mathrm{T}}x : x \in \mathscr{F}\}$ is any optimisation problem (R) $w = \max\{g(x) : x \in \mathscr{R}\}$ with feasible set $\mathscr{R} \supseteq \mathscr{F}$ and an objective function $g(x)$ that satisfies $g(x) \geq c^{\mathrm{T}}x$ for all $x \in \mathscr{F}$.

i) Show that if (R) is a relaxation of (IP), then $w \geq z$.

ii) Consider the *equality knapsack problem*

$$\begin{aligned} \text{(EKP)} \quad \max_x \ & \sum_{j=1}^n c_j x_j \\ \text{s.t.} \ & \sum_{j=1}^n a_j x_j = b, \\ & x_j \in \mathbb{Z}_+, \quad (j = 1, \dots, n), \end{aligned}$$

where $b > 0$ is a positive integer, $a_j > 0$ are positive integers for all j, and $c_j > 0$ are positive reals. Let $k \in \arg\max\{c_j/a_j : j \in [1, n]\}$. Show that the following problem (called *group relaxation*) is a relaxation of (EKP),

$$\begin{aligned} \text{(GR)} \quad \frac{c_k}{a_k} b + \max_x \ & \sum_{j \neq k} \left(c_j - \frac{c_k}{a_k} a_j \right) x_j \\ \text{s.t.} \ & \sum_{j \neq k} a_j x_j \equiv b \, (\operatorname{mod} a_k), \\ & x_j \in \mathbb{Z}_+, \quad (j \neq k). \end{aligned}$$

iii) Now consider a network consisting of a digraph $G = (V, E)$ with vertices $V = \{0, 1, \dots, a_k - 1\}$ and edges

$$E = \{e_{i,j} := (i, s_{ij}) : i \in V, \, j \neq k, \, s_{ij} = i + a_j \, (\operatorname{mod} a_k)\}$$

and edge weights $C_{i,j} := -c_j + \frac{c_k}{a_k} a_j \geq 0$ associated with edge $e_{i,j}$, the non-negativity being guaranteed by our choice of k. Show that (GR) can be solved as a shortest path problem in this network from vertex 0 to vertex $b \, (\operatorname{mod} a_k)$. Work out the complexity when the Bellman-Ford Algorithm (the shortest path algorithm from Problem Sheet 2, Problem 5.ii) is applied for this purpose.

iv) Group relaxation can be used in a branch & bound algorithm to solve (EKP), with subproblems of the following form, where $\ell = (\ell_1, \dots, \ell_n)$ is a vector of non-negative integers,

$$\begin{aligned} \text{(EKP}(\ell)) \quad \max_x \ & \sum_{j=1}^n c_j x_j \\ \text{s.t.} \ & \sum_{j=1}^n a_j x_j = b, \\ & x_j \geq \ell_j, \quad (j = 1, \dots, n), \\ & x_j \in \mathbb{Z}, \quad (j = 1, \dots, n). \end{aligned}$$

Show how group relaxation can be applied to (EKP(ℓ)) and, starting with $\ell = 0$ for the root problem, derive a branching rule that allows one to branch each problem (EKP(ℓ)) into n subproblems of the same type (EKP($\ell^{[s]}$)), ($s = 1, \dots, n$).

v) Give a termination criterion and an upper bound on the number of subproblems that have to be solved before the criterion applies.

Problem C.3: Recall the branching rule in the branch-and-price algorithm for *binary* IPs (that is, $x_i^{k,t} \in \{0,1\}$ for all k,t,i) from the slides of Lecture 11: Let the vector $\tilde{x}^k = \sum_{t=1}^{T_k} \tilde{\lambda}_{k,t} x^{k,t}$ be constructed using the optimal weights $\tilde{\lambda}_{k,t}$ of the LP master problem.

i) Prove that $\tilde{x}^k$ is integer valued if and only if $\tilde{\lambda}$ is integer valued.

ii) Discuss how standard fractional branching can be used when applying the delayed column generation approach to LP based branch-and-bound of non-binary IPs, that is, when $x_i^{k,t}$ is merely constrained to lie in $\mathbb{Z}$ for all k,t,i. [*Hint: Show how the subproblems can be cast in block-angular form.*]

iii) Why is the branching method of the branch-and-price algorithm to be preferred when the IP is binary? Discuss how you would choose the variable to branch on when there are multiple indices κ for which $\tilde{x}^k$ is not integer valued.

Problem C.4: let $G = (V, E, c)$ be di-graph with vertex set V, edge set E and a set of edge weights c_e for each $e \in E$. We consider two origin-destination pairs (s,t) and (o,d) in $V \times V$ and the problem of finding two paths $p_{s\to t}$ from s to t and $p_{o\to d}$ from o to d such that $p_{s\to t} \cap p_{o\to d} = \emptyset$, that is, the set of edges occupied by the two paths must be non-intersecting, and so as to minimise the total length $\sum_{e\in p_{s\to t}\cup p_{o\to d}} c_e$. Let us explore how to solve this problem via branch & price:

i) Formulate this problem as a *block-angular* problem with two blocks, that is, as a problem in the format

$$
\begin{aligned}
\text{(IP)} \quad z = \max_{x=(x^1,x^2)} \; & c^{k\,\mathrm{T}} x^k + c^{k\,\mathrm{T}} x^k \\
\text{s.t. } & A^1 x^1 + A^2 x^2 = b \\
& x^k \in X^k = \{x^k \in \mathbb{Z}_+^{n_k} : D^k x^k \le d^k\}, \quad (k = 1, 2).
\end{aligned}
$$

[*Hint: Use our model of the shortest path problem to model the sets X^k and use determine the data A^1, A^2, b to model the condition that the two paths be disjoint.*]

ii) Set up the IP master problem (IPM), the LP master problem (LPM) and the dual master problem (DM).

iii) What do the patterns $x^{k,t}$ correspond to, and how would you choose a set of patterns to use in an initial restricted master problem (RM)?

iv) Set up the column generation problems (CGIP(k)). How would you solve these problems?

v) What is the practical interpretation of the branch & price branching rule. Discuss one of the branched sub-problems and show that it has the same block-angular structure as the root problem. Set up the column generation problems (CGIP(k)) for the subproblem and discuss how you would solve it.