Implementing Branch and Bound in C++ using SCIP

Djime Gueye

13 February 2018

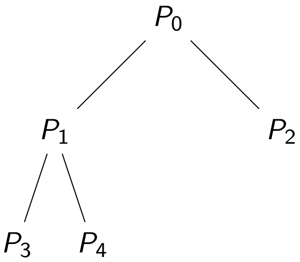
Outline

- Method for solving linear integer programs.
- Simulates complete tree search of the solutions.
- This presentation will consider a maximization problem.

Principle Operations

Branching

- Divide the problem to solve P_0 in subproblems $P_1, P_2, ..., P_k$ so that $X(P_0) = \bigcup_{i=1}^{k} X(P_i)$ and $X(P_i) \cap X(P_j) = \varnothing$ for all $i \neq j$
- The subproblems are on their turn branched and we get a solution tree.With root P_0



Bounding

- The best feasible solution found is a lower bound on the optimal objective value. Use heuristic at the beginning or set $\underline{F} = -\infty$
- For each problem P_i determine an upper bound $\overline{F_i}$ on the optimal objective value of the subproblem.
  - Solve relaxation P_i', a simplification of P_i with $X(P_i') \subset X(P_i)$ we get by removing or "relaxing" some constraints

Pruning

Stop considering i.e. **prune** a subproblem in these cases:

- if $\overline{F_i} \leq \underline{F}$.The optimal solution of this subproblem can not be better than the lower bound.
- if $\overline{F_i} > \underline{F}$ and the solution is feasible.Set new lower bound $\underline{F} = \overline{F_i}$
- if $X(P_i') = \varnothing$ P_i is infeasible and doesn't have to be observed anymore.

Example

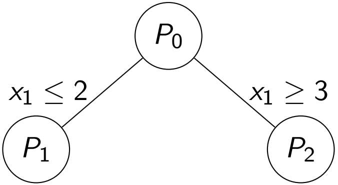
$$
\begin{aligned}
&\max && x_1 + 2x_2 \\
&\text{subject to} && \\
&x_1 + 3x_2 \leq 7 && \\
&3x_1 + 2x_2 \leq 10 && \\
&x_1, x_2 \geq 0 \text{ and integers} &&
\end{aligned}
$$

Example cont.

P_0

- Since $(0,0)$ is feasible, set $\underline{F}$=0
- Build relaxation of P_0 the start problem by removing the integrality constraint, and solve
  max $x_1 + 2x_2$ under $x_1 + 3x_2 \leq 7, 3x_1 + 2x_2 \leq 10$ and $x_1, x_2 \geq 0$
  With the simplex method or graphically. The optimal solution is $(2.29, 1.57)$ and $F_0 = 5.43$.
- $\overline{F_0} = 5.43$
- Build Subproblems P_1= max $x_1 + 2x_2$ under $x_1 + 3x_2 \leq 7, 3x_1 + 2x_2 \leq 10, x_1 \leq 2$ and $x_1, x_2 \geq 0$ and integers and
  P_2= max $x_1 + 2x_2$ under $x_1 + 3x_2 \leq 7, 3x_1 + 2x_2 \leq 10, x_1 \geq 3$ and $x_1, x_2 \geq 0$ and integers.

Tree

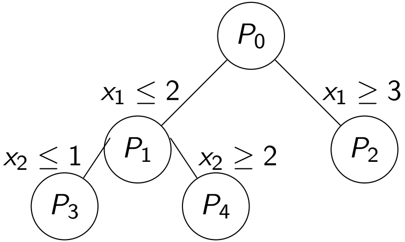


Example cont.

P_1

- Solve relaxation of P_1
  max $x_1 + 2x_2$ under $x_1 + 3x_2 \leq 7, 3x_1 + 2x_2 \leq 10, x_1 \leq 2$ and $x_1, x_2 \geq 0$
- Optimal Solution is $(2, 1.667)$ objective value F_1=5.33
- $\overline{F_1} = 5.33$
- Build Subproblems P_3= max $x_1 + 2x_2$ under $x_1 + 3x_2 \leq 7, 3x_1 + 2x_2 \leq 10, x_1 \leq 2, x_2 \leq 1$ and $x_1, x_2 \geq 0$ and integers and P_4= max $x_1 + 2x_2$ under $x_1 + 3x_2 \leq 7, 3x_1 + 2x_2 \leq 10, x_1 \leq 2, x_2 \geq 2$ and $x_1, x_2 \geq 0$ and integers.

Tree



Example

P_2

- Solve relaxation of P_2
  max $x_1 + 2x_2$ under $x_1 + 3x_2 \leq 7, 3x_1 + 2x_2 \leq 10, x_1 \geq 3$ and $x_1, x_2 \geq 0$
- Optimal Solution is $(3, 0.5)$ objective value F_2=4
- $\overline{F_2} = 4$

Example

P_3

- Solve relaxation of P_3 max $x_1 + 2x_2$ under $x_1 + 3x_2 \leq 7, 3x_1 + 2x_2 \leq 10, x_1 \leq 2, x_2 \leq 1$ and $x_1, x_2 \geq 0$
- Optimal Solution is $(2, 1)$ objective value F_3=4
- $(2, 1)$ is the new best solution, $\underline{F}$=4.

Example

P_4

- Solve relaxation of P_4 max $x_1 + 2x_2$ under $x_1 + 3x_2 \leq 7, 3x_1 + 2x_2 \leq 10, x_1 \leq 2, x_2 \geq 2$ and $x_1, x_2 \geq 0$
- Optimal Solution is $(1, 2)$ objective value $F_4{=}5$
- $(1, 2)$ is the new best solution, $\underline{F}{=}5$.
- $\overline{F_2} < 5$, prune P_2
- There is no more problem to observe the optimal solution is $(1, 2)$ and optimal objective value 5.

Strategies for choosing next Candidate

The LIFO_Rule

The problem that has been added at last to the candidate list will be branched first.

The Maximum upper bound strategy

Choose the problem from the list that has a highest Upper bound.

Implementation

The problem should be formulated in this form:

$$\begin{array}{ll} \text{min} & c^T x \\ \text{subject to} & Ax \geq b \\ x \geq 0 & \\ x_j \text{ is integer} & j = 1, ..., p \leq n \end{array}$$

- Create text file with problem data in this format
- Create problem instance as object of the Class MBP.
- The problem data will be stored in this class

The LP class

- LP solver that uses scip
- will be used to solve the subproblems

The Node Class

- represents a subproblem
- Has an LP- attribute: the relaxed problem
- Linked to its subproblems

The solving process

The BranchandBound Class

- Has a Linked list as Datastructure for the branch and Bound tree
- important attributes
  - The global upper bound
  - The lowest lower Bound
- The important methods
  - The Branch method
  - the bound method
  - the solve method

For Further Reading I

W. Domschke
Einfuehrung in Operations Research.
Springer,2015.