Universiteit Utrecht

[Faculty of **Science**
Information and Computing Sciences]

Column generation models

The Beunhaas problem

- Beun de Haas is an independent entrepreneur.
- Clients contact him for small jobs.
- Planning period: days 1, . . . , T.
- For each job j is given:
  - the reward (c_j);
  - the time it takes (a_j);
- Beun has *Q* time on each day
- **Goal.** Choose and plan the work to earn as much as possible.

- First: a simple model

Advanced ILP formulation

- Formulation with day plans.
- A day plan for day t is a set of jobs that Beun can do on day t.
- S is the set of feasible day plans
- The reward of day plan p is equal to C_p
- Use a binary variable:
  - x_p = 1 if day plan p from S is chosen, 0 otherwise
- *This is the Master problem*

ILP with day plans

- **Disadvantage:** solving ILP may take a long time
- **Solution**: relax integrality constraints, LP-relaxation.
  - Maybe fractional solution
  - Upperbound

ILP with day plans (2)

- **Disadvantage:** There are so many possible day plans
- **Solution:** Consider only interesting day plans **Column generation.**

Column generation for LP

1. Start with Restricted Master Problem: a small set of day plans
2. Solve LP-relaxation.
3. Find out if there is a new dayplans that can improve the solution (= **pricing**)
4. No $\Rightarrow$ optimum found
5. Yes $\Rightarrow$ add plan to model and go to 2.

Pricing= (Lagrangean) subproblem

- Finding out if there are day plans to improve solution
- Recall: variable can improve solution if and only if reduced cost are positive

- Pricing problem:
  - Find day plan with maximal reduced cost
    - If maximum > 0, add day plan
    - Otherwise stop
  - Knapsack problem
  - Solved by dynamic programming

The Beunhaas problem: generalization

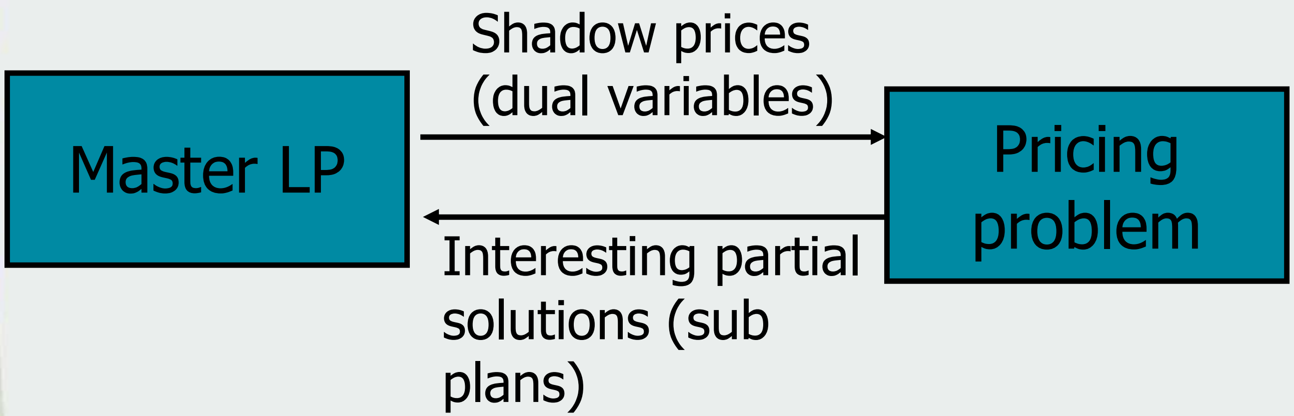
- Beun de Haas is an independent entrepreneur.
- Clients contact him for small jobs.
- Planning period: days 1, . . . , T.
- For each job j is given:
  - the reward (c_j);
  - the time it takes (a_j);
  - the days on which they can be done,
  - J_t set of jobs that available on day t
- Beun has Q_t time on day t (t = 1, . . . , T).
- **Goal.** Choose and plan the work to earn as much as possible.

Beun de Haas

Day	Working time
Monday	6 hours
Tuesday	8 hours
Wednesday	4 hours
Thursday	8 hours
Friday	4 hours

Job	Duration	Revenue	Days
1	2 hours	5	Mon, wed, fri
2	3 hours	6	Mon, tue, thu
3	2 hours	4	Wed, thu, fri
etc	…..	…..	…..

Column generation: Ping-pong



Column generation = cutting plane algorithm in dual

Primal:

- Restricted problem has limited set of variables
- *Pricing problem:* find variable that improves current solution (use reduced cost)

Column generation

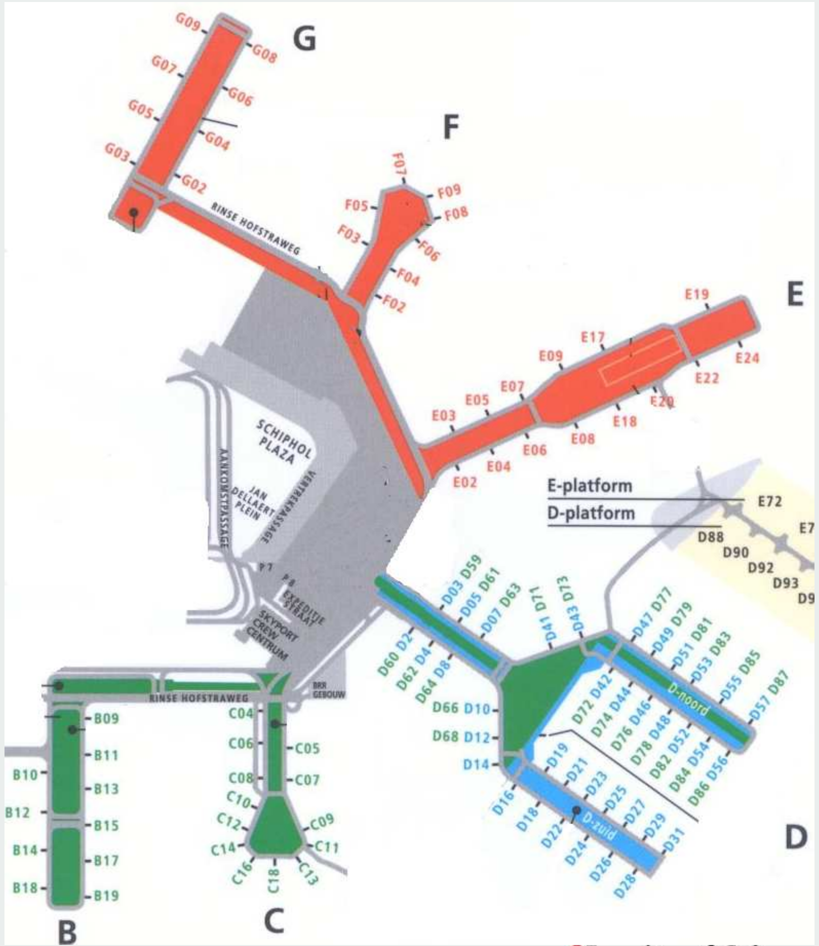
↔

Dual

- Restricted problem has limited set of constraints
- *Separation problem:* find constraint that violates current optimal solution

Cutting plane algorithm

Gate assignment at Schiphol



Gate assignment at Schiphol

We have a set of flights:

- Arrival and departure time
- Type of aircraft
- Region of origin/destination (Schengen/EU/Non-EU)
- Preferences of airline
- Ground handler

And we have a set of gates

- Possible regions (Schengen/EU/Non-EU)
- Possible aircraft
- Possible ground handlers

Gate assignment at Schiphol (2)

Goal:

- find assignment one day ahead
- maximize *robustness*

that satisfies:

- region constraints
- aircraft constraints
- ground handler constraints
- time constraints
- preferences

Gate assignment at Schiphol (3)

Cost of non-robustness is function of separation time between two flights

- High for small separation times
- Low for long separation times
- Descending steeply in beginning

$$t_{v,w}^{sep} = t_w^{arrival} - t_v^{departure}$$

$$C = \sum\nolimits_{v,w \text{ consecutive flights on the same gate}} c(t_{v,w}^{sep})$$

$$c(t^{sep}) = 1000(\arctan(0.21(5 - t^{sep})) + \frac{\pi}{2})$$

Refinements:

- Certain combinations of flights are more desirable
- Certain assignments are less desirable

Gate plans

- Distinguish only between gate types (not between individual gates): set of gates with the same ground handler, security region, aircraft size
- Gate plan for gate of type *a*:
  - Set of flights assigned to the same gate
  - All fights must be allowed on gate of type *a*
  - Time between two consecutive flights must be at least 20 minutes
  - Cost of gate plan = cost due to corresponding separation times
- Decision variable x_i = 0/1 if gate plan i is (not) selected.

Gate assignment: decomposition model

- Master problem:
  - Variables are plans for one gate
  - Each flight is on exactly one gate
  - Flight assigned to gate of correct type
  - Preference constraints and other
  - Maximize robustness
- Subproblem:
  - Feasible plans for one gate
    - All flights are allowed on the type of gate under consideration
    - At least 20 minutes between two consecutive flights.
  - Solved as shortest path problem on directed acyclic graph with topological ordering on the nodes.

Finding integral solutions

After solving the LP-relaxation by column generation, we have to find an integral solution. Possible methods:

- Branch-and-price: combination of branch-and-bound with column generation.
  - Finds optimal IP value Z_{GA}

$$Z_{LP} \leq Z_{GA}$$

- Solve ILP only with variables you generated during column generation. Finds $Z_{GA\text{-generated—columns}}$

$$Z_{LP} \leq Z_{GA} \leq Z_{GA\_\text{generated\_columns}}$$

Finding integral solutions (2)

- Generate pool of additional columns:
  1. Take optimal solution of pricing problem
  2. For each flight in this solution:
     1. Remove flight from the graph
     2. Compute shortest path
     3. Add new gate plan to pool
     4. Put flight back in graph
- Not added when solving the LP-relaxation by column generation but after that when we want to find an integral solution.

$$Z_{LP} \leq Z_{GA} \leq Z_{GA\,\text{with additional columns}} \leq Z_{GA\,\text{generated columns}}$$

Computational results

Standard ILP

- 80 flights
- 20 gates
- 66420 variabelen
- 64981 constraints
- 1.5 to 2 hours computation time

Column generation

- 560 flights
- 100 gates
- Generated columns: 12.000 - 13.000
- Additional columns: 65.000 - 85.000
- Computation time:
  - LP: 70 - 234 sec
  - ILP: 5 - 333 seconden
- Number of iteraties: 500 – 700
- Integrality gap: 0 – 2 ‰

(2) *(2 points)* A large production company owns m distribution centers from which goods are sent to the customers. For each center i ($i = 1, \ldots, m$) we know the cost k_i of keeping the center open, its capacity M_i, and the transportation cost q_i per unit for transporting goods from the production facility to center i. For each customer j ($j = 1, \ldots, n$) we know his demand v_j and the transportation cost c_{ij} per unit for sending goods from center i to customer j. Each customer has to be served by exactly one distribution center.

(c) An alternative way to formulate the problem in (b) is by using **customer groups**. A customer group for center i is a set of customers served by distribution center i such that the capacity M_i of center i is not exceeded. Define S_i as the collection of all feasible customers groups for center i. Give an integer linear programming formulation for the problem of part (b) based on customer groups.

(d) Describe how the LP-relaxation of this formulation can be solved by column generation. Your description should include a formulation of the pricing problem for a given center i. You do not have to describe how to solve the pricing problem (but you are allowed to do so).

Exam 2012

- m distribution centers, M_i capacity of center i
- n customers, v_j demand of customer j
- Cost:
  - k_i for opening depot i
  - q_i per unit for transportation from production facility to depot i
  - c_{ij} per unit for transportation from depot i to customer j

Which depots should be opened, what is optimal transportation plan?

How to solve by column generation?