Applications in Research and Industry

Michael Ritter Matthias Silbernagl

Technische Universität München

Workshop on Optimization Applications with FICO Xpress 2010

Xpress in Mathematics Education

- courses complementing mathematical programming classes
- hands-on modelling experience
- implementation using Mosel
- trial and error ➡ learning experience
- involvement in industry/research projects

flight scheduling at airports

- increasing air traffic
- limited airport capacity
- operational constraints
- legal restrictions

objective: optimal airport schedule

flight scheduling at airports

Issue:

- limited airport capacity $\leftrightarrow$ increasing demand
- allocation of arrival/departure slots

flight scheduling at airports

Issue:

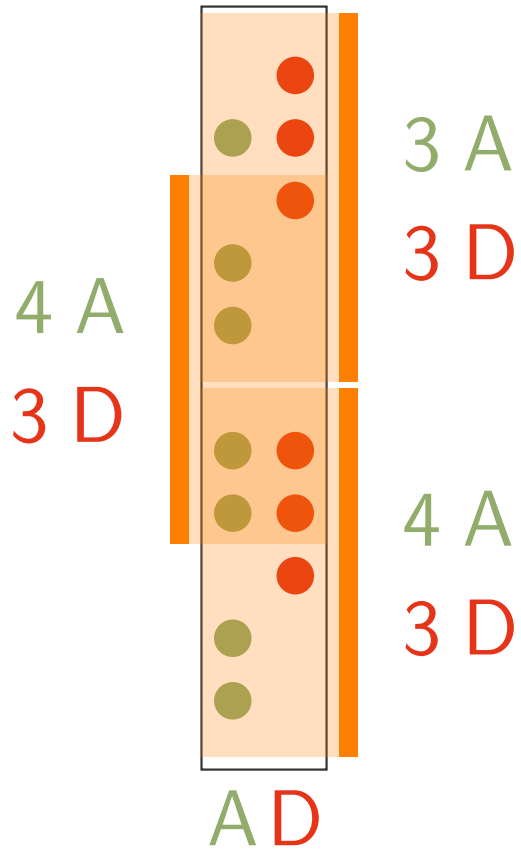
- limited airport capacity $\leftrightarrow$ increasing demand
- allocation of arrival/departure slots

Constraints:

- airlines' demands
- arrival / departure correspondance
- limited ground times
- coupled series
- hub/feeder flights
- legal restrictions
- time windows

time window constraints

- bounds on arrivals, departures, total flights
- time windows overlapping
- local bounds
  ↔ “globalized” effect



model characteristics

- integer linear program
- one season (6 months)
  - ➡ $\approx$ 190 000 time windows
  - ➡ $\approx$ 200 000 requests
  - ➡ many linked requests
- $\approx$ 150 000 binary variables
- $\approx$ 300 000 constraints

implementation

- Java software embedding Xpress Mosel
- data preprocessing
  - ➡ decrease number of variables
- SOS-type constraints
  - ➡ guide branching process
- fast heuristics
  - ➡ feasible solutions
  - ➡ bound generation
  - ➡ speed up branch & bound

problems in flight scheduling practice

- limited influence on scheduling process
- non-optimal scheduling procedures still implemented
- combinatorics of time windows

➡ “blocking” in real flight schedules (full, but not maximum)

problems in flight scheduling practice

new question

Design time windows such that ...

- capacity limits are respected
- no “bad” schedule is possible
- optimal number of flights does not change

optimal time window structure

- structural analysis
- “optimal” blocking strategies ➡ IP model
- devise strategies to avoid blocking
- extensive tests using Java/Mosel software

results summary

flight scheduling:

- increase by $\approx 4\,000$ movements
- generally better schedule

time window design:

- several strategies devised
- extensive tests using Java/Mosel
- small trade-off:
  flights $\leftrightarrow$ robustness

traffic infrastructure repair works

- bridge repair works
  ➡ traffic impact
- staff/budget/time restrictions
- third-party works
  (e. g. railway tracks)

objective: minimize traffic impact

traffic infrastructure repairs works

traffic infrastructure

- size, condition ➡ repair cost
- current condition ➡ repair deadline
- third-party infrastructure (railway tracks)
- traffic impact: “network effect”
  ➡ complex interdependencies

traffic infrastructure repairs works

traffic infrastructure

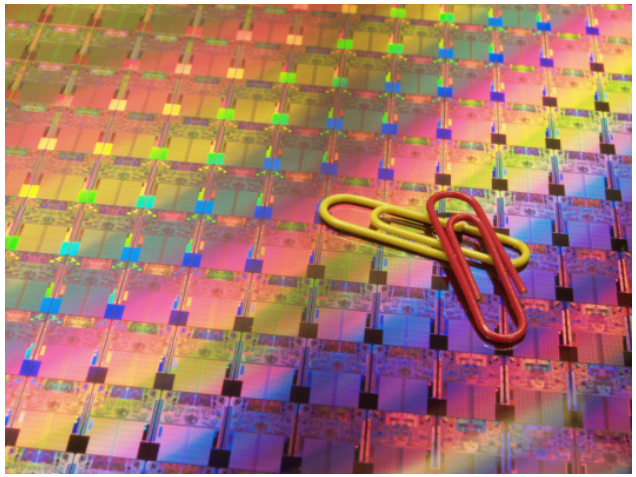
- size, condition ➡ repair cost
- current condition ➡ repair deadline
- third-party infrastructure (railway tracks)
- traffic impact: “network effect”
  ➡ complex interdependencies

objective

- minimize traffic impact
- balance repair costs
- utilize third-party repairs whenever possible

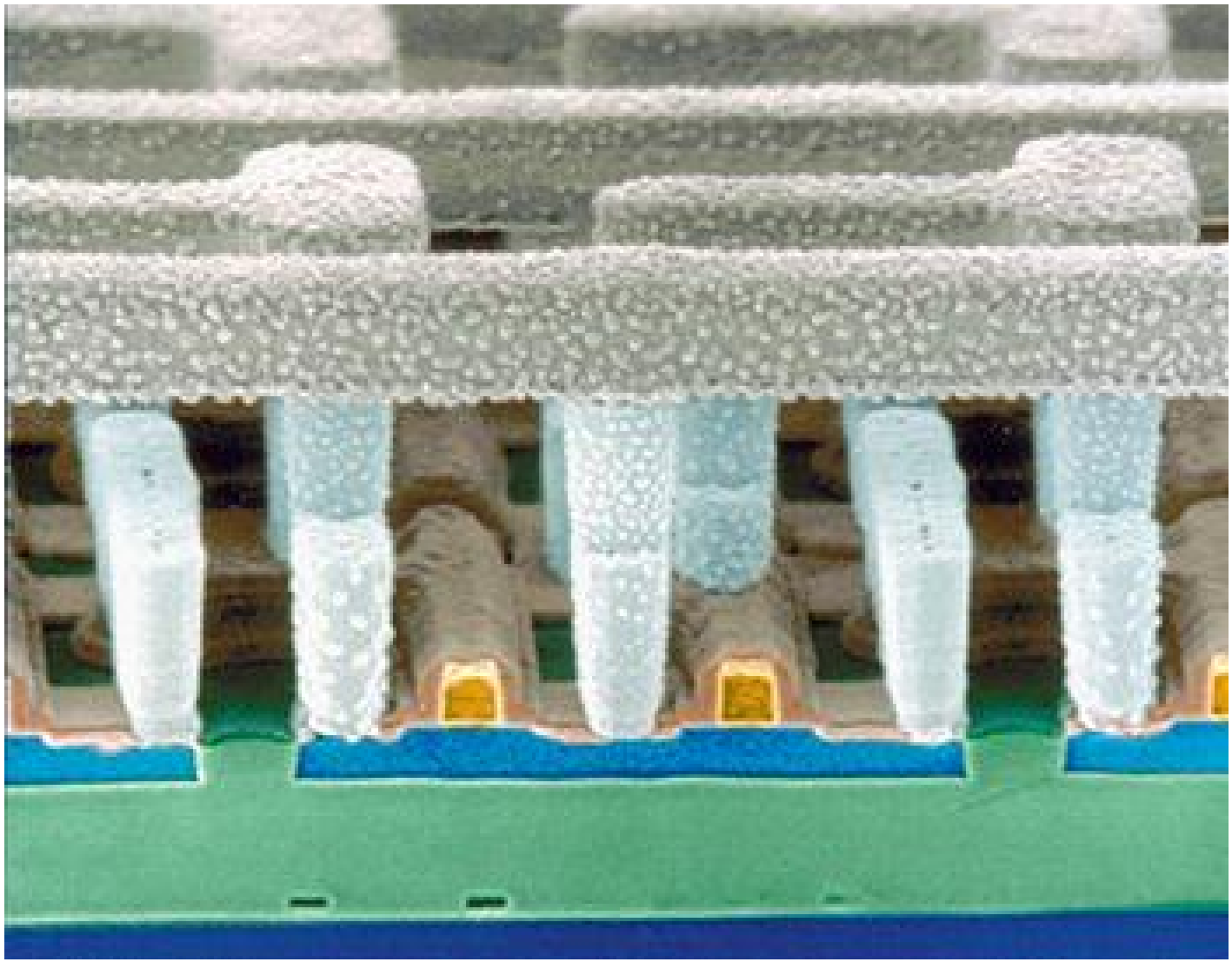
power loss in semiconductor circuits

- energy consumption, heat dissipation
- increased risk of failure
  ➡ sophisticated cooling devices
- mobile devices, medicine, aeronautics

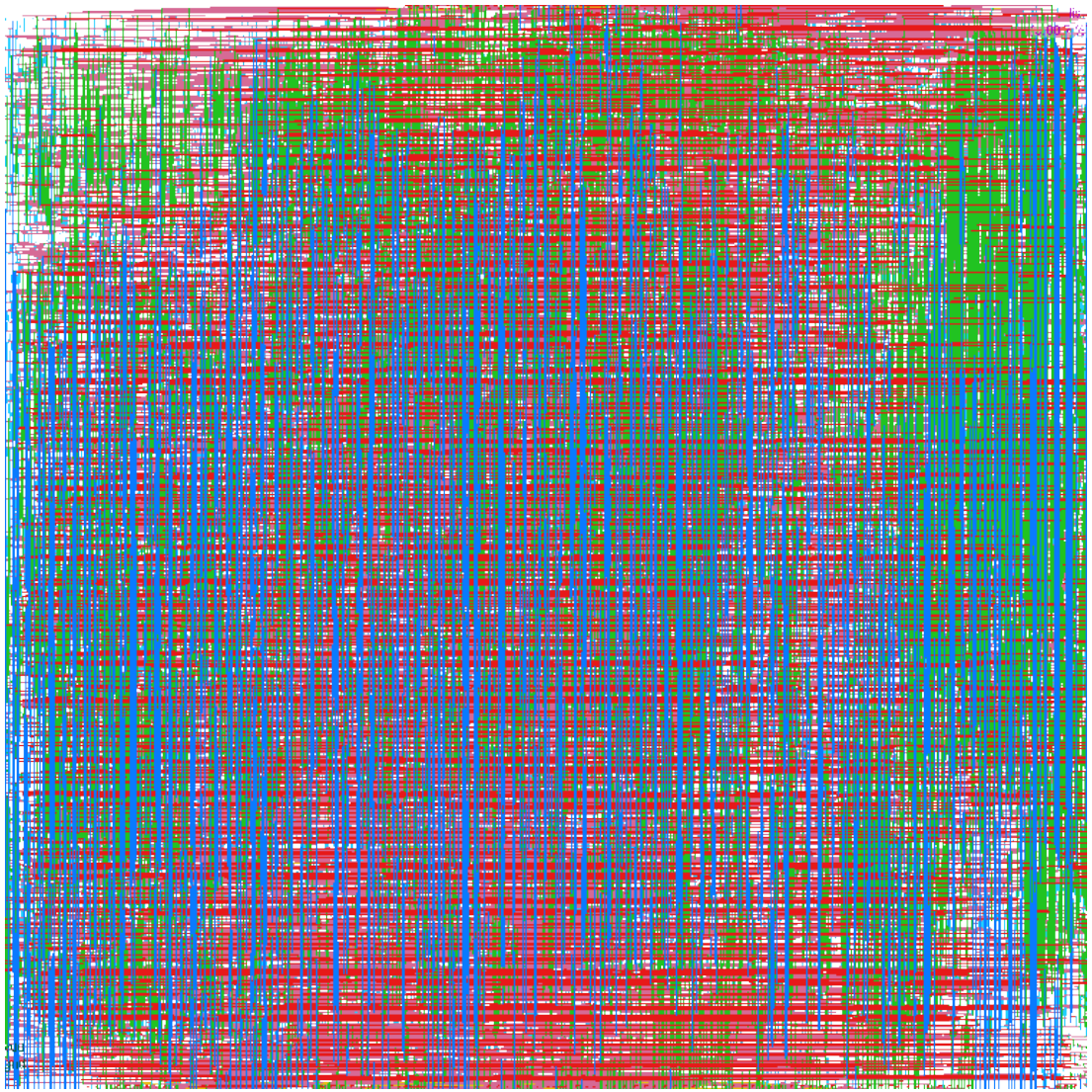


objective: decrease power loss

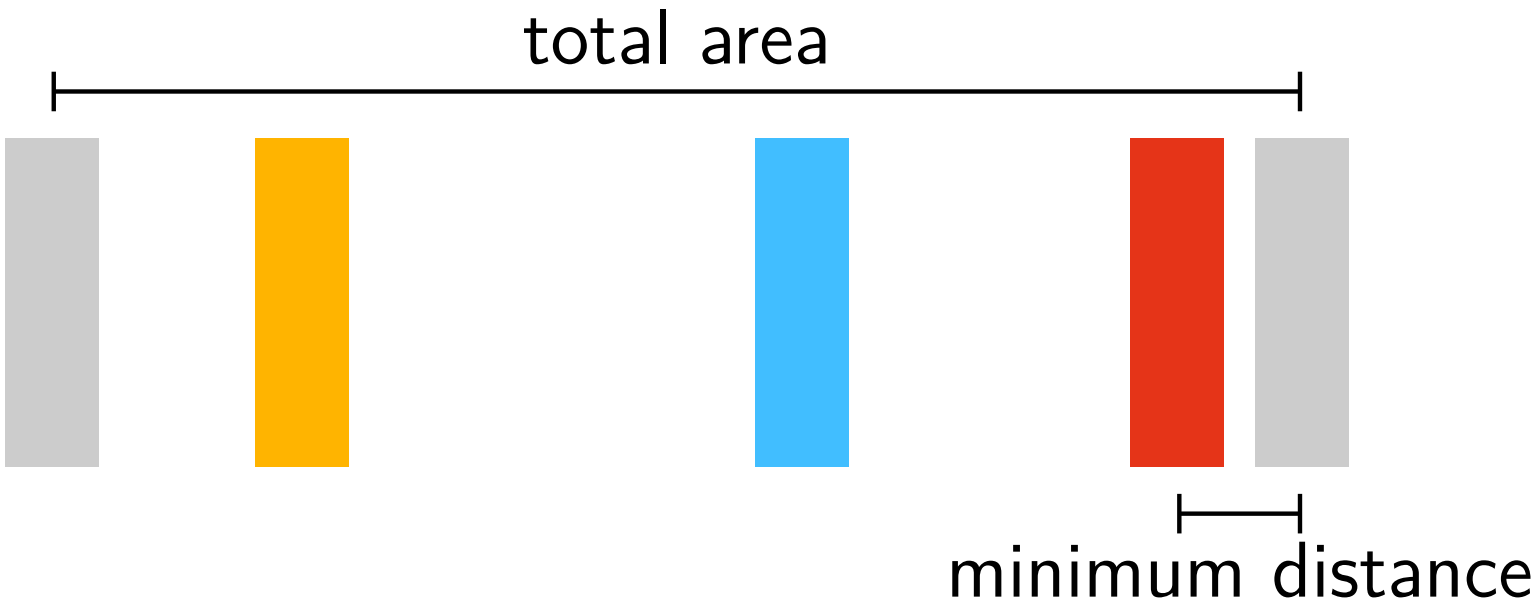
semiconductor circuit



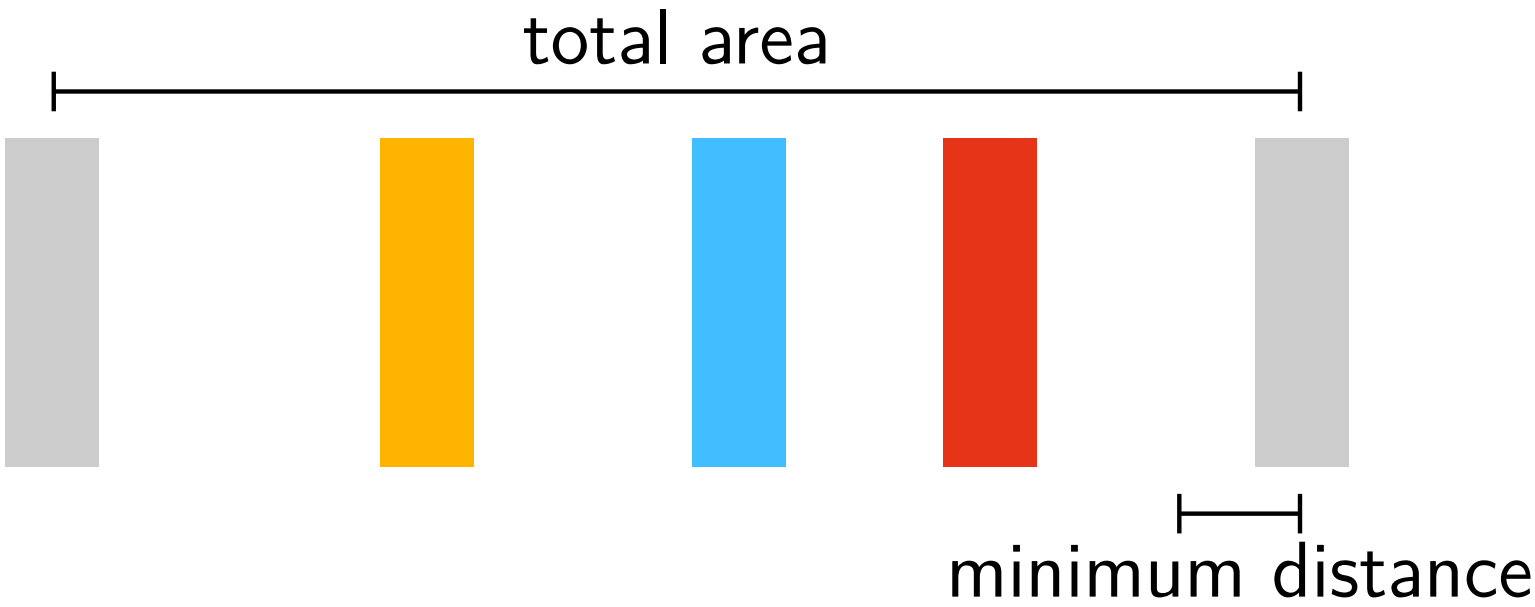
semiconductor circuit



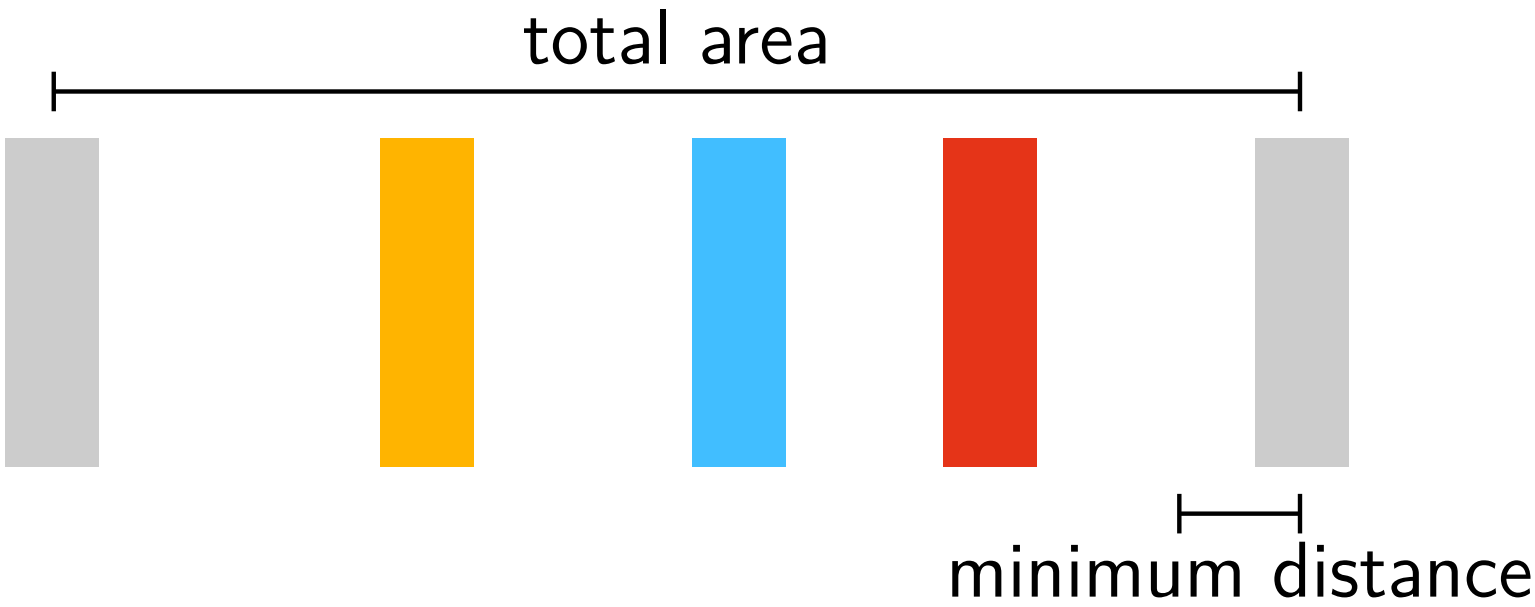
schematic model



schematic model



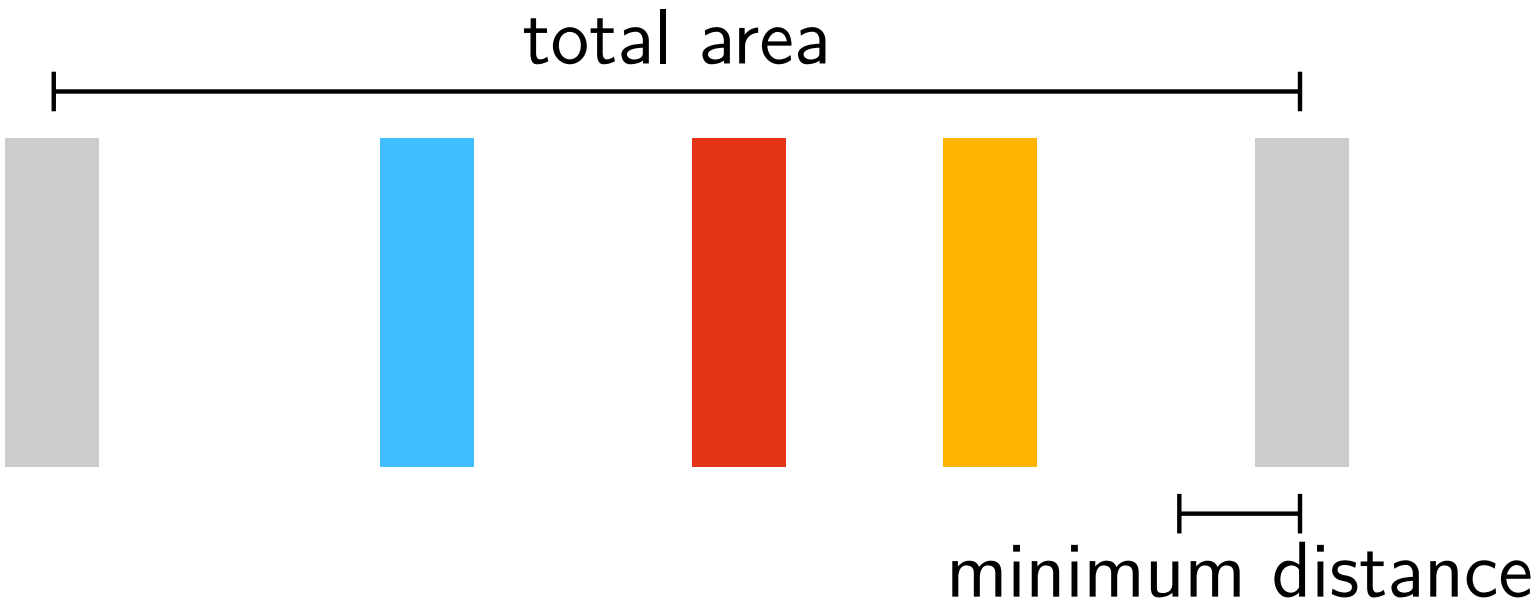
schematic model



questions

- optimal distances?

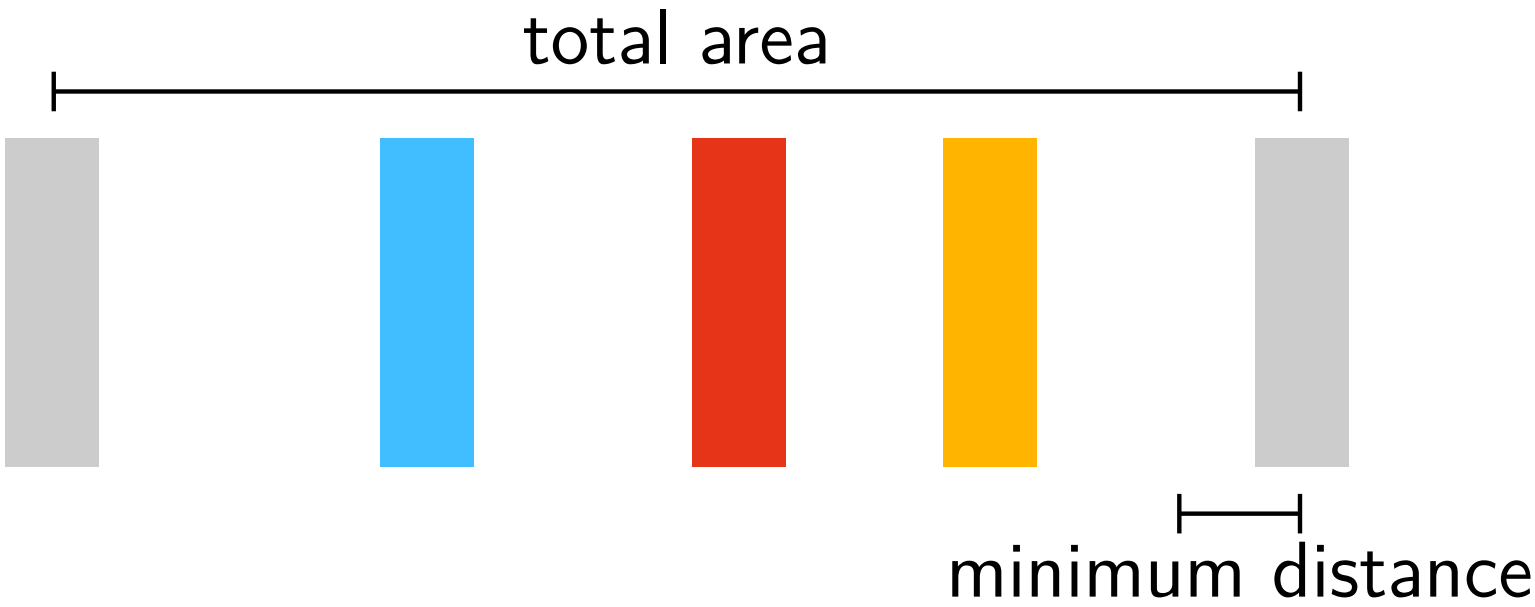
schematic model



questions

- optimal distances?

schematic model



questions

- optimal distances?
- optimal order?

model characteristics

initial approach

- nonlinear objective function
- continuous and integer variables
- hundreds of wires

model characteristics

initial approach

- nonlinear objective function
- continuous and integer variables
- hundreds of wires

structural analysis

- spacing and ordering ➡ two subproblems
- spacing analytically solvable

model characteristics

initial approach

- nonlinear objective function
- continuous and integer variables
- hundreds of wires

structural analysis

- spacing and ordering ➡ two subproblems
- spacing analytically solvable
- ordering turns into special TSP
- even TSP efficiently solvable(!)

enhanced models

- correlation in wire switches (simultaneous switches)
- more complicated objective (timing, crosstalk, etc.)
- deplacement costs
- interaction with more than two neighboring wires

enhanced models

- correlation in wire switches (simultaneous switches)
- more complicated objective (timing, crosstalk, etc.)
- deplacement costs
- interaction with more than two neighboring wires

significant structural changes

➡ much harder to solve

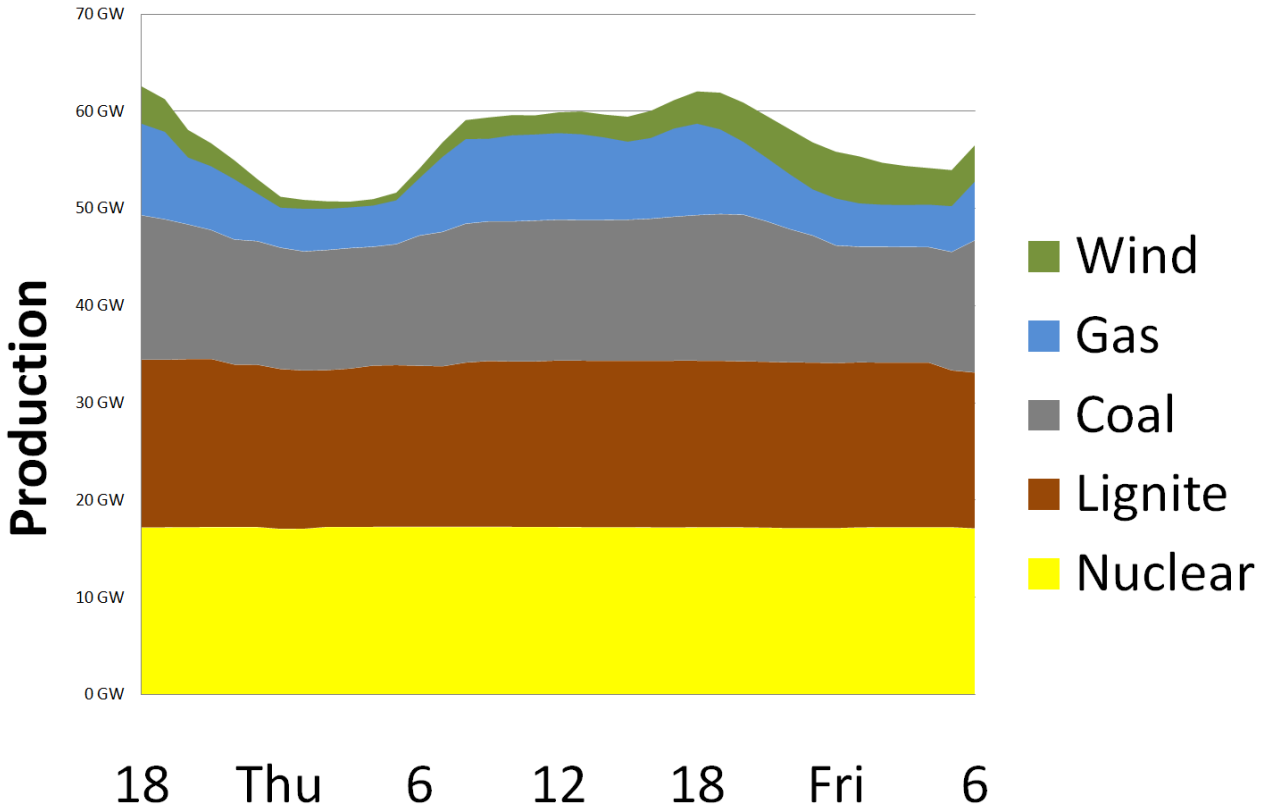
modelling the power market

- renewable energy production grows rapidly
- especially production from wind units

modelling the power market

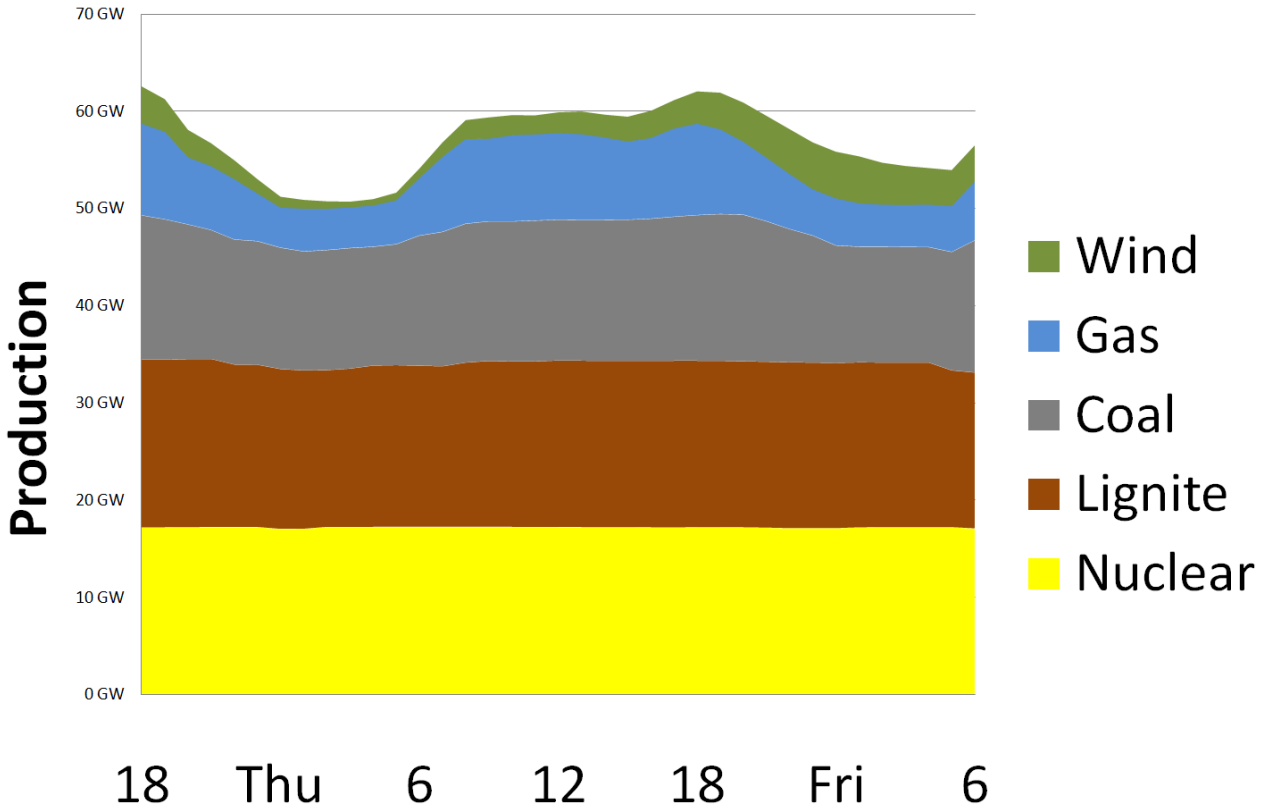
- renewable energy production grows rapidly
- especially production from wind units
- but: production fluctuates heavily

power production



- cheap units are used for base load
- fast but expensive units are used for peak load

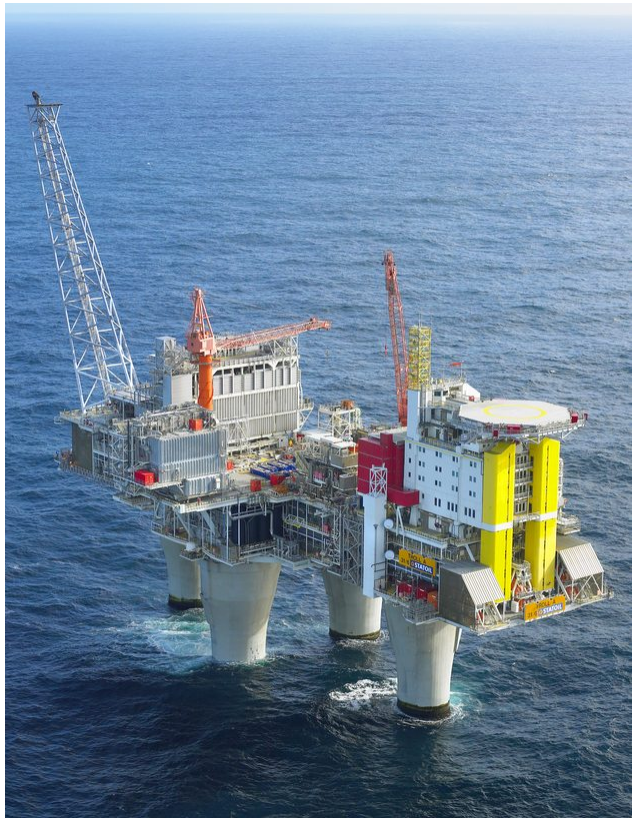
power production



- production of wind units fluctuates heavily
- more wind units in the future intensify situation

problem

- expectation: more fast gas units needed
- how many, at which price?



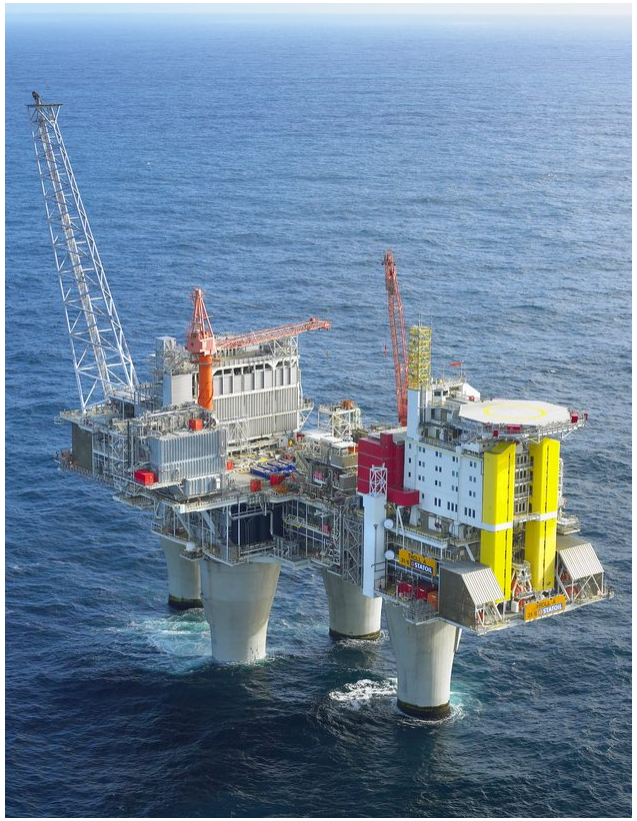
problem

- expectation: more fast gas units needed
- how many, at which price?

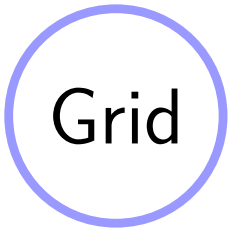
⬇

- goal: quantify competitive advantage of gas units

  ➡ model the power market



model

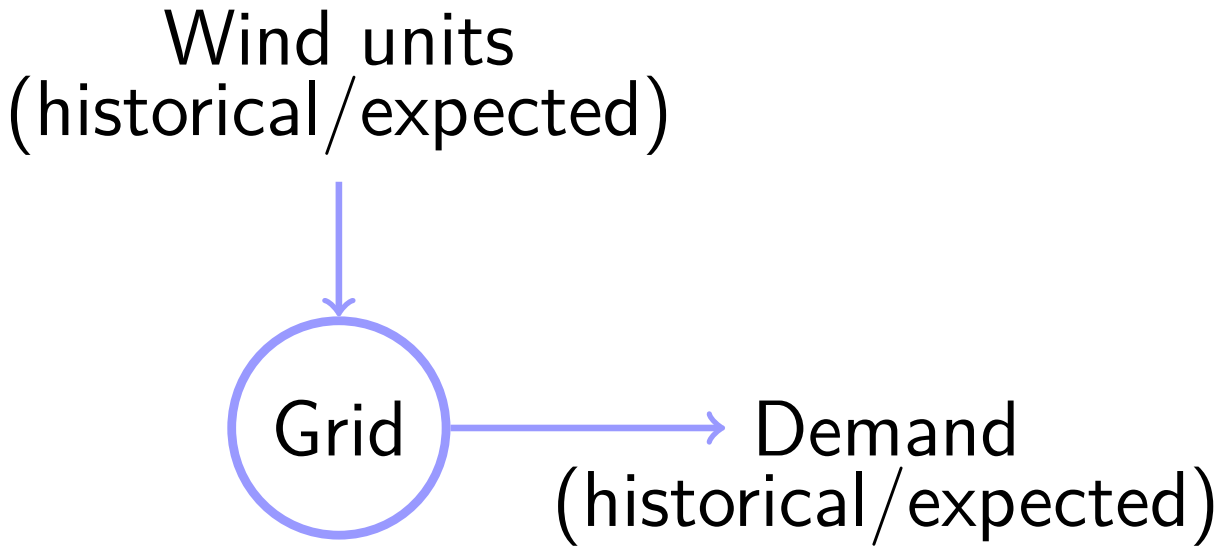


Unit Commitment formulation

model

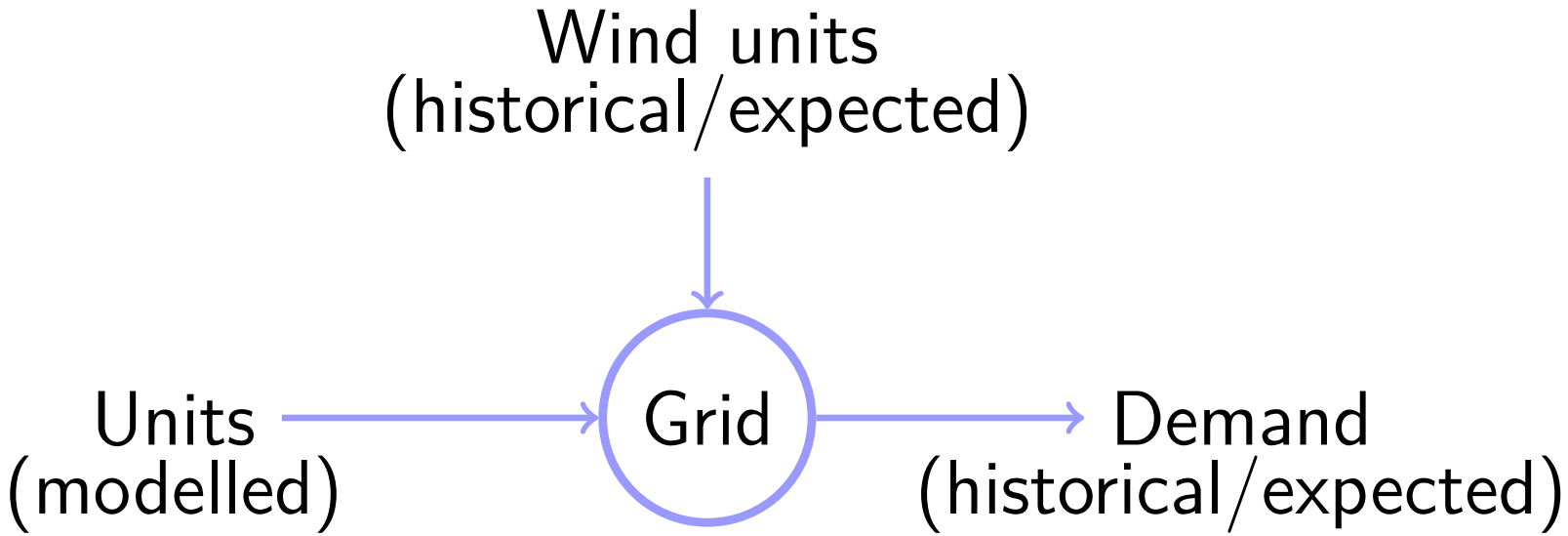
Unit Commitment formulation

model



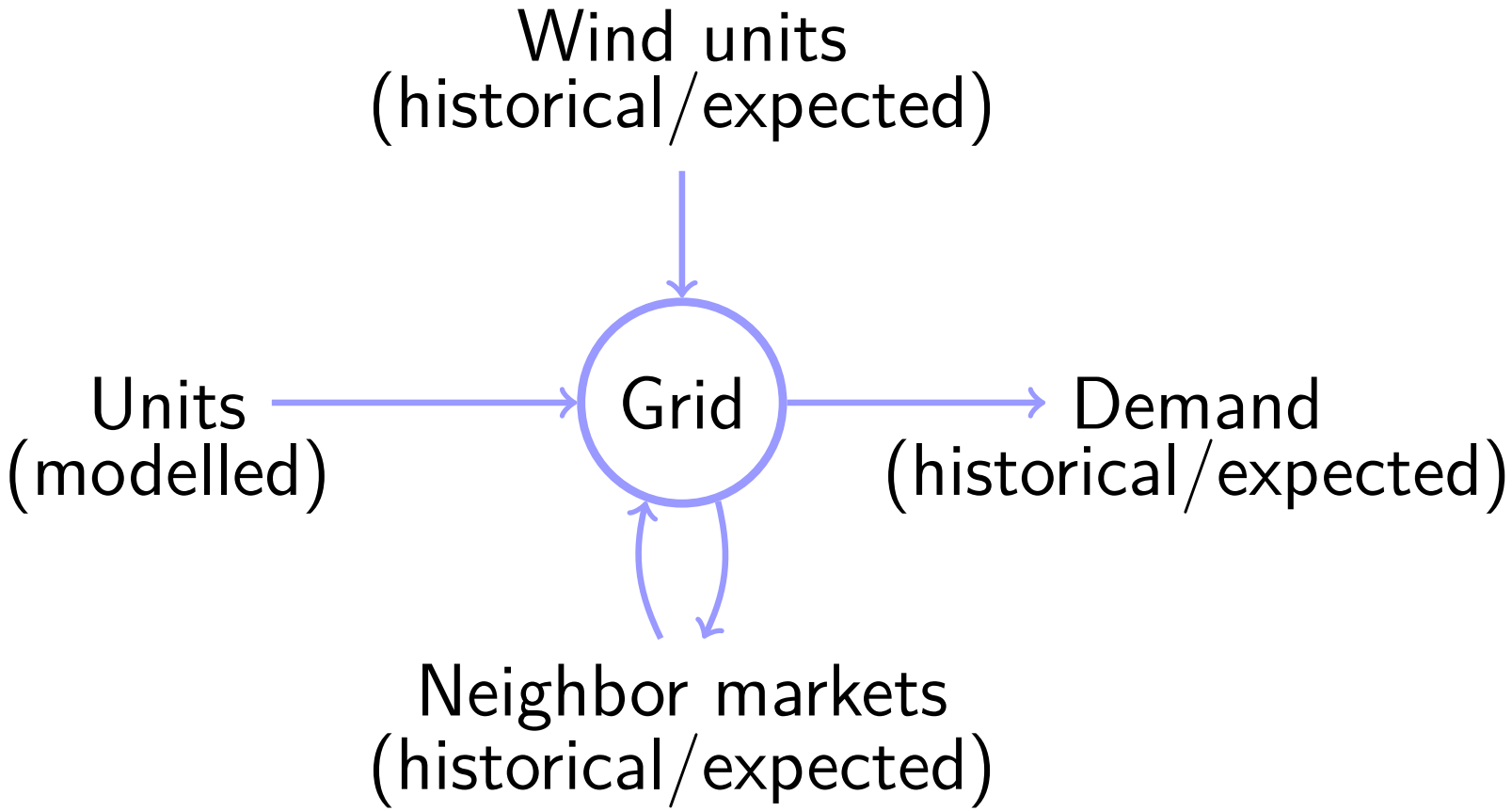
Unit Commitment formulation

model



Unit Commitment formulation

model



Unit Commitment formulation

implementation

- completely in Xpress Mosel

implementation

- completely in Xpress Mosel
- flexible data sources and targets
  - ➡ Oracle database
  - ➡ Excel worksheet $\rightarrow$ frontend for non-programmers

implementation

- completely in Xpress Mosel
- flexible data sources and targets
  - ➡ Oracle database
  - ➡ Excel worksheet $\rightarrow$ frontend for non-programmers
- data preprocessing
  - ➡ real-world parameters $\rightarrow$ model parameters
  - ➡ substantial reduction of constraints
- no customization of Xpress Optimizer needed