Non-optimal solutions
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
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 GAMS

- `iterlim = a`: iteration limit: Solver terminates after $a$ iterations (default $2 \cdot 10^{10}$)
- `reslim = a`: time limit in seconds: Solver terminates after $a$ seconds (default 1000)
- `optcr = a`: relative gap tolerance: Solver terminates at relative gap $a$ (default 0.1)

GAMS syntax:

```
Options	hrenlim = 600
optcr = 0
;
```
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.
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'.

\[
\begin{align*}
\text{min} & \quad \sum_{j \in J} c_j x_j \\
\text{s.t.} & \quad \sum_{j \in J} a_{ij} x_j \geq b_i \quad \forall i \neq k \\
& \quad x_j \geq 0 \quad \forall j
\end{align*}
\]
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'.

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

Known as: Lagrangian relaxation
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'.

$$\begin{align*}
\min & \quad \sum_{j \in J} c_j x_j + \lambda_k \left( b_k - \sum_{j \in J} a_{kj} x_j \right) \\
\text{s.t.} & \quad \sum_{j \in J} a_{ij} x_j \geq b_i \quad \forall i \neq k \\
& \quad x_j \geq 0 \quad \forall j
\end{align*}$$
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'.

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

Known as: Lagrangian relaxation

[Credits to Universität Bremen]