

Hands-on Tutorial on Optimization

F. Eberle, R. Hoeksma, and N. Megow

September 27, 2019

Non-optimal solutions

What do we do if the problem is too difficult for the solver?

What do we do if the problem is too difficult for the solver?

- ▶ Wait

What do we do if the problem is too difficult for the solver?

- ▶ Wait
- ▶ Increase computing power (buy a better computer)

What do we do if the problem is too difficult for the solver?

- ▶ Wait
- ▶ Increase computing power (buy a better computer)
- ▶ Find a different formulation

What do we do if the problem is too difficult for the solver?

- ▶ Wait
- ▶ Increase computing power (buy a better computer)
- ▶ Find a different formulation
- ▶ Use a different solver

What do we do if the problem is too difficult for the solver?

- ▶ Wait
- ▶ Increase computing power (buy a better computer)
- ▶ Find a different formulation
- ▶ Use a different solver
- ▶ Solve the problem non-optimally

What do we do if the problem is too difficult for the solver?

- ▶ Wait
- ▶ Increase computing power (buy a better computer)
- ▶ Find a different formulation
- ▶ Use a different solver
- ▶ Solve the problem non-optimally

Solver options in CPLEX

- ▶ Iteration / node limits: Solver terminates after a iterations / nodes.
- ▶ Time limit in seconds: Solver terminates after a seconds
- ▶ Relative gap tolerance: Solver terminates at relative gap a
- ▶ Absolute gap tolerance: Solver terminates at absolute gap a

Decompose the problem

Some problems allow for splitting into different subproblems, e.g., when timetabling trains, we could plan international trains before national trains and regional trains.

Decompose the problem

Some problems allow for splitting into different subproblems, e.g., when timetabling trains, we could plan international trains before national trains and regional trains.

Often used decomposition method is according to time.

- ▶ First solve a $T + t$ length time interval
- ▶ Then treat T as input and solve the next $T + t$ time interval.



Decompose the problem

Some problems allow for splitting into different subproblems, e.g., when timetabling trains, we could plan international trains before national trains and regional trains.

Often used decomposition method is according to time.

- ▶ First solve a $T + t$ length time interval
- ▶ Then treat T as input and solve the next $T + t$ time interval.

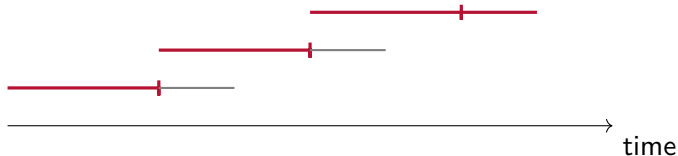


Decompose the problem

Some problems allow for splitting into different subproblems, e.g., when timetabling trains, we could plan international trains before national trains and regional trains.

Often used decomposition method is according to time.

- ▶ First solve a $T + t$ length time interval
- ▶ Then treat T as input and solve the next $T + t$ time interval.

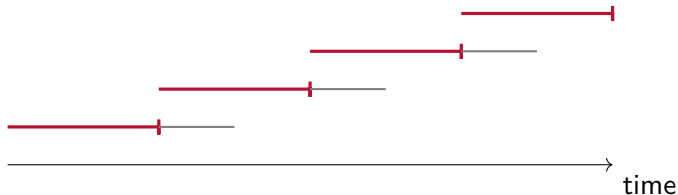


Decompose the problem

Some problems allow for splitting into different subproblems, e.g., when timetabling trains, we could plan international trains before national trains and regional trains.

Often used decomposition method is according to time.

- ▶ First solve a $T + t$ length time interval
- ▶ Then treat T as input and solve the next $T + t$ time interval.



Decompose the problem

Another example of a decomposable problem is one that is geographically distributed.

Decompose the problem

Another example of a decomposable problem is one that is geographically distributed.

For example, a facility location problem where customers are clearly partitioned.

Decompose the problem

Another example of a decomposable problem is one that is geographically distributed.

For example, a facility location problem where customers are clearly partitioned.



What if my problem is infeasible?

Sometimes the data that you have results in an infeasible problem.

What if my problem is infeasible?

Sometimes the data that you have results in an infeasible problem.

- ▶ The client gave too many constraints.

What if my problem is infeasible?

Sometimes the data that you have results in an infeasible problem.

- ▶ The client gave too many constraints.
- ▶ Some of the constraints might not be 'hard'.

What if my problem is infeasible?

Sometimes the data that you have results in an infeasible problem.

- ▶ The client gave too many constraints.
- ▶ Some of the constraints might not be 'hard'.

$$\begin{array}{ll} \min & \sum_{j \in J} c_j x_j \\ \text{s.t.} & \sum_{j \in J} a_{ij} x_j \geq b_i \quad \forall i \\ & x_j \geq 0 \quad \forall j \end{array}$$

What if my problem is infeasible?

Sometimes the data that you have results in an infeasible problem.

- ▶ The client gave too many constraints.
- ▶ Some of the constraints might not be 'hard'.

$$\begin{array}{ll} \min & \sum_{j \in J} c_j x_j \\ \text{s.t.} & \sum_{j \in J} a_{ij} x_j \geq b_i \quad \forall i \neq k \\ & \sum_{j \in J} a_{kj} x_j \geq b_k \\ & x_j \geq 0 \quad \forall j \end{array}$$

What if my problem is infeasible?

Sometimes the data that you have results in an infeasible problem.

- ▶ The client gave too many constraints.
- ▶ Some of the constraints might not be 'hard'.

$$\begin{aligned} \min \quad & \sum_{j \in J} c_j x_j + \lambda_k \left(b_k - \sum_{j \in J} a_{kj} x_j \right) \\ \text{s.t.} \quad & \sum_{j \in J} a_{ij} x_j \geq b_i && \forall i \neq k \\ & x_j \geq 0 && \forall j \end{aligned}$$

What if my problem is infeasible?

Sometimes the data that you have results in an infeasible problem.

- ▶ The client gave too many constraints.
- ▶ Some of the constraints might not be 'hard'.

$$\begin{aligned} \min \quad & \sum_{j \in J} c_j x_j + \lambda_k \left(b_k - \sum_{j \in J} a_{kj} x_j \right) \\ \text{s.t.} \quad & \sum_{j \in J} a_{ij} x_j \geq b_i && \forall i \neq k \\ & x_j \geq 0 && \forall j \end{aligned}$$

Known as: Lagrangian relaxation