

Optimierung und Entscheidungsunterstützung (OuE)

Anna Prenzel

26.04.2013

Hochschule Zittau/Görlitz

Topics for today:

- 1 Branch and Bound for IP
- 2 Performance Speedup with Cuts
- 3 Ideas for Job Shop Scheduling

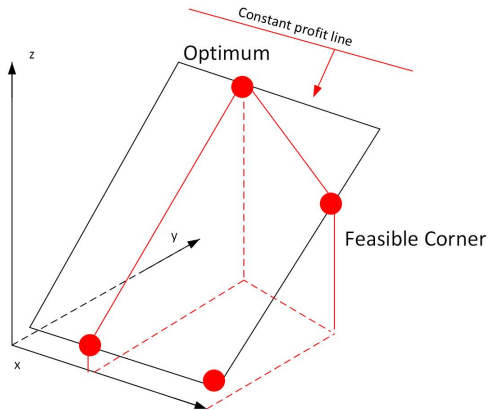
Important facts

- The feasible region for any IP must be contained in the feasible region for the corresponding LP relaxation.

For any IP that is a max problem:

Optimal z-value for LP relaxation $\geq$ optimal z-value for IP

Important facts



Branch and Bound

- Divide and conquer method
- We *divide* a large problem into a few smaller ones → branch part
- We *conquer* by estimating the quality of the solution we can get for each smaller problem → bound part

B and B Steps

- ① Divide a problem into subproblems.
- ② We fathom a subproblem, if:
 - The LP problem has no feasible solution, done.
 - The LP problem has an integer optimal solution; done.
(update incumbent solution)
 - The LP problem has an optimal solution that is worse than the incumbent, done.
- ③ The LP problem has an optimal solution that are not all integer, better than the incumbent. In this case we would have to divide this subproblem further and repeat.

Example

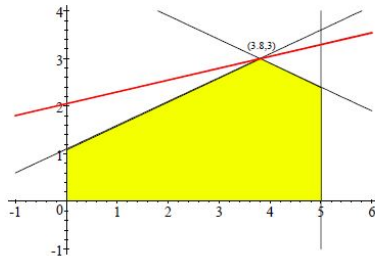
$x_i \geq 0$, x_i integer

$$\max Z = -x_1 + 4x_2 \quad (1)$$

$$-10x_1 + 20x_2 \leq 22 \quad (2)$$

$$5x_1 + 10x_2 \leq 49 \quad (3)$$

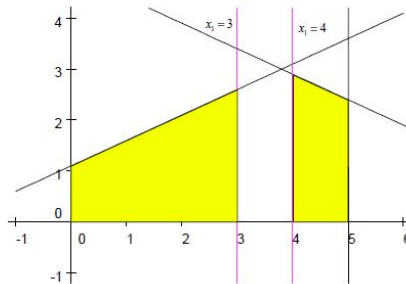
$$x_1 \leq 5 \quad (4)$$



Example

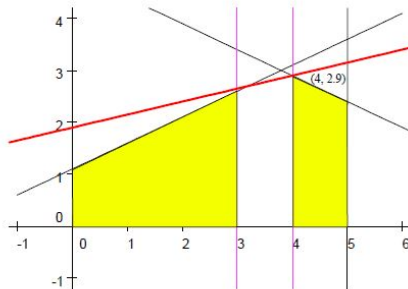
We solve the LP relaxation ($x_i \geq 0$). Optimal solution of the relaxation is $(3.8, 3)$ with $z=8.2$. Then we consider two cases:

- $x_1 \geq 4$
- $x_1 \leq 3$

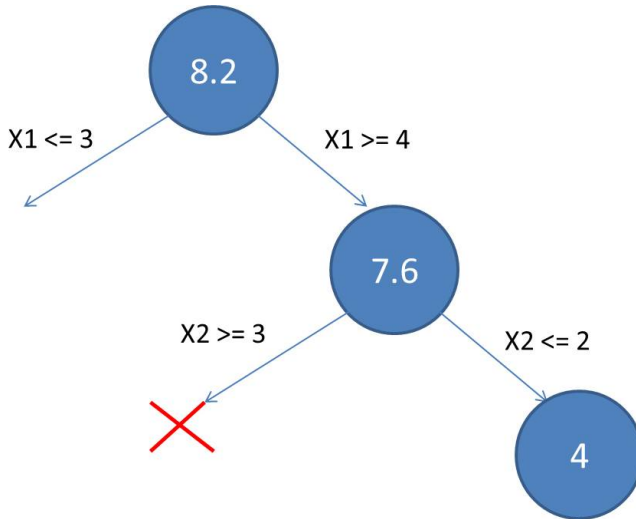


Example

The LP relaxation $x_1 \geq 4$ has an optimal solution at $(4, 2.9)$ with $Z = 7.6$.

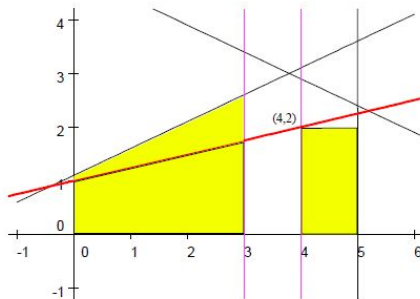


Example



Example

- In the next step, we branch x_2 , $x_2 \geq 3$, $x_2 \leq 2$.
- The LP relaxation with $x_2 \geq 3$ has no feasible solution.
- The LP relaxation with $x_2 \leq 2$ has an optimal solution at $(4,2)$ with $Z = 4$.

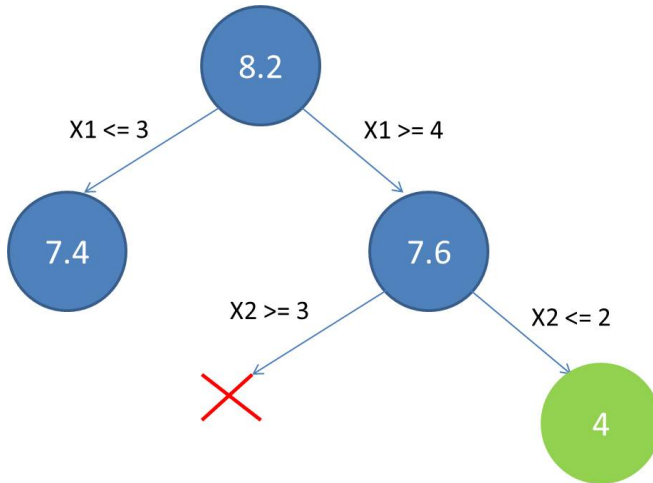


Example

Update incumbent

Currently, the best value of Z for the original IP is $Z = 4$.

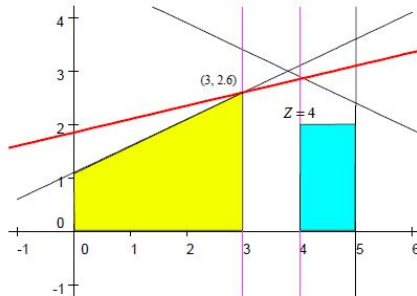
Example



Example

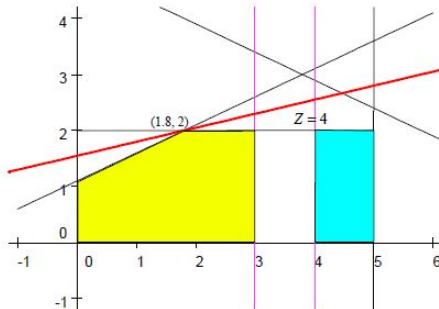
Now we consider the branch of $x_1 \leq 3$. The LP relaxation has an optimal solution at $(3, 2.6)$ with $Z = 7.4$. We branch out further:

- $x_2 \leq 2$
- $x_2 \geq 3$

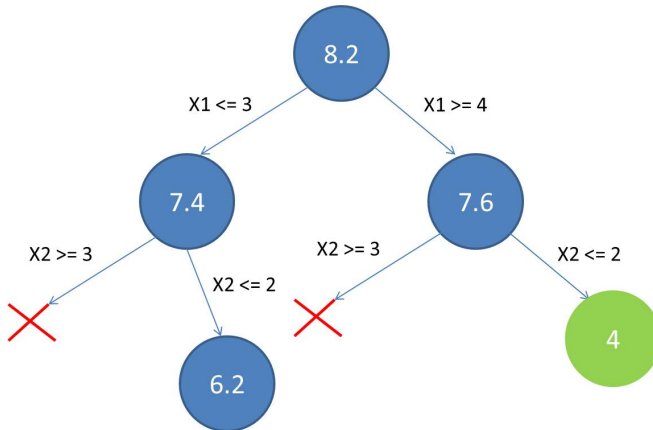


Example

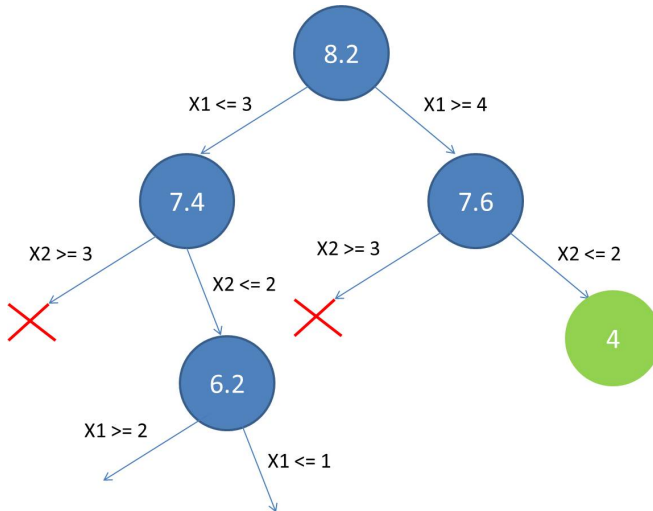
The LP relaxation with $x_2 \geq 3$ has no solution. The LP relaxation with $x_2 \leq 2$ has an optimal solution at $(1.8, 2)$ with $Z = 6.2$.



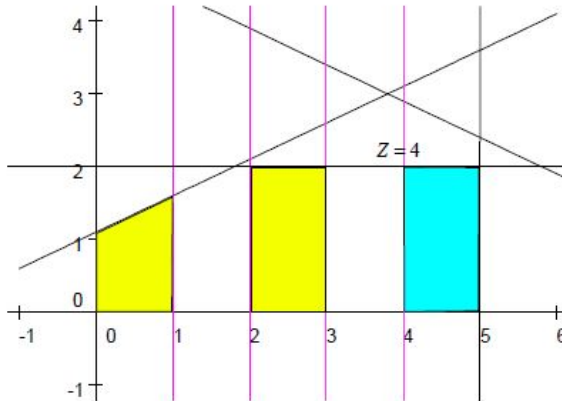
Example



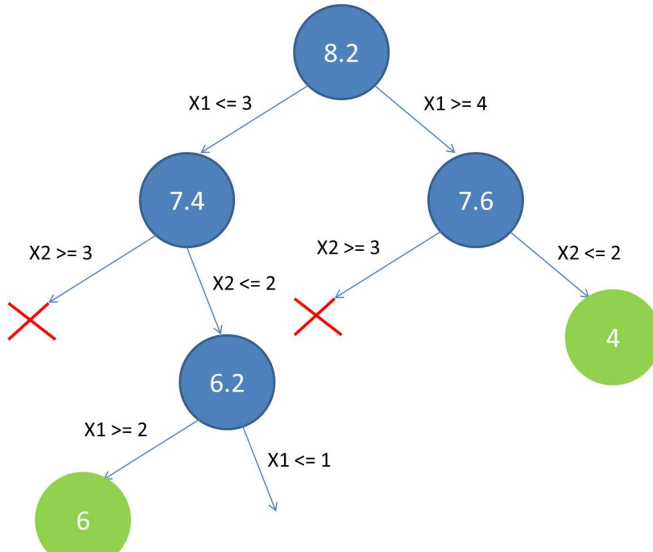
Example



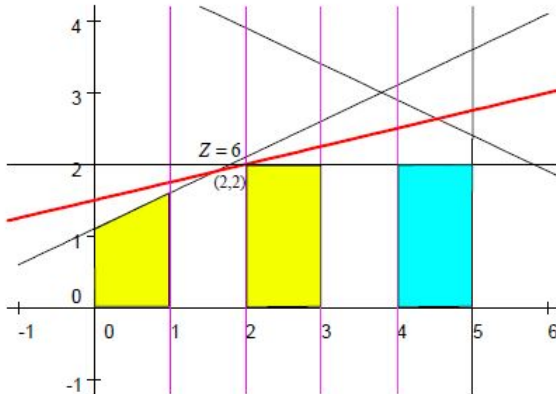
Example



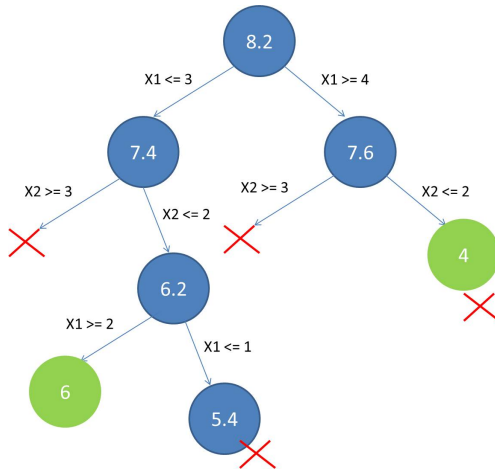
Example



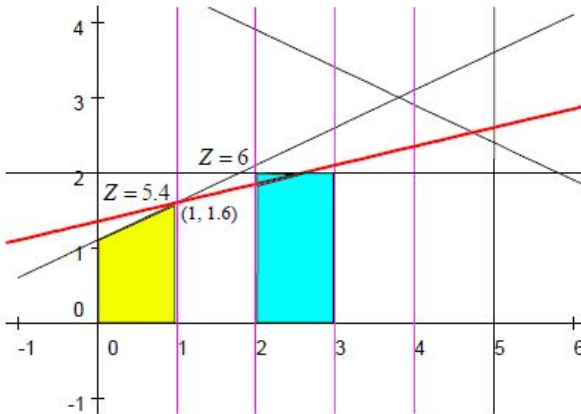
Example



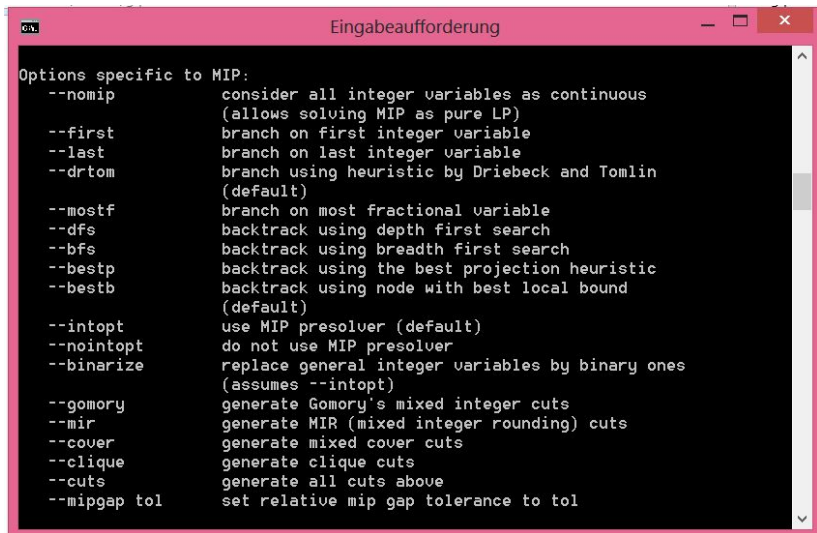
Example



Example



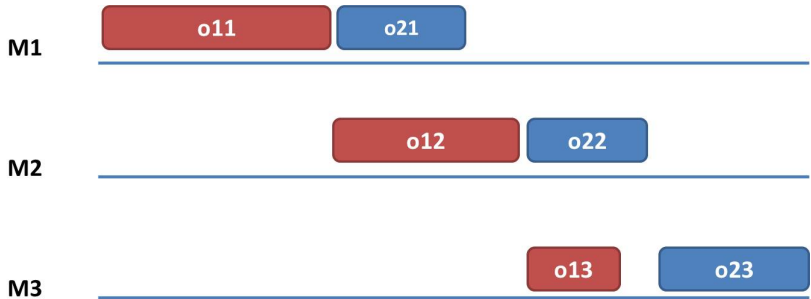
GLPK-settings for MIP



```
Options specific to MIP:
--nomip          consider all integer variables as continuous
                  (allows solving MIP as pure LP)
--first          branch on first integer variable
--last           branch on last integer variable
--drtom          branch using heuristic by Driebeck and Tomlin
                  (default)
--mostf          branch on most fractional variable
--dfs            backtrack using depth first search
--bfs            backtrack using breadth first search
--bestp          backtrack using the best projection heuristic
--bestb          backtrack using node with best local bound
                  (default)
--intopt         use MIP presolver (default)
--nointopt       do not use MIP presolver
--binarize       replace general integer variables by binary ones
                  (assumes --intopt)
--gomory         generate Gomory's mixed integer cuts
--mir            generate MIR (mixed integer rounding) cuts
--cover          generate mixed cover cuts
--clique         generate clique cuts
--cuts           generate all cuts above
--mipgap tol     set relative mip gap tolerance to tol
```

JSSP

o_{ij} = operation j of job i



JSSP

Variables:

- s_{im} , start time of job i on machine m
- ??

JSSP

Constraints:

- $s_{im} + \text{time}[i][m] \leq ??$
- What is the successor of Job i on machine m ? $\rightarrow$ we do not know it.

JSSP - Constraints

Approach:

- ① Introduce *decision* variable, that decides, if job i precedes job j on machine m or not.
- ② For each jobs and machines, introduce two constraints:
 - start time of job i is before start time of job j , if decision variable is 1
 - start time of job i is after start time of job j , if decision variable is 0

Attention

Both constraints cannot be active at the same time! Use Big M to turn the correct constraint on / off.

JSSP - Constraints

Second Constraint

Ensure that operation $j-1$ of job i on machine h precedes operation j of job i on machine k .

Das war's für heute :-)