ip programming

**library**("lpSolveAPI")

**library**("data.tree")

base\_ip <- read.lp("ip\_someike.lp")

base\_ip

## Model name:

## x1 x2

## Maximize 8 5

## R1 1 1 <= 6

## R2 9 5 <= 45

## Kind Std Std

## Type Real Real

## Upper Inf Inf

## Lower 0 0

solve(base\_ip)

## [1] 0

get.objective(base\_ip)

## [1] 41.25

get.variables(base\_ip)

## [1] 3.75 2.25

base <- Node$new("Z = 41.25 [X1 = 3.75, X2 = 2.25]")

base\_ip\_left\_1 <- read.lp("ip\_someike\_left\_1.lp")

base\_ip\_left\_1

## Model name:

## x1 x2

## Maximize 8 5

## R1 1 1 <= 6

## R2 9 5 <= 45

## Kind Std Std

## Type Real Real

## Upper 3 Inf

## Lower 0 0

solve(base\_ip\_left\_1)

## [1] 0

get.objective(base\_ip\_left\_1)

## [1] 39

get.variables(base\_ip\_left\_1)

## [1] 3 3

left\_1 <- base$AddChild("Branching on X1 <= 3 --> Z = 39 [X1 = 3, X2 = 3]")

base\_ip\_right\_1 <- read.lp("ip\_someike\_right\_1.lp")

base\_ip\_right\_1

## Model name:

## x1 x2

## Maximize 8 5

## R1 1 1 <= 6

## R2 9 5 <= 45

## Kind Std Std

## Type Real Real

## Upper Inf Inf

## Lower 4 0

solve(base\_ip\_right\_1)

## [1] 0

get.objective(base\_ip\_right\_1)

## [1] 41

get.variables(base\_ip\_right\_1)

## [1] 4.0 1.8

right\_1 <- base$AddChild(">>Branching on X1 >= 4 --> Z = 41 [X1 = 4, X2 = 1.80]")

base\_ip\_right\_1\_left\_2 <- read.lp("ip\_someike\_right\_1\_left\_2.lp")

base\_ip\_right\_1\_left\_2

## Model name:

## x1 x2

## Maximize 8 5

## R1 1 1 <= 6

## R2 9 5 <= 45

## Kind Std Std

## Type Real Real

## Upper Inf 1

## Lower 4 0

solve(base\_ip\_right\_1\_left\_2)

## [1] 0

get.objective(base\_ip\_right\_1\_left\_2)

## [1] 40.55556

get.variables(base\_ip\_right\_1\_left\_2)

## [1] 4.444444 1.000000

left\_2 <- right\_1$AddChild(">>Branching on X2 <= 1 --> Z = 40.56 [X1 = 4.44, X2 = 1.00]")

right\_1\_left\_2\_left\_3 <- read.lp("ip\_someike\_right\_1\_left\_2\_left\_3.lp")

right\_1\_left\_2\_left\_3

## Model name:

## x1 x2

## Maximize 8 5

## R1 1 1 <= 6

## R2 9 5 <= 45

## Kind Std Std

## Type Real Real

## Upper 4 1

## Lower 4 0

solve(right\_1\_left\_2\_left\_3)

## [1] 0

get.objective(right\_1\_left\_2\_left\_3)

## [1] 37

get.variables(right\_1\_left\_2\_left\_3)

## [1] 4 1

left\_3 <- left\_2$AddChild("Branching on X1 <= 4 --> Z = 37 [X1 = 4.00, X2 = 1.00]")

base\_ip\_right\_1\_left\_2\_right\_3 <- read.lp("ip\_someike\_right\_1\_left\_2\_right\_3.lp")

base\_ip\_right\_1\_left\_2\_right\_3

## Model name:

## x1 x2

## Maximize 8 5

## R1 1 1 <= 6

## R2 9 5 <= 45

## Kind Std Std

## Type Real Real

## Upper Inf 1

## Lower 5 0

solve(base\_ip\_right\_1\_left\_2\_right\_3)

## [1] 0

get.objective(base\_ip\_right\_1\_left\_2\_right\_3)

## [1] 40

get.variables(base\_ip\_right\_1\_left\_2\_right\_3)

## [1] 5 0

right\_3 <- left\_2$AddChild(">>Branching on X1 >= 5 --> Z = 40 [X1 = 5.00, X2 = 1.00]")

base\_ip\_right\_1\_right\_2 <- read.lp("ip\_someike\_right\_1\_right\_2.lp")

base\_ip\_right\_1\_right\_2

## Model name:

## x1 x2

## Maximize 8 5

## R1 1 1 <= 6

## R2 9 5 <= 45

## Kind Std Std

## Type Real Real

## Upper Inf Inf

## Lower 4 2

solve(base\_ip\_right\_1\_right\_2)

## [1] 2

get.objective(base\_ip\_right\_1\_right\_2)

## [1] -1e+30

get.variables(base\_ip\_right\_1\_right\_2)

## [1] 0 0

right\_2 <- right\_1$AddChild("Branching on X2 >= 2 --> Infeasible")

print(base)

## levelName

## 1 Z = 41.25 [X1 = 3.75, X2 = 2.25]

## 2 ¦--Branching on X1 <= 3 --> Z = 39 [X1 = 3, X2 = 3]

## 3 °-->>Branching on X1 >= 4 --> Z = 41 [X1 = 4, X2 = 1.80]

## 4 ¦-->>Branching on X2 <= 1 --> Z = 40.56 [X1 = 4.44, X2 = 1.00]

## 5 ¦ ¦--Branching on X1 <= 4 --> Z = 37 [X1 = 4.00, X2 = 1.00]

## 6 ¦ °-->>Branching on X1 >= 5 --> Z = 40 [X1 = 5.00, X2 = 1.00]

## 7 °--Branching on X2 >= 2 --> Infeasible

The final solution is: Z = 40 [X1 = 5.00, X2 = 1.00]