DM204, 2010
SCHEDULING, TIMETABLING AND ROUTING

Lecture 27
Crew Scheduling and Column Generation

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

Course Overview

- ✔ Problem Introduction
  - ✔ Scheduling classification
  - ✔ Scheduling complexity
  - ✔ RCPSP
- ✔ General Methods
  - ✔ Integer Programming
  - ✔ Constraint Programming
  - ✔ Heuristics
  - ✔ Dynamic Programming
  - ✔ Branch and Bound
- ✔ Scheduling Models
  - ✔ Single Machine
  - ✔ Parallel Machine and Flow Shop
  - ✔ Job Shop
  - ✔ Resource-Constrained Project Scheduling
- Timetabling
  - ✔ Reservations and Education
  - ✔ Course Timetabling
  - ✔ Workforce Timetabling
  - Crew Scheduling
- Vehicle Routing
  - Capacited Models
  - Time Windows models
  - Rich Models

Outline

1. Crew Scheduling
2. Avanced Methods for IP
   - Dantzig-Wolfe Decomposition
   - Delayed Column Generation
   - Ryan's branching rule in Set Partitioning

Outline

1. Crew Scheduling
2. Avanced Methods for IP
   - Dantzig-Wolfe Decomposition
   - Delayed Column Generation
   - Ryan's branching rule in Set Partitioning

Crew Scheduling

Crew Scheduling
(sec. 12.6)

Input:

- A set of m flight legs (departure, arrival, duration)
- A set of crews
- A set of n (very large) feasible and permissible combinations of flights legs that a crew can handle (eg, round trips)
- A flight leg i can be part of more than one round trip
- Each round trip j has a cost c_j

Output: A set of round trips of mimimun total cost

Crew Scheduling
(sec. 12.6)

Input:

- A set of m flight legs (departure, arrival, duration)
- A set of crews
- A set of n (very large) feasible and permissible combinations of flights legs that a crew can handle (eg, round trips)
- A flight leg i can be part of more than one round trip
- Each round trip j has a cost c_j

Output: A set of round trips of mimimun total cost

Set partitioning problem:

$$\begin{aligned}
\min \quad & c_1x_1 + c_2x_2 + \ldots + c_nx_n \\
& a_{11}x_1 + a_{12}x_2 + \ldots + \ldots a_{1n}x_n = 1 \\
& a_{21}x_1 + a_{22}x_2 + \ldots + \ldots a_{2n}x_n = 1 \\
& \vdots \\
& a_{m1}x_1 + a_{m2}x_2 + \ldots + \ldots a_{mn}x_n = 1 \\
& x_j \in \{0,1\}, \qquad \forall j = 1, \ldots, n
\end{aligned}$$

Truck Routing
(sec. 12.6)

Input:

- Central depot and clients
- Single delivery to each client.
- Each truck can visit at most two costumers in each trip.

Output: Determine which truck should go to which client and the routing of trucks that minimize the total distance travelled.

Truck Routing
(sec. 12.6)

Input:

- Central depot and clients
- Single delivery to each client.
- Each truck can visit at most two costumers in each trip.

Output: Determine which truck should go to which client and the routing of trucks that minimize the total distance travelled.

$$
\begin{array}{ll}
\min & c_1x_1 + c_2x_2 + \ldots + c_nx_n \\
& a_{11}x_1 + a_{12}x_2 + \ldots + \ldots a_{1n}x_n = 1 \\
& a_{21}x_1 + a_{22}x_2 + \ldots + \ldots a_{2n}x_n = 1 \\
& \vdots \\
& a_{m1}x_1 + a_{m2}x_2 + \ldots + \ldots a_{mn}x_n = 1 \\
& x_j \in \{0,1\}, \qquad \forall j = 1, \ldots, n
\end{array}
$$

Route	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
c_j	8	10	4	4	2	14	10	8	8	10	11	12	6	6	5
	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0
	0	1	0	0	0	1	0	0	0	1	1	1	0	0	0
	0	0	1	0	0	0	1	0	0	1	0	0	1	1	0
	0	0	0	1	0	0	0	1	0	0	1	0	1	0	1
	0	0	0	0	1	0	0	0	1	0	0	1	0	1	1

Set partitioning or set covering??

Set partitioning or set covering??

Often treated as set covering because:

- its linear programming relaxation is numerically more stable and thus easier to solve
- it is trivial to construct a feasible integer solution from a solution to the linear programming relaxation
- it makes it possible to restrict to only rosters of maximal length

Tanker Scheduling
(sec. 11.2)

Input:

- p ports

  limits on the physical characteristics of the ships

- n cargoes:

  type, quantity, load port, delivery port, time window constraints on the load and delivery times

- ships (tanker): s company-owned plus others chartered
  Each ship has a capacity, draught, speed, fuel consumption, starting location and times

  These determine the costs of a shipment: c_i^l (company-owned) c_j^* (chartered)

Tanker Scheduling
(sec. 11.2)

Input:

- p ports

  limits on the physical characteristics of the ships

- n cargoes:

  type, quantity, load port, delivery port, time window constraints on the load and delivery times

- ships (tanker): s company-owned plus others chartered
  Each ship has a capacity, draught, speed, fuel consumption, starting location and times

  These determine the costs of a shipment: c_i^l (company-owned) c_j^* (chartered)

Output: A schedule for each ship, that is, an itinerary listing the ports visited and the time of entry in each port within the rolling horizon such that the total cost of transportation is minimized

Two phase approach:

1. determine for each ship i the set S_i of all possible itineraries
2. select the itineraries for the ships by solving an IP problem

Two phase approach:

1. determine for each ship i the set S_i of all possible itineraries
2. select the itineraries for the ships by solving an IP problem

Phase 1 can be solved by some ad-hoc enumeration or heuristic algorithm that checks the feasibility of the itinerary and its cost.
Phase 2 Set packing problem with additional constraints (next slide)

For each itinerary l of ship i compute the profit with respect to charter:

$$\pi_i^l = \sum_{j=1}^{n} a_{ij}^l c_j^* - c_i^l$$

where $a_{ij}^l = 1$ if cargo j is shipped by ship i in itinerary l and 0 otherwise.

A set packing model with additional constraints

Variables

$$x_i^l \in \{0,1\} \qquad \forall i = 1,\dots,s;\ l \in S_i$$

Each cargo is assigned to at most one ship:

$$\sum_{i=1}^{s} \sum_{l \in S_i} a_{ij}^l x_i^l \le 1 \qquad \forall j = 1,\dots,n$$

Each tanker can be assigned at most one itinerary

$$\sum_{l \in S_i} x_i^l \le 1 \qquad \forall i = 1,\dots,s$$

Objective: maximize profit

$$\max \sum_{i=1}^{s} \sum_{l \in S_i} \pi_i^l x_i^l$$

Daily Aircraft Routing and Scheduling (Sec. 11.3)

[Desaulniers, Desrosiers, Dumas, Solomon and Soumis, 1997]

Input:

- L set of flight legs with airport of origin and arrival, departure time windows $[e_i, l_i]$, $i \in L$, duration, cost/revenue
- Heterogeneous aircraft fleet T, with m_t aircrafts of type $t \in T$

Daily Aircraft Routing and Scheduling (Sec. 11.3)

[Desaulniers, Desrosiers, Dumas, Solomon and Soumis, 1997]

Input:

- L set of flight legs with airport of origin and arrival, departure time windows $[e_i, l_i]$, $i \in L$, duration, cost/revenue
- Heterogeneous aircraft fleet T, with m_t aircrafts of type $t \in T$

Output: For each aircraft, a sequence of operational flight legs and departure times such that operational constraints are satisfied:

- number of planes for each type
- restrictions on certain aircraft types at certain times and certain airports
- required connections between flight legs (thrus)
- limits on daily traffic at certain airports
- balance of airplane types at each airport

and the total profits are maximized.

- L_t denotes the set of flights that can be flown by aircraft of type t
- S_t the set of feasible schedules for an aircraft of type t (inclusive of the empty set)
- $a^l_{ti} = \{0, 1\}$ indicates if leg i is covered by $l \in S_t$
- π_{ti} profit of covering leg i with aircraft of type i

$$\pi^l_t = \sum_{i \in L_t} \pi_{ti} a^l_{ti} \qquad \text{for } l \in S_t$$

- P set of airports, P_t set of airports that can accommodate type t
- o^l_{tp} and d^l_{tp} equal to 1 if schedule l, $l \in S_t$ starts and ends, resp., at airport p

A set partitioning model with additional constraints

Variables

$$x^l_t \in \{0, 1\} \qquad \forall t \in T;\ l \in S_t \qquad \text{and} \qquad x^0_t \in \mathbb{N} \qquad \forall t \in T$$

Maximum number of aircraft of each type:

$$\sum_{l \in S_t} x^l_t = m_t \qquad \forall t \in T$$

Each flight leg is covered exactly once:

$$\sum_{t \in T} \sum_{l \in S_t} a^l_{ti} x^l_t = 1 \qquad \forall i \in L$$

Flow conservation at the beginning and end of day for each aircraft type

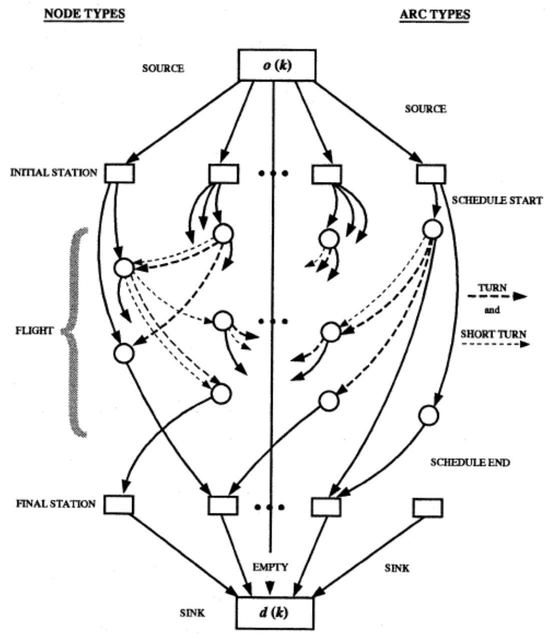
$$\sum_{l \in S_t} (o^l_{tp} - d^l_{tp}) x^l_t = 0 \qquad \forall t \in T;\ p \in P$$

Maximize total anticipate profit

$$\max \sum_{t \in T} \sum_{l \in S_t} \pi^l_t x^l_t$$

Solution Strategy: branch-and-price

- At the high level branch-and-bound similar to the Tanker Scheduling case
- Upper bounds obtained solving linear relaxations by column generation.
  - Decomposition into
    - Restricted Master problem, defined over a restricted number of schedules
    - Subproblem, used to test the optimality or to find a new feasible schedule to add to the master problem (column generation)
  - Each restricted master problem solved by LP.
    It finds current optimal solution and dual variables
  - Subproblem (or pricing problem) corresponds to finding longest path with time windows in a network defined by using dual variables of the current optimal solution of the master problem. Solve by dynamic programming.



$$\text{Maximize} \sum_{k\in K}\sum_{(i,j)\in A^k} c_{ij}^k X_{ij}^k \tag{8}$$

subject to:

$$\sum_{k\in K}\sum_{j:(i,j)\in A^k} X_{ij}^k = 1 \quad \forall i \in N, \tag{9}$$

$$\sum_{i:(i,s)\in NS_2^k} X_{is}^k - \sum_{j:(s,j)\in S_1N^k} X_{sj}^k = 0 \quad \forall k \in K, \forall s \in S^k, \tag{10}$$

$$\sum_{s\in S_1^k} X_{o(k),s}^k + X_{o(k),d(k)}^k = n^k \quad \forall k \in K, \tag{11}$$

$$\sum_{i:(i,j)\in A^k} X_{ij}^k - \sum_{i:(j,i)\in A^k} X_{ji}^k = 0$$

$$\forall k \in K, \forall j \in V^k \backslash \{o(k), d(k)\}, \tag{12}$$

$$\sum_{s\in S_2^k} X_{s,d(k)}^k + X_{o(k),d(k)}^k = n^k \quad \forall k \in K, \tag{13}$$

$$X_{ij}^k \geq 0 \quad \forall k \in K, \forall (i,j) \in A^k, \tag{14}$$

$$a_i^k \leq T_i^k \leq b_i^k \quad \forall k \in K, \forall i \in V^k, \tag{15}$$

$$X_{ij}^k (T_i^k + d_{ij}^k - T_j^k) \leq 0 \quad \forall k \in K, \forall (i,j) \in A^k, \tag{16}$$

$$X_{ij}^k \text{ integer} \quad \forall k \in K, \forall (i,j) \in A^k. \tag{17}$$

OR in Air Transport Industry

- Aircraft and Crew Schedule Planning
  - Schedule Design (specifies legs and times)
  - Fleet Assignment
  - Aircraft Maintenance Routing
  - Crew Scheduling
    - crew pairing problem
    - crew assignment problem (bidlines)
- Airline Revenue Management
  - number of seats available at fare level
  - overbooking
  - fare class mix (nested booking limits)
- Aviation Infrastructure
  - airports
    - runaways scheduling (queue models, simulation; dispatching, optimization)
    - gate assignments
  - air traffic management

Outline

1. Crew Scheduling

2. Avanced Methods for IP
   - Dantzig-Wolfe Decomposition
   - Delayed Column Generation
   - Ryan's branching rule in Set Partitioning

Outline

1. Crew Scheduling

2. Avanced Methods for IP
   - Dantzig-Wolfe Decomposition
   - Delayed Column Generation
   - Ryan's branching rule in Set Partitioning

Dantzig-Wolfe Decomposition

Motivation: Large difficult IP models

➨ split them up into smaller pieces

Dantzig-Wolfe Decomposition

Motivation: Large difficult IP models

➨ split them up into smaller pieces

Applications

- Cutting Stock problems
- Multicommodity Flow problems
- Facility Location problems
- Capacitated Multi-item Lot-sizing problem
- Air-crew and Manpower Scheduling
- Vehicle Routing Problems
- Scheduling (current research)

Dantzig-Wolfe Decomposition

Motivation: Large difficult IP models

➨ split them up into smaller pieces

Applications

- Cutting Stock problems
- Multicommodity Flow problems
- Facility Location problems
- Capacitated Multi-item Lot-sizing problem
- Air-crew and Manpower Scheduling
- Vehicle Routing Problems
- Scheduling (current research)

Leads to methods also known as:

- Branch-and-price (column generation + branch and bound)
- Branch-and-cut-and-price (column generation + branch and bound + cutting planes)

Dantzig-Wolfe Decomposition

The problem is split into a master problem and a subproblem

+ Tighter bounds
+ Better control of subproblem
– Model may become (very) large

Delayed column generation

Write up the decomposed model gradually as needed

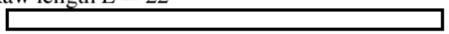
- Generate a few solutions to the subproblems
- Solve the master problem to LP-optimality
- Use the dual information to find most promising solutions to the subproblem
- Extend the master problem with the new subproblem solutions.

Motivation: Cutting stock problem

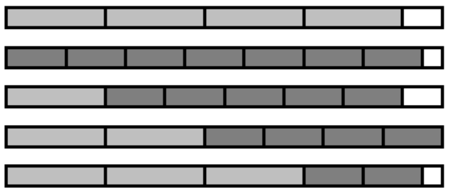
- Infinite number of raw stocks, having length L.
- Cut m piece types i, each having width w_i and demand b_i.
- Satisfy demands using least possible raw stocks.

Example:

- $w_1 = 5, b_1 = 7$ 
- $w_2 = 3, b_2 = 3$ 
- Raw length $L = 22$



Some possible cuts



Formulation 1

$$\begin{array}{ll} \text{minimize} & u_1 + u_2 + u_3 + u_4 + u_5 \\ \text{subject to} & 5x_{11} + 3x_{12} \leq 22u_1 \\ & 5x_{21} + 3x_{22} \leq 22u_2 \\ & 5x_{31} + 3x_{32} \leq 22u_3 \\ & 5x_{41} + 3x_{42} \leq 22u_4 \\ & 5x_{51} + 3x_{52} \leq 22u_5 \\ & x_{11} + x_{21} + x_{31} + x_{41} + x_{51} \geq 7 \\ & x_{12} + x_{22} + x_{32} + x_{42} + x_{52} \geq 3 \\ & u_j \in \{0,1\} \\ & x_{ij} \in \mathbb{Z}_+ \end{array}$$

LP-relaxation gives solution value $z = 2$ with

$$u_1 = u_2 = 1, x_{11} = 2.6, x_{12} = 3, x_{21} = 4.4$$

Block structure

$$\begin{array}{llllllll} \text{min} & & u_1 & +u_2 & +u_3 & +u_4 & +u_5 & \\ \text{s.t} & x_{11} & & +x_{21} & +x_{31} & +x_{41} & +x_{51} & \geq 7 \\ & x_{12} & & +x_{22} & +x_{32} & +x_{42} & +x_{51} & \geq 3 \\ & 5x_{11} + 3x_{12} - 22u_1 & & & & & & \leq 0 \\ & & & 5x_{21} + 3x_{22} - 22u_2 & & & & \leq 0 \\ & & & & 5x_{31} + 3x_{32} - 22u_3 & & & \leq 0 \\ & & & & & 5x_{41} + 3x_{42} - 22u_4 & & \leq 0 \\ & & & & & & 5x_{51} + 3x_{52} - 22u_5 & \leq 0 \end{array}$$

Formulation 2

The matrix A contains all different cutting patterns
All (undominated) patterns:

$$A = \begin{pmatrix} 4 & 0 & 1 & 2 & 3 \\ 0 & 7 & 5 & 4 & 2 \end{pmatrix}$$

Problem

$$\begin{aligned} \text{minimize } & \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 \\ \text{subject to } & 4\lambda_1 + 0\lambda_2 + 1\lambda_3 + 2\lambda_4 + 3\lambda_5 \geq 7 \\ & 0\lambda_1 + 7\lambda_2 + 5\lambda_3 + 4\lambda_4 + 2\lambda_5 \geq 3 \\ & \lambda_j \in \mathbb{Z}_+ \end{aligned}$$

LP-relaxation gives solution value $z = 2.125$ with

$$\lambda_1 = 1.375, \lambda_4 = 0.75$$

Due to integer property a lower bound is $\lceil 2.125 \rceil = 3$.
Optimal solution value is $z^* = 3$.

Round up LP-solution getting heuristic solution $z_H = 3$.

Dantzig-Wolfe decomposition

If model has "block" structure

$$\begin{array}{lcccccccl} \max & c^1x^1 & + & c^2x^2 & + \dots + & c^Kx^K & & \\ \text{s.t.} & A^1x^1 & + & A^2x^2 & + \dots + & A^Kx^K & = & b \\ & D^1x^1 & + & & & & \leq & d_1 \\ & & + & D^2x^2 & & & \leq & d_2 \\ & & & & \dots & & \leq & \vdots \\ & & & & & D^Kx^K & \leq & d_K \\ & x^1 \in \mathbb{Z}_+^{n_1} & & x^2 \in \mathbb{Z}_+^{n_2} & \dots & x^K \in \mathbb{Z}_+^{n_K} & & \end{array}$$

Describe each set X^k, $k = 1, \dots, K$

$$\begin{array}{lcccccccl} \max & c^1x^1 & + & c^2x^2 & + \dots + & c^Kx^K & & \\ \text{s.t.} & A^1x^1 & + & A^2x^2 & + \dots + & A^Kx^K & = & b \\ & x^1 \in X^1 & & x^2 \in X^2 & \dots & x^K \in X^K & & \end{array}$$

where $X^k = \{x^k \in \mathbb{Z}_+^{n_k} : D^kx^k \leq d_k\}$

Assuming that X^k has finite number of points $\{x^{k,t}\}$ $t \in T_k$

$$X^k = \left\{ \begin{array}{rl} x^k \in \mathbb{R}^{n_k} : & x^k = \sum_{t \in T_k} \lambda_{k,t} x^{k,t}, \\ & \sum_{t \in T_k} \lambda_{k,t} = 1, \\ & \lambda_{k,t} \in \{0,1\}, t \in T_k \end{array} \right\}$$

Decomposition

If model has "block" structure

$$\begin{array}{lcccccccl} \max & c^1x^1 & + & c^2x^2 & + \dots + & c^Kx^K & & \\ \text{s.t.} & A^1x^1 & + & A^2x^2 & + \dots + & A^Kx^K & = & b \\ & D^1x^1 & + & & & & \leq & d_1 \\ & & + & D^2x^2 & & & \leq & d_2 \\ & & & & \dots & & \leq & \vdots \\ & & & & & D^Kx^K & \leq & d_K \\ & x^1 \in \mathbb{Z}_+^{n_1} & & x^2 \in \mathbb{Z}_+^{n_2} & \dots & x^K \in \mathbb{Z}_+^{n_K} & & \end{array}$$

Lagrangian relaxation

Objective becomes

$$\begin{aligned} & c^1x^1 + c^2x^2 + \dots + c^Kx^K \\ & -\lambda \left(A^1x^1 + A^2x^2 + \dots + A^Kx^K - b\right) \end{aligned}$$

Decomposed into

$$\begin{array}{lccccl} \max c^1x^1 - \lambda A^1x^1 + c^2x^2 - \lambda A^2x^2 + \dots + c^Kx^K - \lambda A^Kx^K + b & & & & & \\ \text{s.t.} \quad D^1x^1 \quad + & & & & & \leq d_1 \\ \quad + \quad D^2x^2 & & & & & \leq d_2 \\ \quad \dots & & & & & \leq \vdots \\ \quad D^Kx^K & & & & & \leq d_K \\ x^1 \in \mathbb{Z}_+^{n_1} \quad x^2 \in \mathbb{Z}_+^{n_2} \quad \dots \quad x^K \in \mathbb{Z}_+^{n_K} & & & & & \end{array}$$

Model is separable

Dantzig-Wolfe decomposition

Substituting X^k in original model getting *Master Problem*

$$\max c^1 \Big(\sum_{t \in T_1} \lambda_{1,t} x^{1,t}\Big) + c^2 \Big(\sum_{t \in T_2} \lambda_{2,t} x^{2,t}\Big) + \dots + c^K \Big(\sum_{t \in T_K} \lambda_{K,t} x^{K,t}\Big)$$

$$\text{s.t. } A^1 \Big(\sum_{t \in T_1} \lambda_{1,t} x^{1,t}\Big) + A^2 \Big(\sum_{t \in T_2} \lambda_{2,t} x^{2,t}\Big) + \dots + A^K \Big(\sum_{t \in T_K} \lambda_{K,t} x^{K,t}\Big) = b$$

$$\sum_{t \in T_k} \lambda_{k,t} = 1 \qquad k = 1, \dots, K$$

$$\lambda_{k,t} \in \{0,1\}, \qquad t \in T_k \quad k = 1, \dots, K$$

Strength of linear master model

Solving LP-relaxation of master problem, is equivalent to (Wolsey Prop 11.1)

$$
\begin{array}{lccccccc}
\max & c^1x^1 & + & c^2x^2 & +\ldots+ & c^kx^k & & \\
\text{s.t.} & A^1x^1 & + & A^2x^2 & +\ldots+ & A^kx^k & = b \\
 & x^1 \in \text{conv}(X^1) & & x^2 \in \text{conv}(X^2) & \ldots & x^k \in \text{conv}(X^k) & &
\end{array}
$$

Proof: Consider LP-relaxation

$$\max c^1(\sum_{t\in T_1} \lambda_{1,t}x^{1,t}) + c^2(\sum_{t\in T_2} \lambda_{2,t}x^{2,t}) + \ldots + c^K(\sum_{t\in T_K} \lambda_{K,t}x^{K,t})$$

$$\text{s.t. } A^1(\sum_{t\in T_1} \lambda_{1,t}x^{1,t}) + A^2(\sum_{t\in T_2} \lambda_{2,t}x^{2,t}) + \ldots + A^K(\sum_{t\in T_K} \lambda_{K,t}x^{K,t}) = b$$

$$
\begin{array}{lll}
\sum_{t\in T_k} \lambda_{k,t} = 1 & & k = 1,\ldots,K \\
\lambda_{k,t} \geq 0, & t \in T_k & k = 1,\ldots,K
\end{array}
$$

Informally speaking we have

- joint constraint is solved to LP-optimality
- block constraints are solved to IP-optimality

Strength of Lagrangian relaxation

- z^{LPM} be LP-solution value of master problem
- z^{LD} be solution value of lagrangian dual problem

(Theorem 11.2)

$$z^{LPM} = z^{LD}$$

Proof: Lagrangian relaxing joint constraint in

$$
\begin{array}{lccccccc}
\max & c^1x^1 & + & c^2x^2 & +\ldots+ & c^Kx^K & & \\
\text{s.t.} & A^1x^1 & + & A^2x^2 & +\ldots+ & A^Kx^K & = & b \\
 & D^1x^1 & + & & & & \leq & d_1 \\
 & & + & D^2x^2 & & & \leq & d_2 \\
 & & & & \ldots & & \leq & \vdots \\
 & & & & & D^Kx^K & \leq & d_K \\
 & x^1 \in \mathbb{Z}_+^{n_1} & & x^2 \in \mathbb{Z}_+^{n_2} & \ldots & x^K \in \mathbb{Z}_+^{n_K} & &
\end{array}
$$

Using result next page

$$
\begin{array}{lccccccc}
\max & c^1x^1 & + & c^2x^2 & +\ldots+ & c^kx^k & & \\
\text{s.t.} & A^1x^1 & + & A^2x^2 & +\ldots+ & A^kx^k & = b \\
 & x^1 \in \text{conv}(X^1) & & x^2 \in \text{conv}(X^2) & \ldots & x^k \in \text{conv}(X^k) & &
\end{array}
$$

Strength of Lagrangian Relaxation (section 10.2)

Integer Programming Problem

$$
\begin{array}{ll}
\text{maximize} & cx \\
\text{subject to} & Ax \leq b \\
 & Dx \leq d \\
 & x_j \in \mathbb{Z}_+, \quad j = 1,\ldots,n
\end{array}
$$

Lagrange Relaxation, multipliers $\lambda \geq 0$

$$
\begin{array}{ll}
\text{maximize} & z_{LR}(\lambda) = cx - \lambda(Dx - d) \\
\text{subject to} & Ax \leq b \\
 & x_j \in \mathbb{Z}_+, \quad j = 1,\ldots,n
\end{array}
$$

for best multiplier $\lambda \geq 0$

$$\boxed{\max\left\{cx : Dx \leq d, x \in \text{conv}(Ax \leq b, x \in \mathbb{Z}_+)\right\}}$$

Outline

Delayed Column Generation

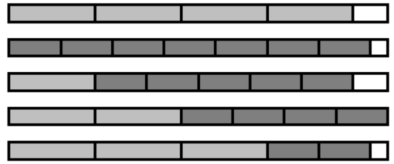
Delayed column generation, linear master

- Master problem can (and will) contain many columns
- To find bound, solve LP-relaxation of master
- Delayed column generation gradually writes up master

Delayed column generation, linear master

- $w_1 = 5, b_1 = 7$
- $w_2 = 3, b_2 = 3$
- Raw length $L = 22$

Some possible cuts



In matrix form

$$A = \begin{pmatrix} 4 & 0 & 1 & 2 & 3 & \cdots \\ 0 & 7 & 5 & 4 & 2 & \cdots \end{pmatrix}$$

LP-problem

$$\begin{aligned} \min \quad & cx \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

where

- $b = (7, 3)$,
- $x = (x_1, x_2, x_3, x_4, x_5, \cdots)$
- $c = (1, 1, 1, 1, 1, \cdots)$.

Reduced Costs

Simplex in matrix form

$$\min \{cx \mid Ax = b, x \geq 0\}$$

In matrix form:

$$\begin{bmatrix} A & 0 \\ c & -1 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}$$

- $\mathcal{B} = \{1, 2, \ldots, p\}$ basic variables
- $\mathcal{L} = \{1, 2, \ldots, q\}$ non-basis variables (will be set to lower bound $= 0$)
- $(\mathcal{B}, \mathcal{L})$ basis structure
- $x_\mathcal{B}, x_\mathcal{L}, c_\mathcal{B}, c_\mathcal{L}$
- $B = [A_1, A_2, \ldots, A_p]$, $L = [A_{p+1}, A_{p+2}, \ldots, A_{p+q}]$

$$\begin{bmatrix} B & L & 0 \\ c_\mathcal{B} & c_\mathcal{L} & -1 \end{bmatrix} \begin{bmatrix} x_\mathcal{B} \\ x_\mathcal{L} \\ z \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}$$

Reduced Costs

Crew Scheduling
Avanced Methods for IP
Dantzig-Wolfe Decompos
Delayed Column Generatic
Ryan's branching rule

Simplex in matrix form

$$\min \{cx \mid Ax = b, x \geq 0\}$$

In matrix form:

$$\begin{bmatrix} A & 0 \\ c & -1 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}$$

- $\mathcal{B} = \{1, 2, \ldots, p\}$ basic variables
- $\mathcal{L} = \{1, 2, \ldots, q\}$ non-basis variables (will be set to lower bound $= 0$)
- $(\mathcal{B}, \mathcal{L})$ basis structure
- $x_\mathcal{B}, x_\mathcal{L}, c_\mathcal{B}, c_\mathcal{L}$
- $B = [A_1, A_2, \ldots, A_p]$, $L = [A_{p+1}, A_{p+2}, \ldots, A_{p+q}]$

$$\begin{bmatrix} B & L & 0 \\ c_\mathcal{B} & c_\mathcal{L} & -1 \end{bmatrix} \begin{bmatrix} x_\mathcal{B} \\ x_\mathcal{L} \\ z \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}$$

Marco Chiarandini .::. 35

$$Bx_{\mathcal{B}} + Lx_{\mathcal{L}} = b \quad \Rightarrow \quad x_{\mathcal{B}} + B^{-1}Lx_{\mathcal{L}} = B^{-1}b \quad \Rightarrow \quad \left[\begin{array}{l} x_{\mathcal{L}} = 0 \\ x_{\mathcal{B}} = B^{-1}b \end{array} \right.$$

Simplex algorithm sets $x_{\mathcal{L}} = 0$ and $x_{\mathcal{B}} = B^{-1}b$
B invertible, hence rows linearly independent

The objective function is obtained by multiplying and subtracting constraints by means of multipliers π (the dual variables)

$$z = \sum_{j=1}^{p} \left[c_j - \sum_{i=1}^{p} \pi_i a_{ij} \right] x_j + \sum_{j=p+1}^{p+q} \left[c_j - \sum_{i=1}^{p} \pi_i a_{ij} \right] x_j + \sum_{i=1}^{p} \pi_i b_i$$

Each basic variable has cost null in the objective function

$$c_j - \sum_{i=1}^{p} \pi_i a_{ij} = 0 \qquad \Longrightarrow \qquad \pi = B^{-1} c_{\mathcal{B}}$$

Reduced costs of non-basic variables:

$$c_j - \sum_{i=1}^{p} \pi_i a_{ij}$$

Delayed column generation (example)

- $w_1 = 5, b_1 = 7$ 
- $w_2 = 3, b_2 = 3$ 
- Raw length $L = 22$

Initially we choose only the trivial cutting patterns

$$A = \begin{pmatrix} 4 & 0 \\ 0 & 7 \end{pmatrix}$$

Solve LP-problem

$$\begin{array}{ll} \min & cx \\ \text{s.t.} & Ax = b \\ & x \geq 0 \end{array}$$

i.e.

$$\begin{pmatrix} 4 & 0 \\ 0 & 7 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 7 \\ 3 \end{pmatrix}$$

with solution $x_1 = \frac{7}{4}$ and $x_2 = \frac{3}{7}$.
The dual variables are $y = c_B A_B^{-1}$ i.e.

$$\begin{pmatrix} 1 & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{4} & 0 \\ 0 & \frac{1}{7} \end{pmatrix} = \begin{pmatrix} \frac{1}{4} \\ \frac{1}{7} \end{pmatrix}$$

Small example (continued)

Find entering variable

$$\begin{array}{lcl} A & = & \begin{pmatrix} 1 & 2 & 3 & \cdots \\ 5 & 4 & 2 & \cdots \end{pmatrix} \qquad \begin{array}{l} \frac{1}{4} \leftarrow y_1 \\ \frac{1}{7} \leftarrow y_2 \end{array} \\ c_N - yA_N & = & \left(1 - \frac{27}{28} \;\; 1 - \frac{30}{28} \;\; 1 - \frac{29}{28} \;\; \cdots \right) \end{array}$$

We could also solve optimization problem

$$\begin{array}{ll} \min & 1 - \frac{1}{4}x_1 - \frac{1}{7}x_2 \\ \text{s.t.} & 5x_1 + 3x_2 \leq 22 \\ & x \geq 0, \text{integer} \end{array}$$

which is equivalent to knapsack problem

$$\begin{array}{ll} \max & \frac{1}{4}x_1 + \frac{1}{7}x_2 \\ \text{s.t.} & 5x_1 + 3x_2 \leq 22 \\ & x \geq 0, \text{integer} \end{array}$$

This problem has optimal solution $x_1 = 2, x_2 = 4$.
Reduced cost of entering variable

$$1 - 2\frac{1}{4} - 4\frac{1}{7} = 1 - \frac{30}{28} = -\frac{1}{14} < 0$$

Small example (continued)

Add new cutting pattern to A getting

$$A = \begin{pmatrix} 4 & 0 & 3 \\ 0 & 7 & 2 \end{pmatrix}$$

Solve problem to LP-optimality, getting primal solution

$$x_1 = \frac{5}{8}, x_3 = \frac{3}{2}$$

and dual variables

$$y_1 = \frac{1}{4}, y_2 = \frac{1}{8}$$

Note, we do not need to care about "leaving variable"
To find entering variable, solve

$$\begin{array}{ll} \max & \frac{1}{4}x_1 + \frac{1}{8}x_2 \\ \text{s.t.} & 5x_1 + 3x_2 \leq 22 \\ & x \geq 0, \text{integer} \end{array}$$

This problem has optimal solution $x_1 = 4, x_2 = 0$.
Reduced cost of entering variable

$$1 - 4\frac{1}{4} - 0\frac{1}{7} = 0$$

Terminate with $x_1 = \frac{5}{8}$, $x_3 = \frac{3}{2}$, and $z_{LP} = \frac{17}{8} = 2.125$.

Questions

- Will the process terminate?

  Always improving objective value. Only a finite number of basis solutions.

- Can we repeat the same pattern?

  No, since the objective functions is improved. We know the best solution among existing columns. If we generate an already existing column, then we will not improve the objective.

Branch-and-Price

Terminology

- Master Problem
- Restricted Master Problem
- Subproblem or Pricing Problem
- Branch and cut:
  Branch-and-bound algorithm using cuts to strengthen bounds.
- Branch and price:
  Branch-and-bound algorithm using column generation to derive bounds.

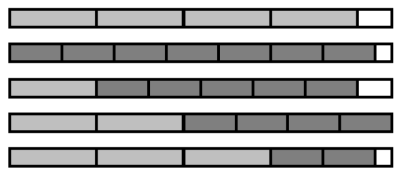
Branch-and-price

- LP-solution of master problem may have fractional solutions
- Branch-and-bound for getting IP-solution
- In each node solve LP-relaxation of master
- Subproblem may change when we add constraints to master problem
- Branching strategy should make subproblem easy to solve

Branch-and-price, example

The matrix A contains all different cutting patterns

$$A = \begin{pmatrix} 4 & 0 & 1 & 2 & 3 \\ 0 & 7 & 5 & 4 & 2 \end{pmatrix}$$



Problem

$$\begin{aligned} \text{minimize } & \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 \\ \text{subject to } & 4\lambda_1 + 0\lambda_2 + 1\lambda_3 + 2\lambda_4 + 3\lambda_5 \geq 7 \\ & 0\lambda_1 + 7\lambda_2 + 5\lambda_3 + 4\lambda_4 + 2\lambda_5 \geq 3 \\ & \lambda_j \in \mathbb{Z}_+ \end{aligned}$$

LP-solution $\lambda_1 = 1.375, \lambda_4 = 0.75$

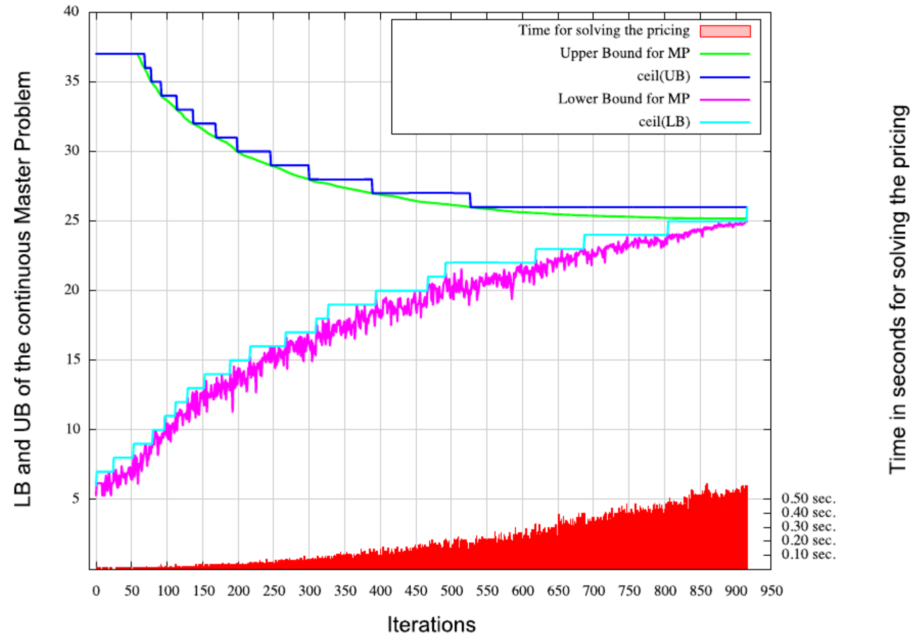
Branch on $\lambda_1 = 0$, $\lambda_1 = 1$, $\lambda_1 = 2$

- Column generation may not generate pattern (4,0)
- Pricing problem is knapsack problem with pattern forbidden

Tailing off effect
Column generation may converge slowly in the end

- We do not need exact solution, just lower bound
- Solving master problem for subset of columns does not give valid lower bound (why?)
- Instead we may use Lagrangian relaxation of joint constraint
- "guess" lagrangian multipliers equal to dual variables from master problem

Convergence in CG



[plot by Stefano Gualandi, Milan University]

Heuristic solution (eg, in sec. 12.6)

- Restricted master problem will only contain a subset of the columns
- We may solve restricted master problem to IP-optimality
- Restricted master is a "set-covering-like" problem which is not too difficult to solve

Outline

Ryan's branching rule in Set Partitioning

Solving the SCP integer program

Branch and bound

- Generate routes such that:
  - they are good in terms of cost
  - they reduce the potential for fractional solutions
- constraint branching [Ryan, Foster, 1981]

$$\exists \text{ constraints } r_1, r_2 : 0 < \sum_{j \in J(r_1, r_2)} x_j < 1$$

$J(r_1, r_2)$ all columns covering r_1, r_2 simultaneously. Branch on:

$$\sum_{j \in J(r_1, r_2)} x_j \leq 0 \qquad \qquad \sum_{j \in J(r_1, r_2)} x_j \geq 1$$