Decentralized provision of active power by means of dynamic virtual power plants

Michael Sonnenschein

Sebastian Beer, Jörg Bremer,

Christian Hinrichs, Astrid Nieße

University of Oldenburg – Environmental Informatics &
OFFIS Institute for Information Technologie – R&D Division Energy

Research Cluster Smart Nord

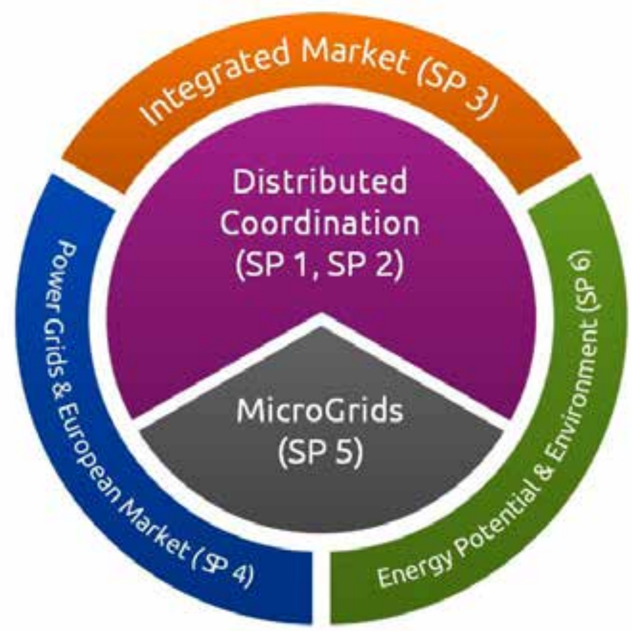
Decentralised, agent based methods for control of medium and low voltage (smart) grids

Stability of medium and low voltage grids considering decentralized supply

Partners:
Uni Oldenburg, Uni Hannover,
TU Braunschweig, TU Clausthal,
NEXT ENERGY, OFFIS

Funding
4.1 M€ by Federal State of Lower Saxony

Duration
3/2012 – 2/2015



Dynamic Virtual Power Plants

DVPP
MOTIVATION AND CONCEPT

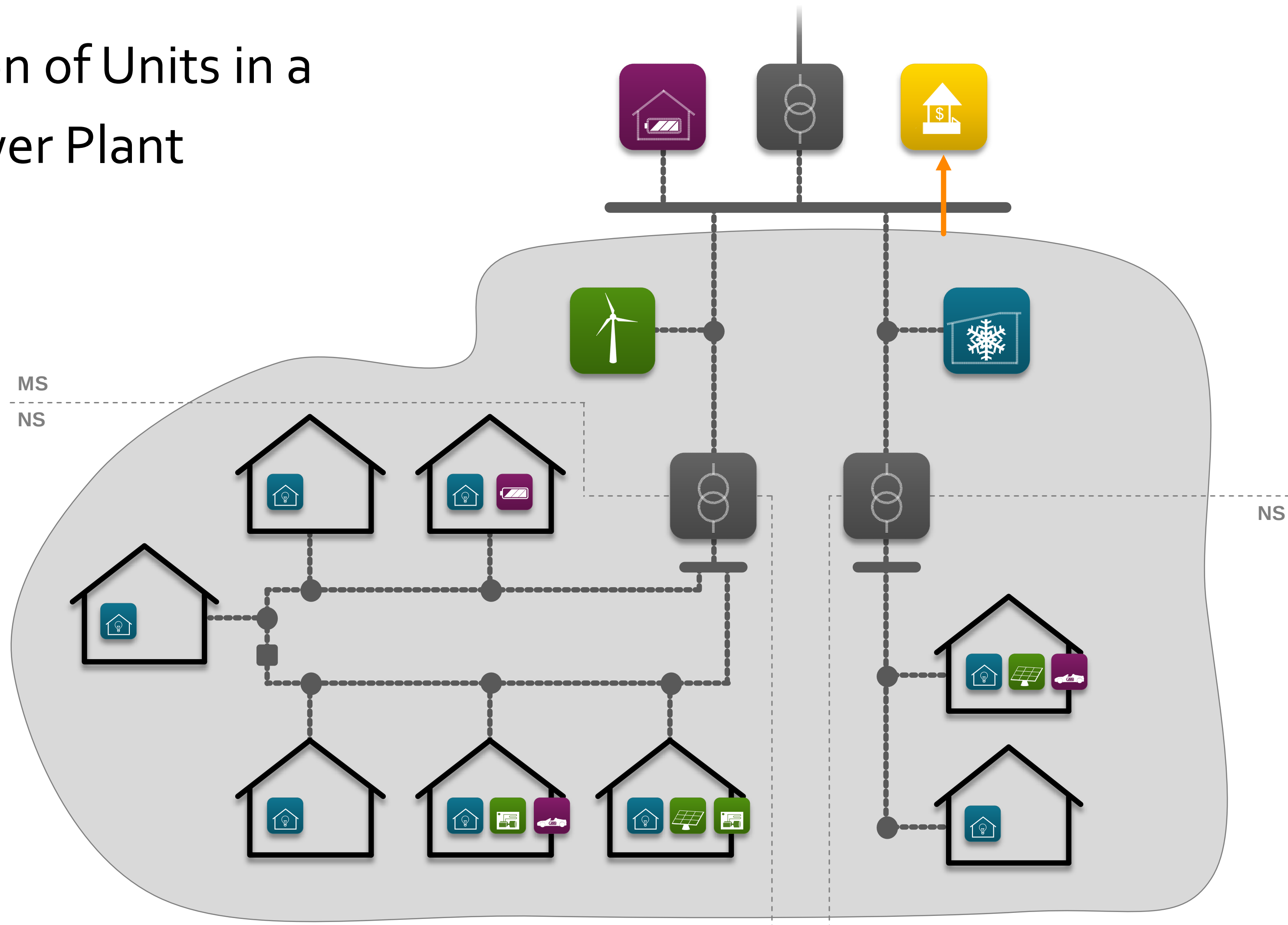
Post EEG challenges for renewables

- **Renewable energy has to be traded at a power market**
- Requirement: Enhance supply of single renewable energy sources
  - Market barriers at the EEX (currently min. 100kW)
  - Reliability of supply regarding an agreed schedule (product)
- Smart Nord: *Dynamic, distributed* aggregation and control of units to adapt to
  - Demand traded at the market
  - Current set of active units in the system
  - Forecast of supply of single units

Virtual Power Plants (VPP)

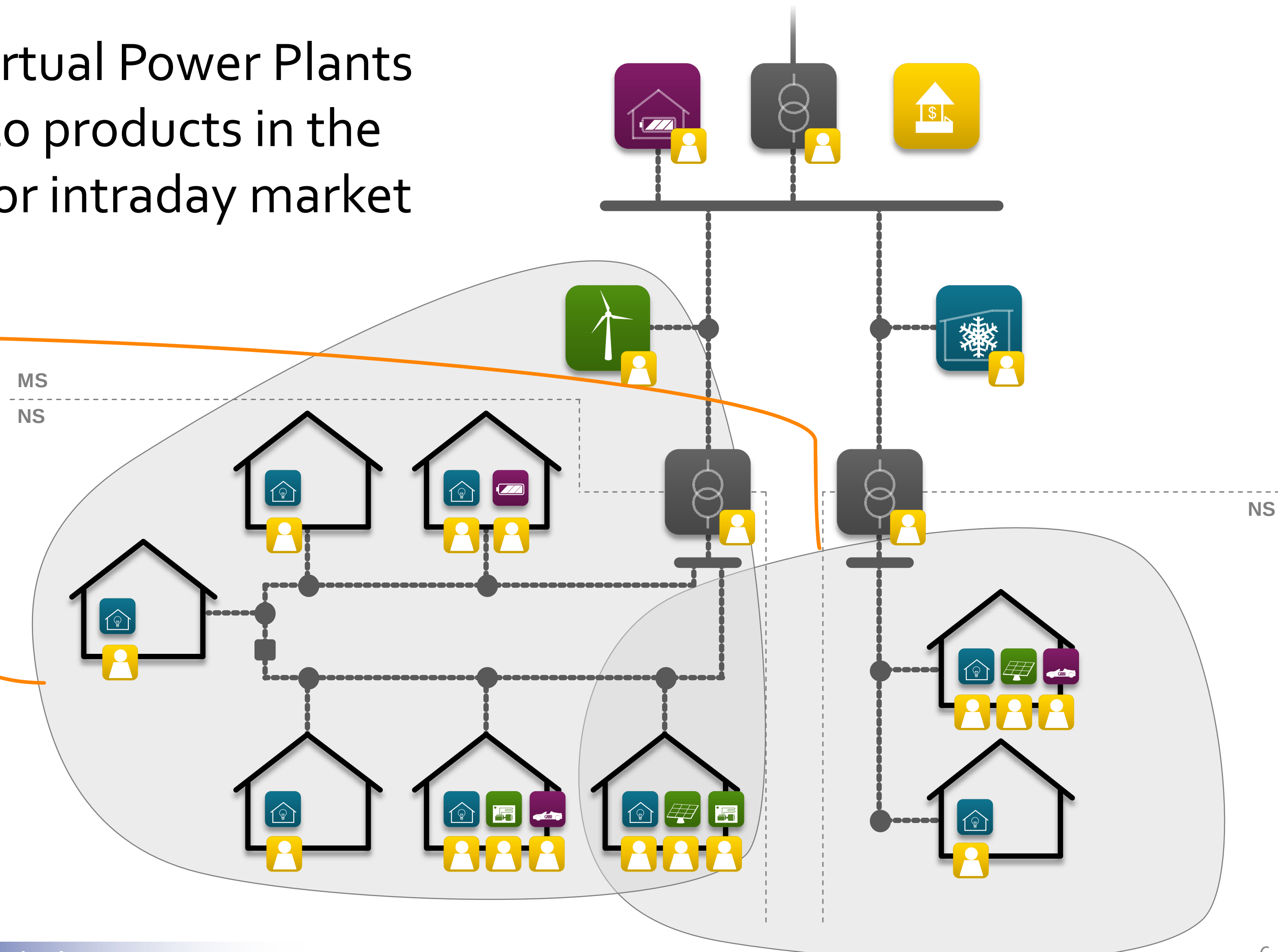
Day Ahead Energy Market

Aggregation of Units in a Virtual Power Plant



Dynamic Virtual Power Plants (DVPP)

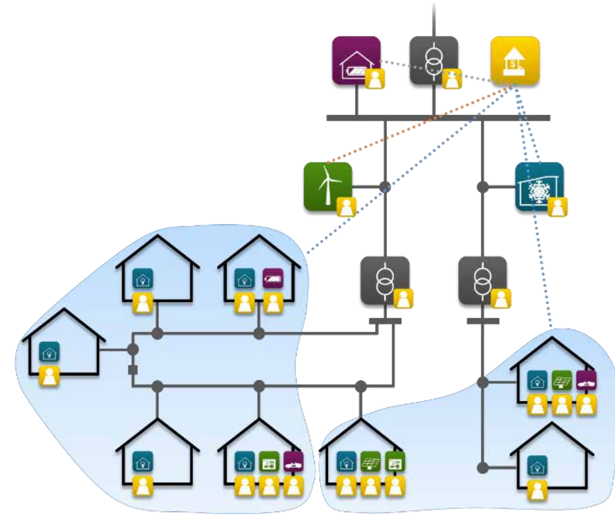
Dynamic Virtual Power Plants dedicated to products in the day ahead or intraday market

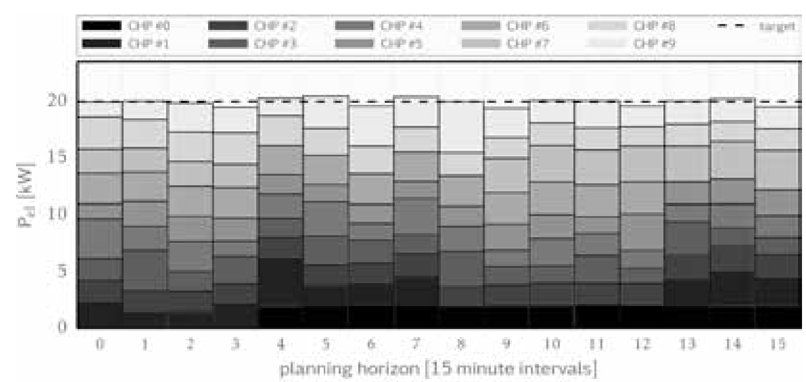


6

Coordination of DVPPs

- **DVPP setup:**
  Coalition formation of plants for trading a specific power product
- **Predictive scheduling:**
  Determine a schedule which
  - guarantees the successful bid
  - and respects all individually defined degrees of freedom
- **Continuous scheduling:**
  Reschedule if unforeseen changes occur
- **Payoff division:**
  Distribute the gained revenues just and reasonable to the coalition members





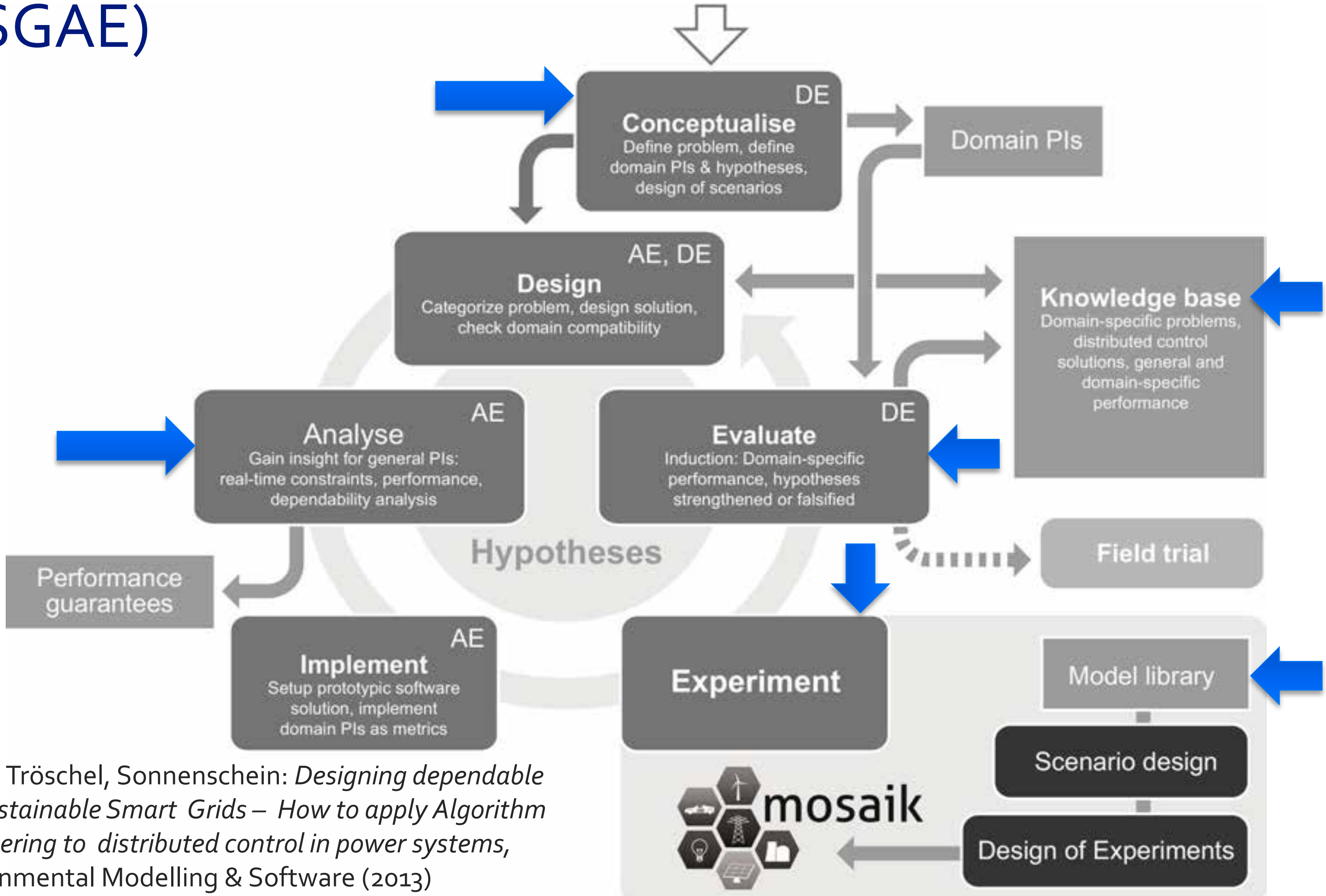
Decentralized, self-organized control

- Dynamic aggregation reacting to market conditions
- Units possibly distributed to several parts of the grid: no control centre, no micro grid approach
- No knowledge about number of units: Scalabity of control methods
- Ability to reconfigure dynamically in case of failures: Robustness regarding failure of single units

→ Decentralized, self-organized control

- Security

Smart Grid Algorithm Engineering (SGAE)



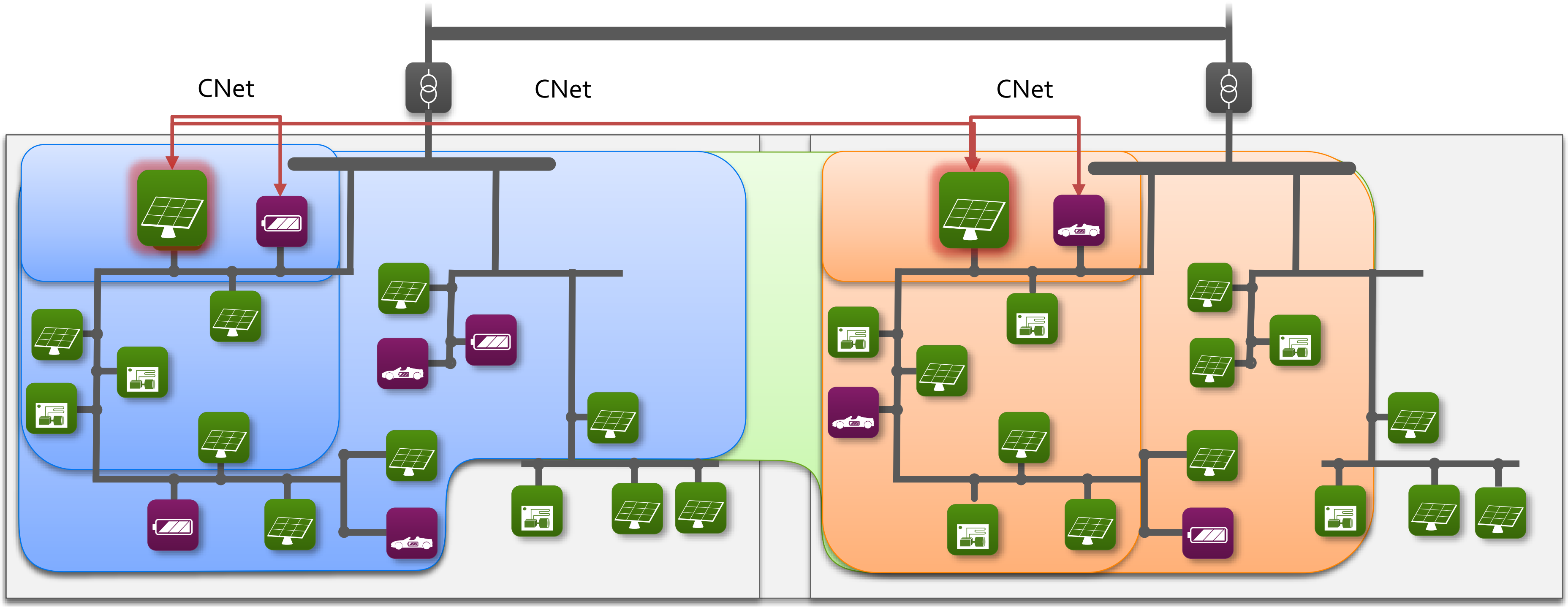
Nieße, Tröschel, Sonnenschein: *Designing dependable and sustainable Smart Grids – How to apply Algorithm Engineering to distributed control in power systems,* Environmental Modelling & Software (2013)

Forming coalitions to trade power at the market

DVPP SETUP

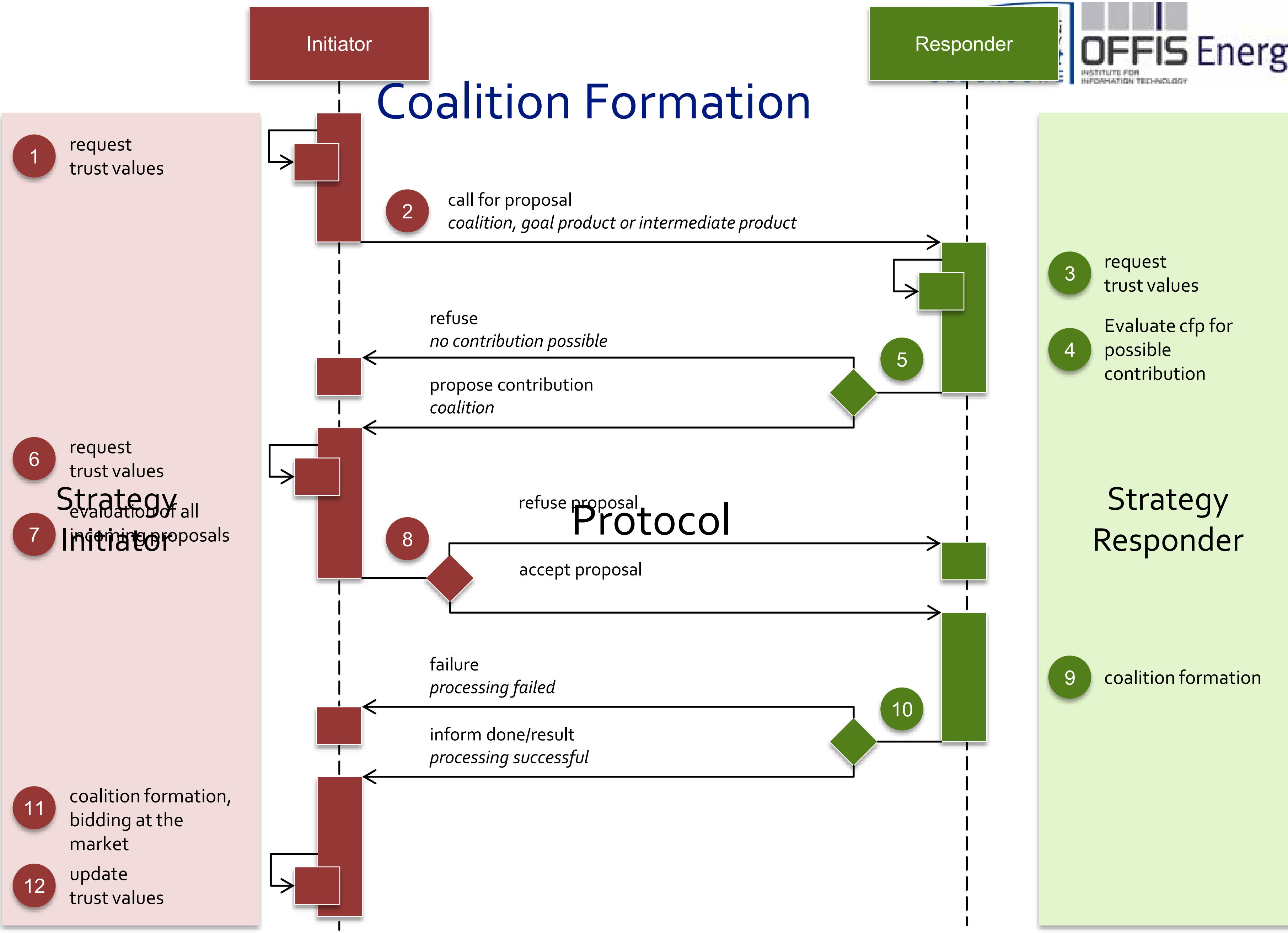
DVPP setup

Goal: deliver 100kW from 1 p.m. to 2 p.m.



1. Selection of Neighborhood
2. Coalition Formation
3. Expansion of Neighborhood

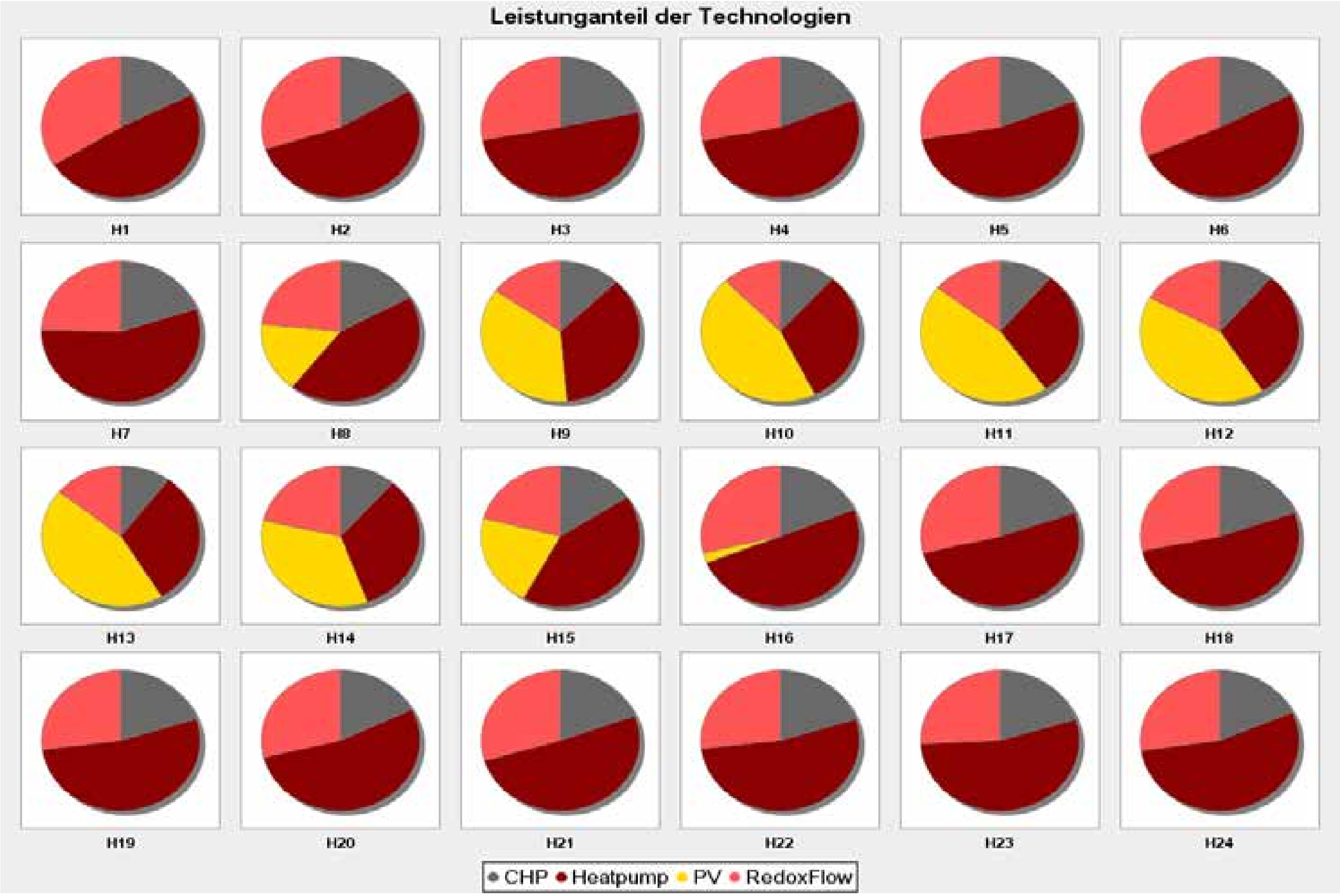
Coalition Formation



Evaluation Scenario 2030

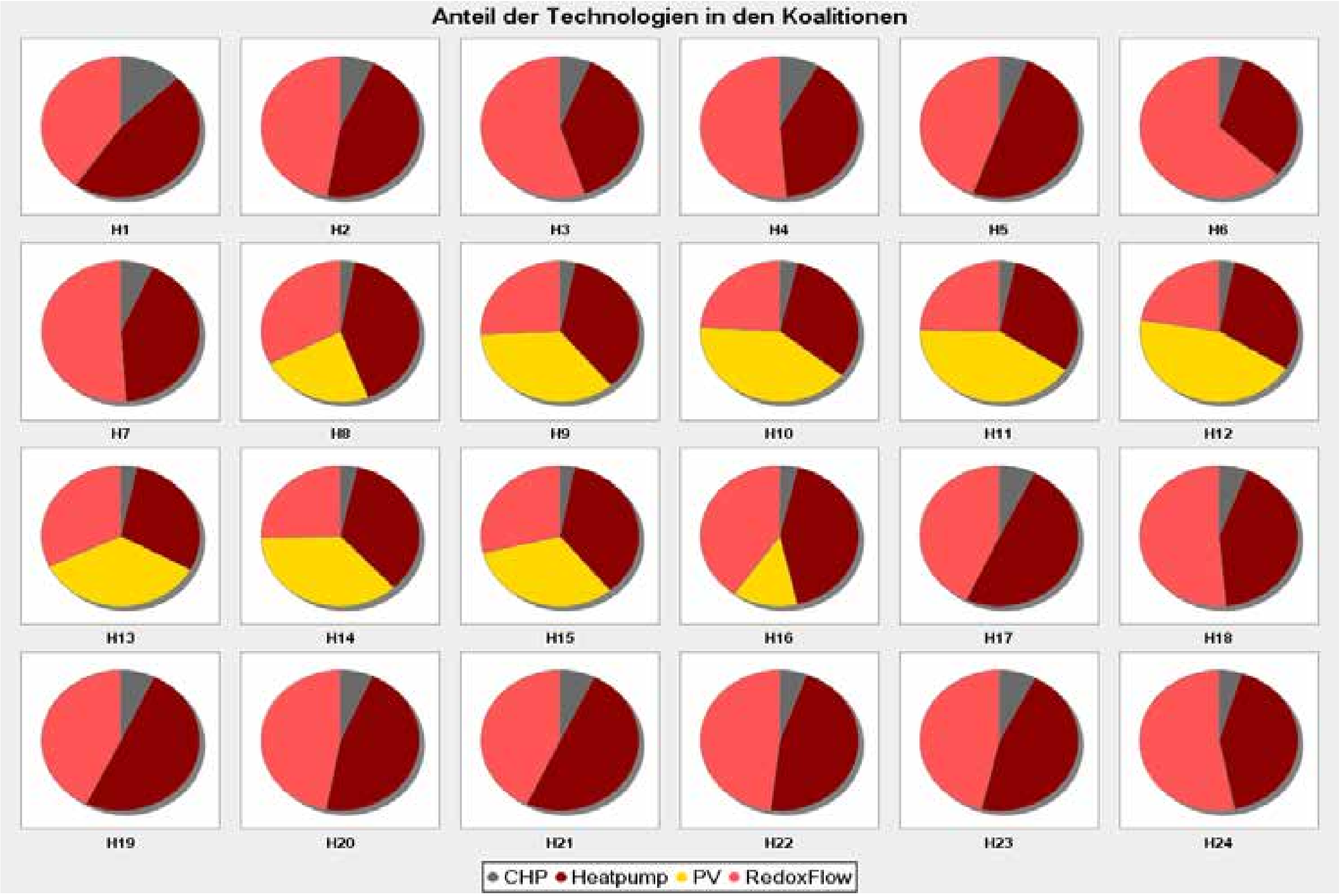
- Simulated Units
  - 8806 households
  - 789 heat pumps
  - 1048 photovoltaic systems
  - 28 wind power generators
  - 122 CHPs
  - 789 redox-flow batteries
- Multi-agent system
  - heat pumps, PVs, CHPs, batteries controlled
  - demand of households, wind power not controlled
  - 2710 units for coalition formation
- Exemplary day: January 31st
- Coalitions form for hourly products day ahead

Configuration, Size, and Technology Profile of Coalitions

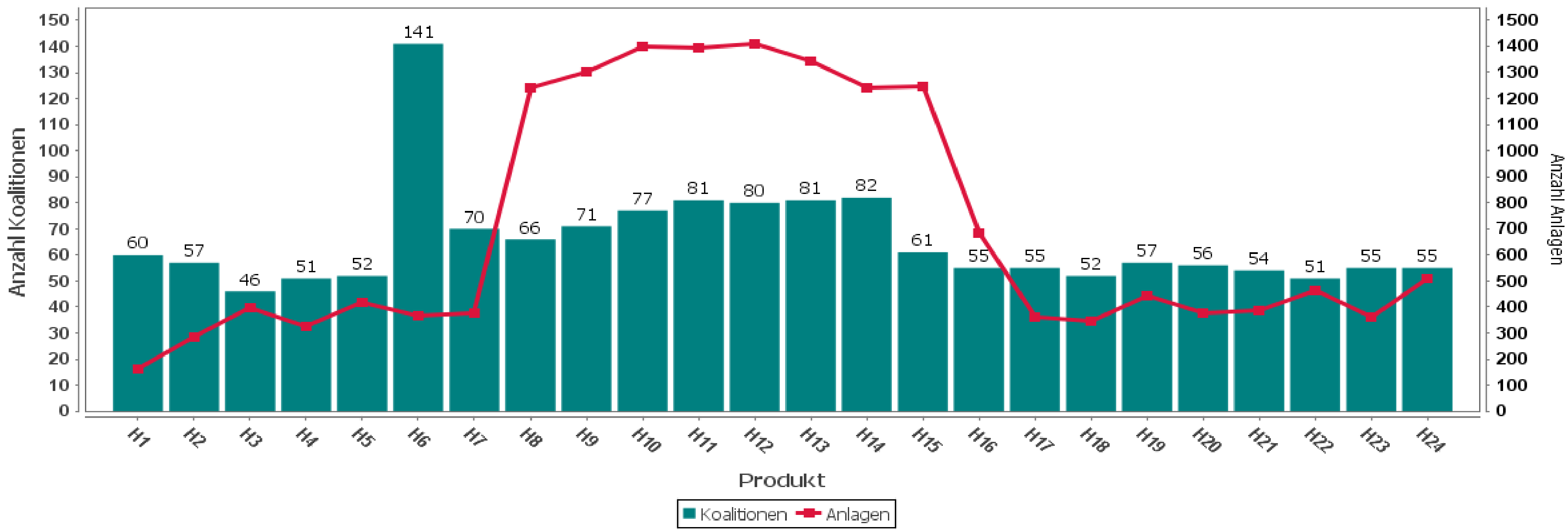


14

Configuration, Size, and Technology Profile of Coalitions



Configuration, Size, and Technology Profile of Coalitions



Planning schedules for units in a DVPP

PREDICTIVE SCHEDULING

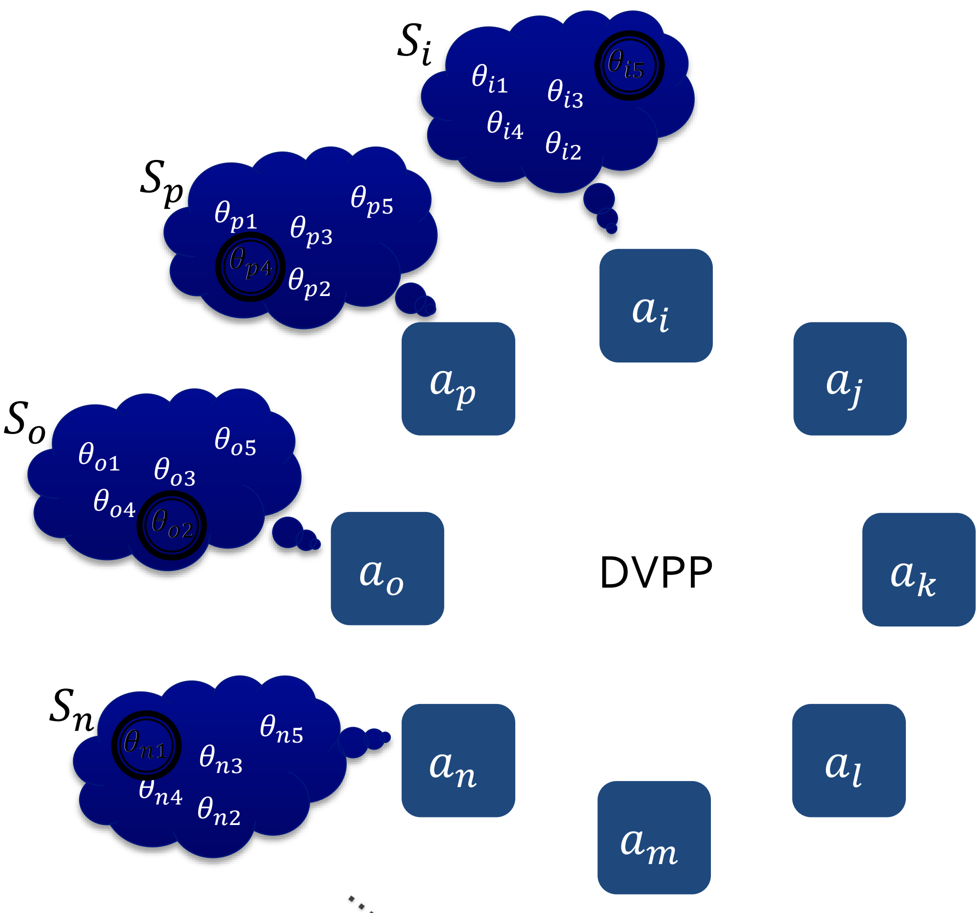
Predictive Planning of Operation Schedules

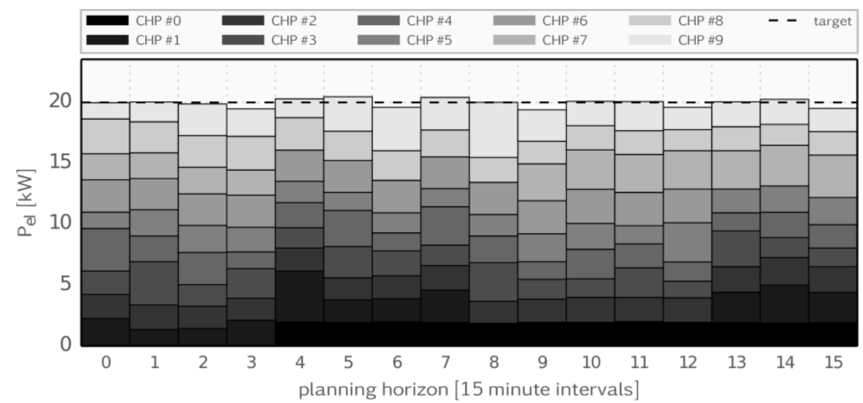
Goal:

Select an optimal schedule for each unit in a DVPP

- with respect to the agreed power product
- with respect to each unit's private preferences
- in a completely decentralized manner

Predictive Planning as Combinatorial Optimization Problem





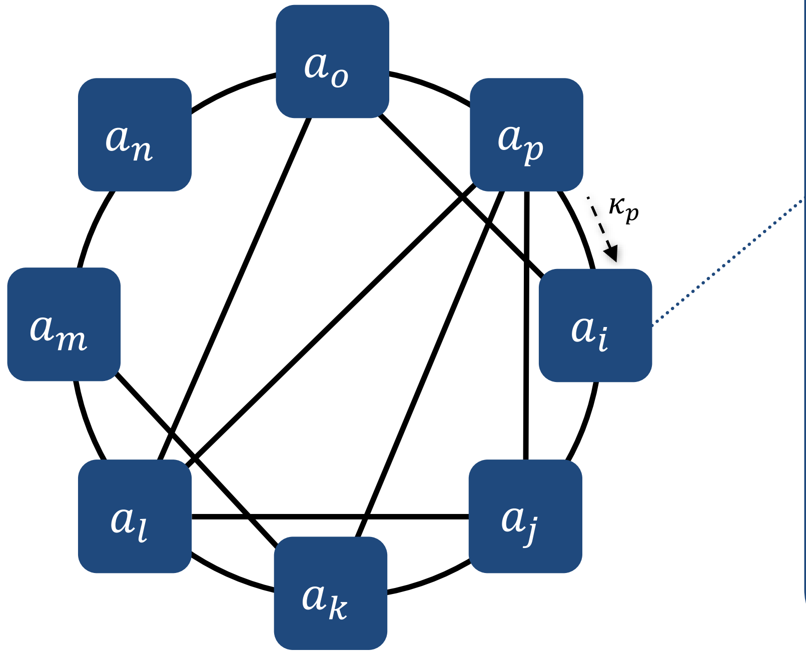
$$\min \left\| \zeta - \sum_{i=1}^{|A|} \sum_{j=1}^{|S_i|} (\theta_{ij} \cdot x_{ij}) \right\|_1$$

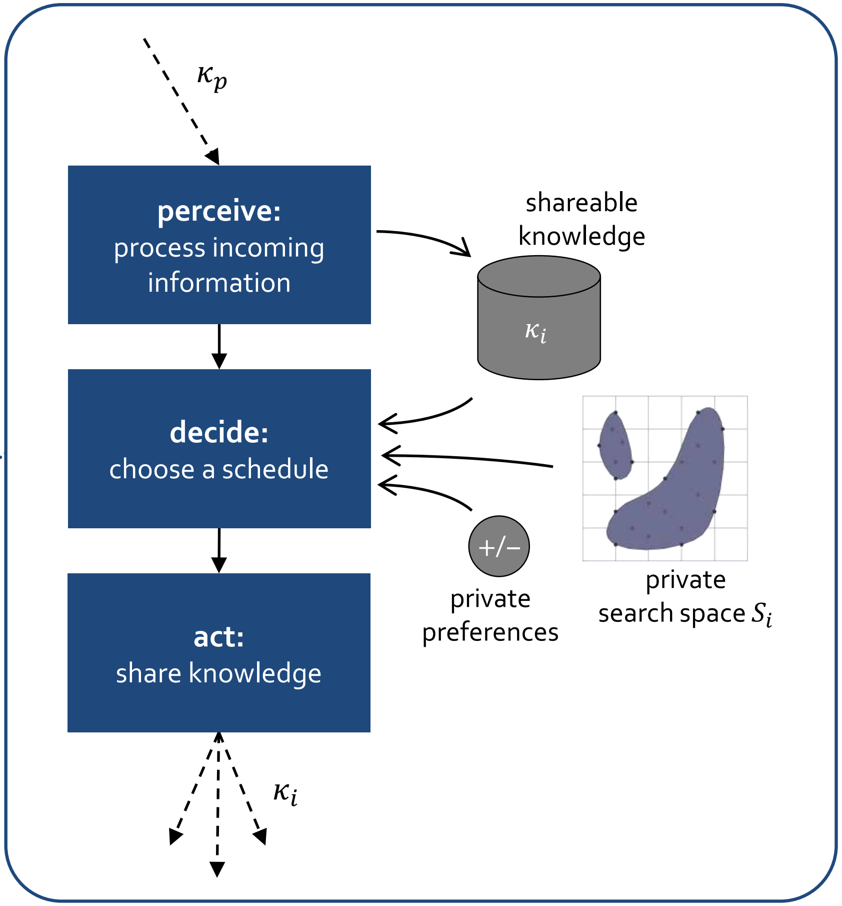
$$\text{subject to} \quad \sum_{j=1}^{|S_i|} x_{ij} = 1\,, \quad x_{ij} \in \{0,1\}\,, \quad i = 1 \ldots |A|$$

COHDA

Combinatorial Optimization Heuristic for Distributed Agents

1. Construction of Overlay Communication Topology
2. Agents react asynchronously to exchanged messages





COHDA – Properties

Convergence and **termination** formally proved using convergence stairs:

- $\mathcal{A}_1$: Working memories of all agents are complete.
- $\mathcal{A}_2$: A final solution candidate is found.
- $\mathcal{A}_3$: Working memories of all agents are identical; heuristic terminates.

After initial setup ($\mathcal{A}_1$), COHDA shows the **anytime** property:

- Newly produced solutions are always better than older solutions.

Run-Time properties:

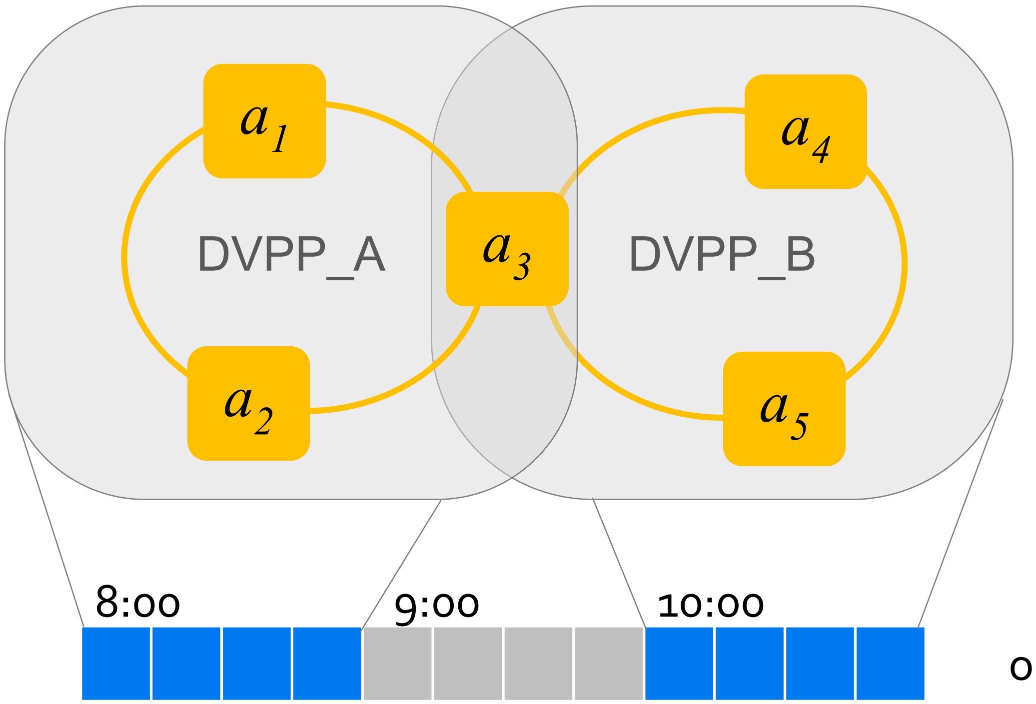
- Best-Case: $\mathcal{O}(3)$
- Worst-Case: $\mathcal{O}(n^m)$
- Empirically: $\mathcal{O}(\log m)$

(m = number of agents,
n = avg. size of search spaces)

Algorithm 1 Behavior of an agent a_i in the COHDA heuristic.

```
Local attributes:
    S_i                                      ▷ Set of feasible schedules
    λ_i                                      ▷ Schedule selection counting variable
    κ_i = (ζ, Ω_i, γ_i)                      ▷ Working memory: target, system config, solution candidate

Objective functions:
    f(γ), f(Ω)                               ▷ Rates γ or Ω according to the global objective function
    ACCEPT(θ)                                ▷ TRUE, iff θ acceptable according to local objectives

Behavior:
if receiving message κ_j then
    ζ_j ← ζ ∈ κ_j                            ▷ Extract target profile from message
    Ω_j ← Ω_j ∈ κ_j                          ▷ Extract system configuration from message
    γ_j ← γ_j ∈ κ_j                          ▷ Extract solution candidate from message
    κ'_i ← (ζ, Ω_i, γ_i)                     ▷ Create backup of own working memory
    ω_i ← ω_i ∈ Ω_i                          ▷ Extract own schedule selection from Ω_i
    K_i ← {a_k | ω_k ∈ Ω_i}                  ▷ Extract agent identifiers from γ_i
    K_j ← {a_k | ω_k ∈ Ω_j}                  ▷ Extract agent identifiers from γ_j

    if ζ = NONE then                         ▷ Store target profile
        ζ ← ζ_j
    end if

    for ω_k ∈ Ω_i, ω'_k ∈ Ω_j do             ▷ Update Ω_i
        if ∀ω_x ∈ Ω_i : x ≠ k or λ_k ∈ ω'_k > λ_k ∈ ω_k then
            Ω_i ← (Ω_i \ {ω_k}) ∪ {ω'_k}
        end if
    end for
                                             ▷ Replace/Extend solution candidate...
    if K_i ⊂ K_j then                        ▷ ...if the new one is larger
        γ_i ← γ_j
    else if K_i ⊄ K_j then                   ▷ ...or if it contains entries from unknown agents
        γ_i ← (a_i, {ω_k ∈ γ_i} ∪ {ω_k | a_k ∈ K_j − K_i})
    else if f(γ_j) > f(γ_i) then             ▷ ...or if it rates better
        γ_i ← γ_j
    else if f(γ_j) = f(γ_i) and a_x ∈ γ_j > a_y ∈ γ_i then   ▷ ...or to break ties
        γ_i ← γ_j
    end if

    ω'_i ← ω_i ∈ γ_i                         ▷ Extract own schedule selection from γ_i
    if ∃θ ∈ S_i : f((Ω_i \ {ω_i}) ∪ {(a_i, θ, λ_i + 1)}) > f(γ_i) and ACCEPT(θ) then   ▷ Better schedule found?
        ω'_i ← (a_i, θ, λ_i + 1)             ▷ Create new schedule selection
        γ_i ← (a_i, (Ω_i \ {ω_i}) ∪ {ω'_i})  ▷ Create new solution candidate
    else if ω_i ≠ ω'_i then                  ▷ Historical schedule chosen?
        ω'_i ← (a_i, θ ∈ ω'_i, λ_i + 1)      ▷ Create new schedule selection using the historical schedule
    end if
    Ω_i ← (Ω_i \ {ω_i}) ∪ {ω'_i}             ▷ Update Ω_i with new schedule selection

    if (ζ, Ω_i, γ_i) ≠ κ'_i then             ▷ Publish working memory to neighbours
        send (ζ, Ω_i, γ_i) to neighbours
    end if

end if
```

COHDA for multiple DVPPs



operation schedule of agent a_3

Evaluation of solution candidate only considering DVPP under negotiation

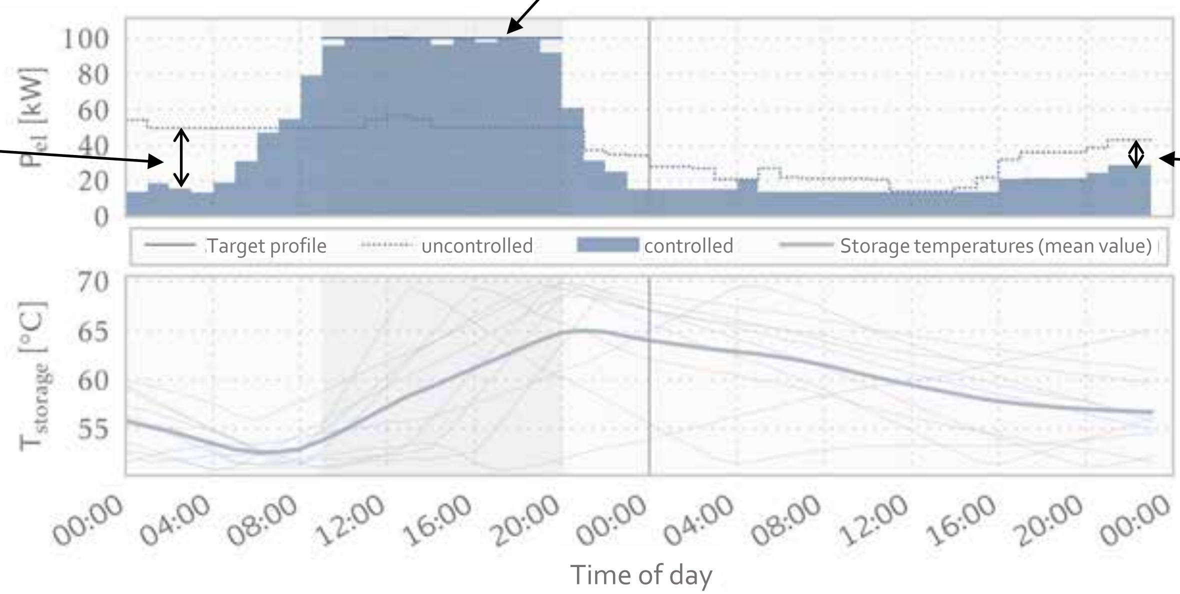
Example: EPEX Spot

Scenario:

- 10 CHP (Vaillant EcoPower 20.0)
- EPEX Spot, Block Product “Peak Load” (100 kW, 9 am – 8 pm)

98 % product coverage

significantly reduced operation before product delivery



slightly reduced operation afterwards

Coping large sets of feasible schedules

REPRESENTING FLEXIBITIES

Goals of representing flexibilites

Essential for the acceptance of planning algorithms: Feasibility of solutions!

How can individual flexibilities, secondary goals, preferences, cost, etc. be considered in optimization?

Representation of flexibilies

Goal:

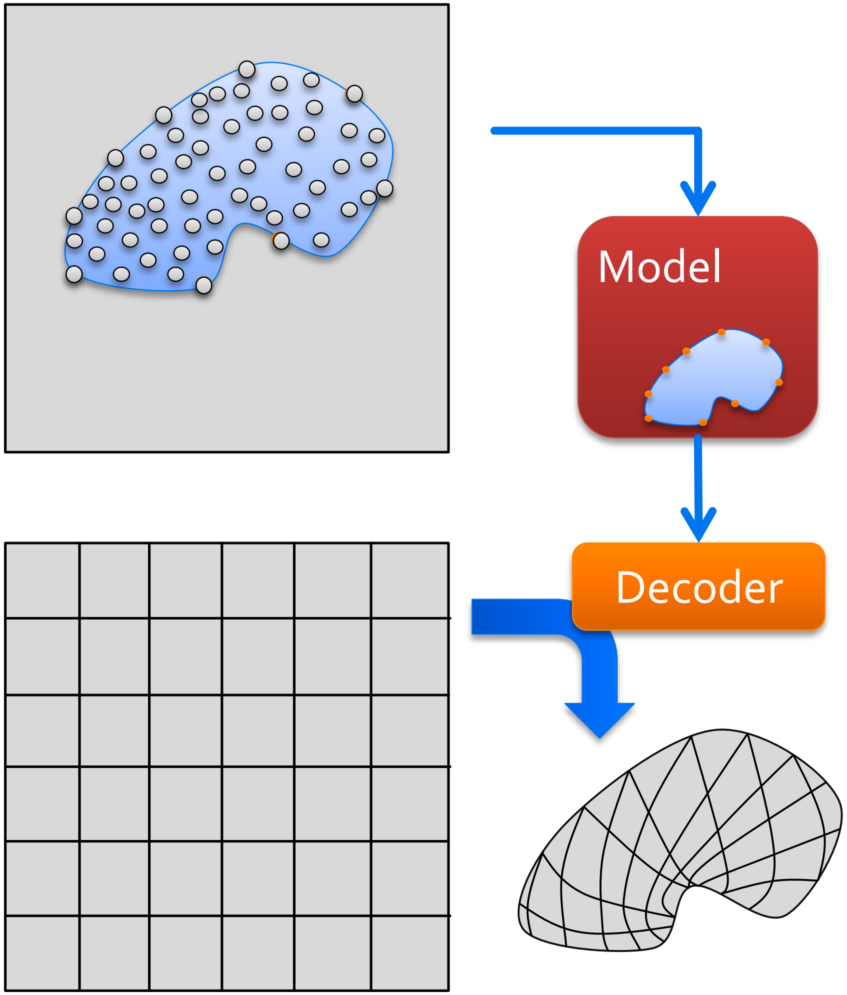
Representation of a unit´s search space must be

- independent of individual configuration of units
- efficiently accessable for optimisation processes
- allow the representation of additional features
- hide private information

Representation of schedules / flexibilies

Process chain

- Samples of feasible schedules simulated as patterns for search space
- Modelling of search space by SVM classifier
- Derivation of decoder function to map arbitrary schedules into the space of feasible schedules
  - Unconstrained optimisation



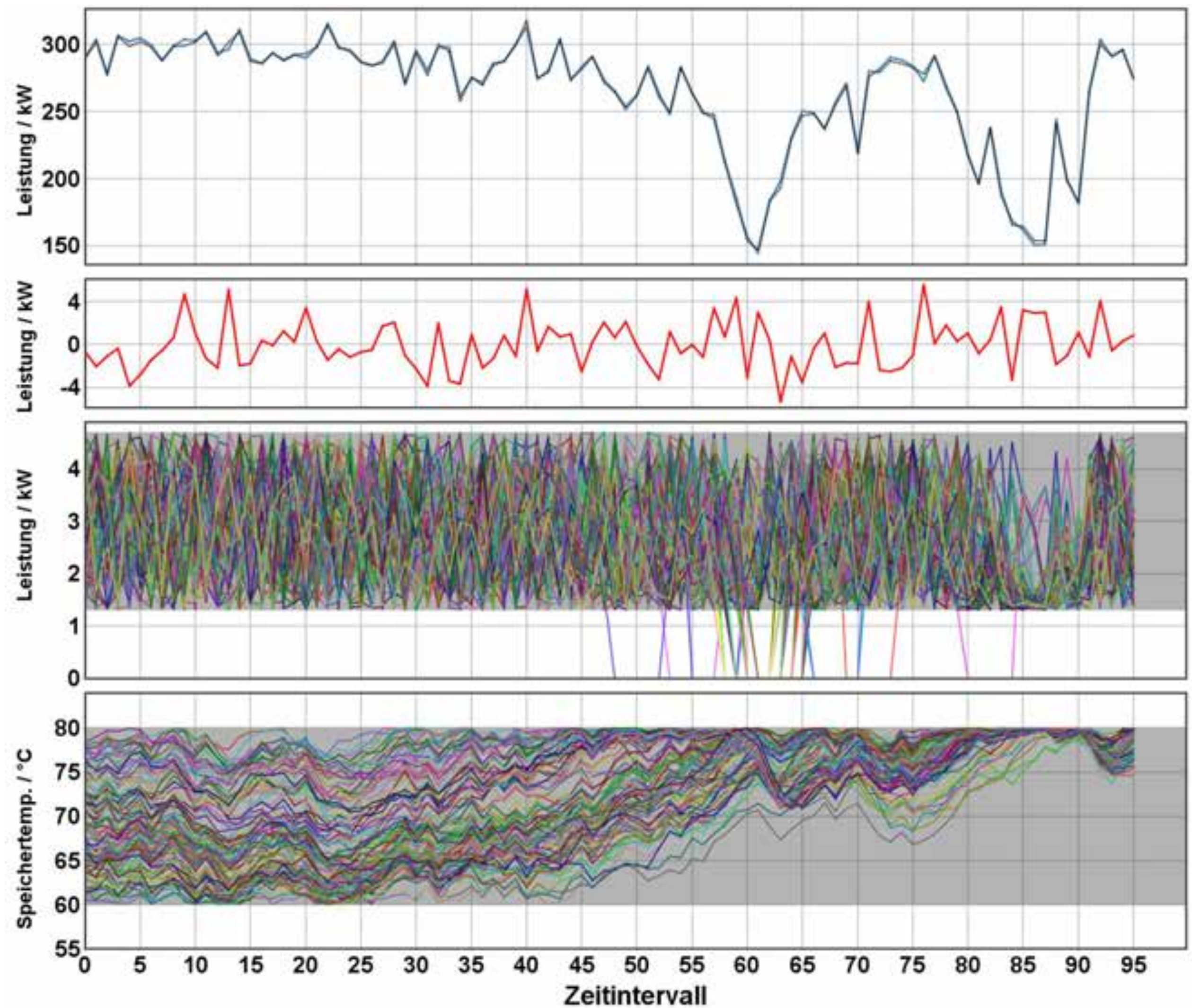
Exemplary result

Goal function and result of optimization

Residual error

Feasible schedules for 100 CHPs

Feasible temperature levels for the heat buffers of the CHPs



Reacting to events during product delivery

CONTINUOUS SCHEDULING

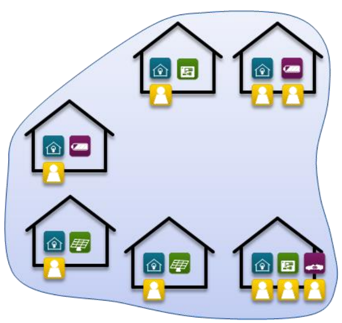
Continuous Energy Scheduling

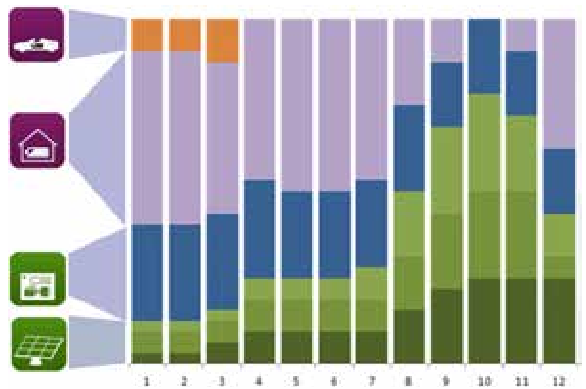
Market

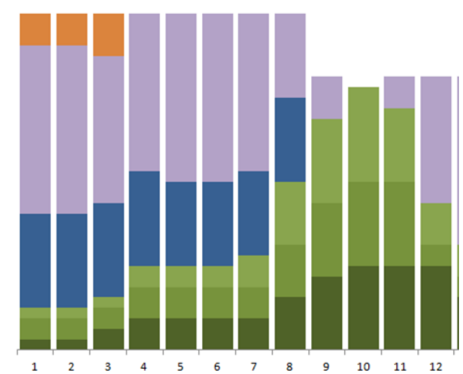
Product

?

Coalition







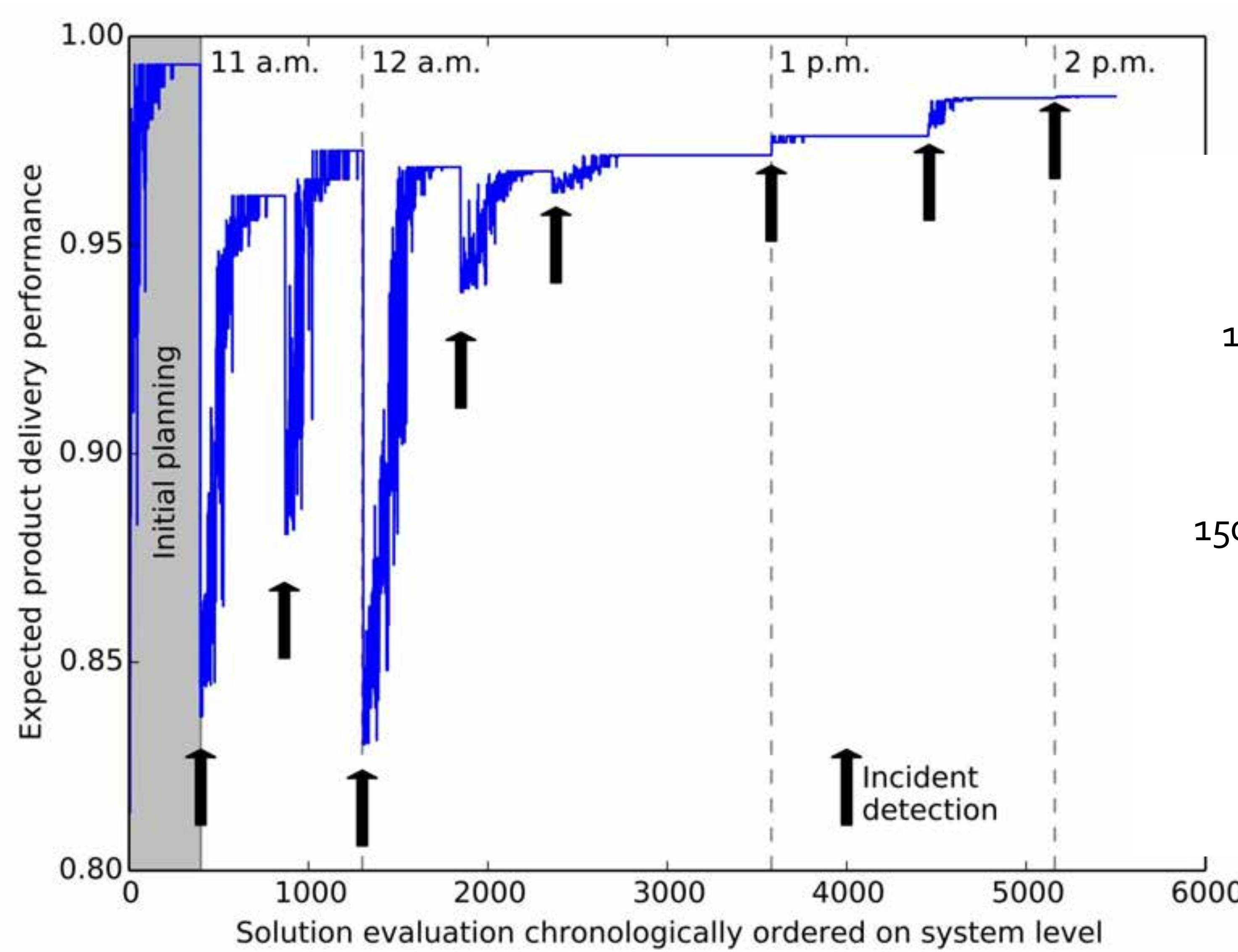
Unit

CHP schedule

Incident types

- Breakdown
- Prognosis fault
- System service

Results: Rescheduling



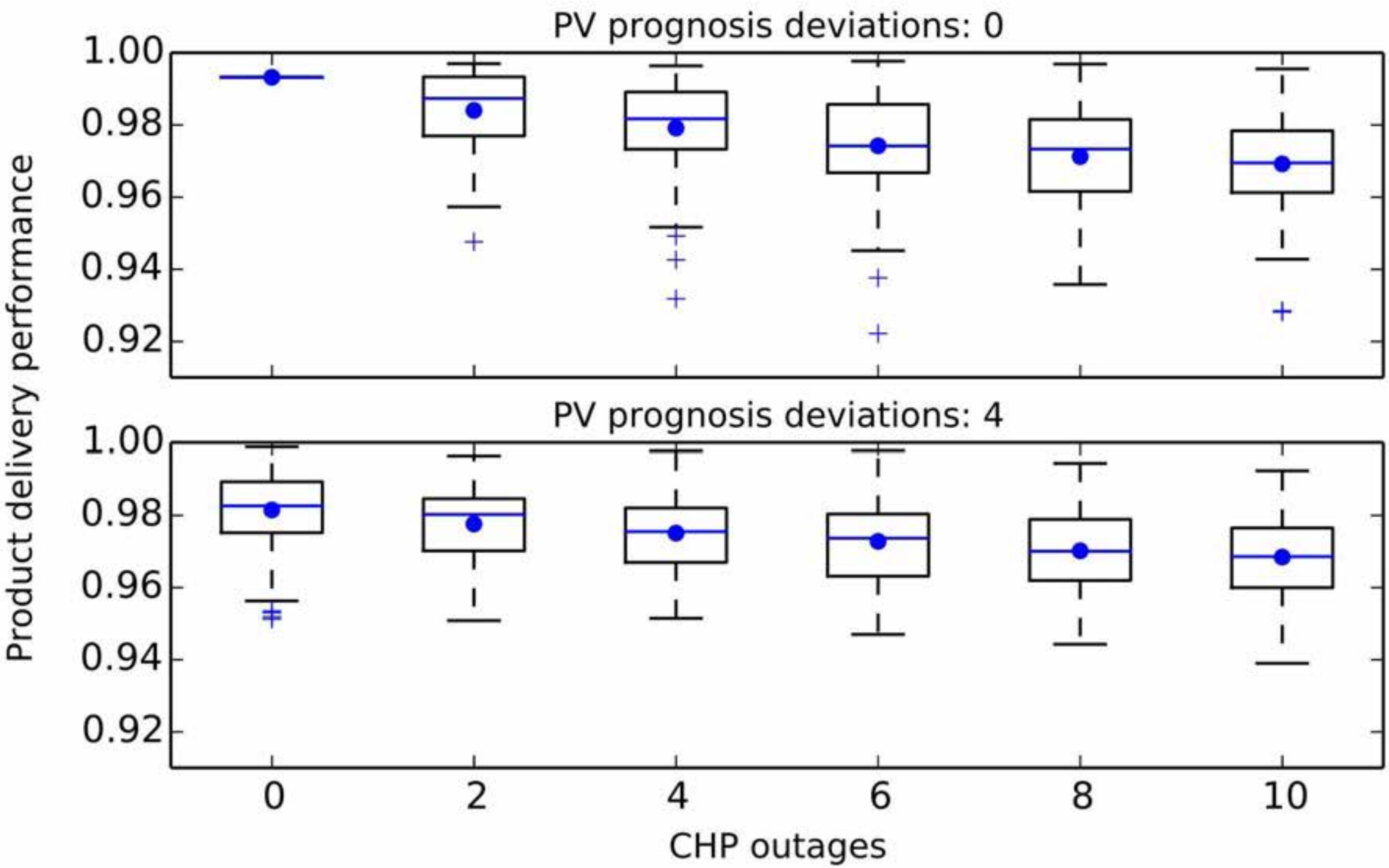
DVPP:
20 CHP(4,7 kW_{el})
20 PV (10 kW_p)
10 heat pumps(5kW)

Product:
4 h
150 kWh for each hour

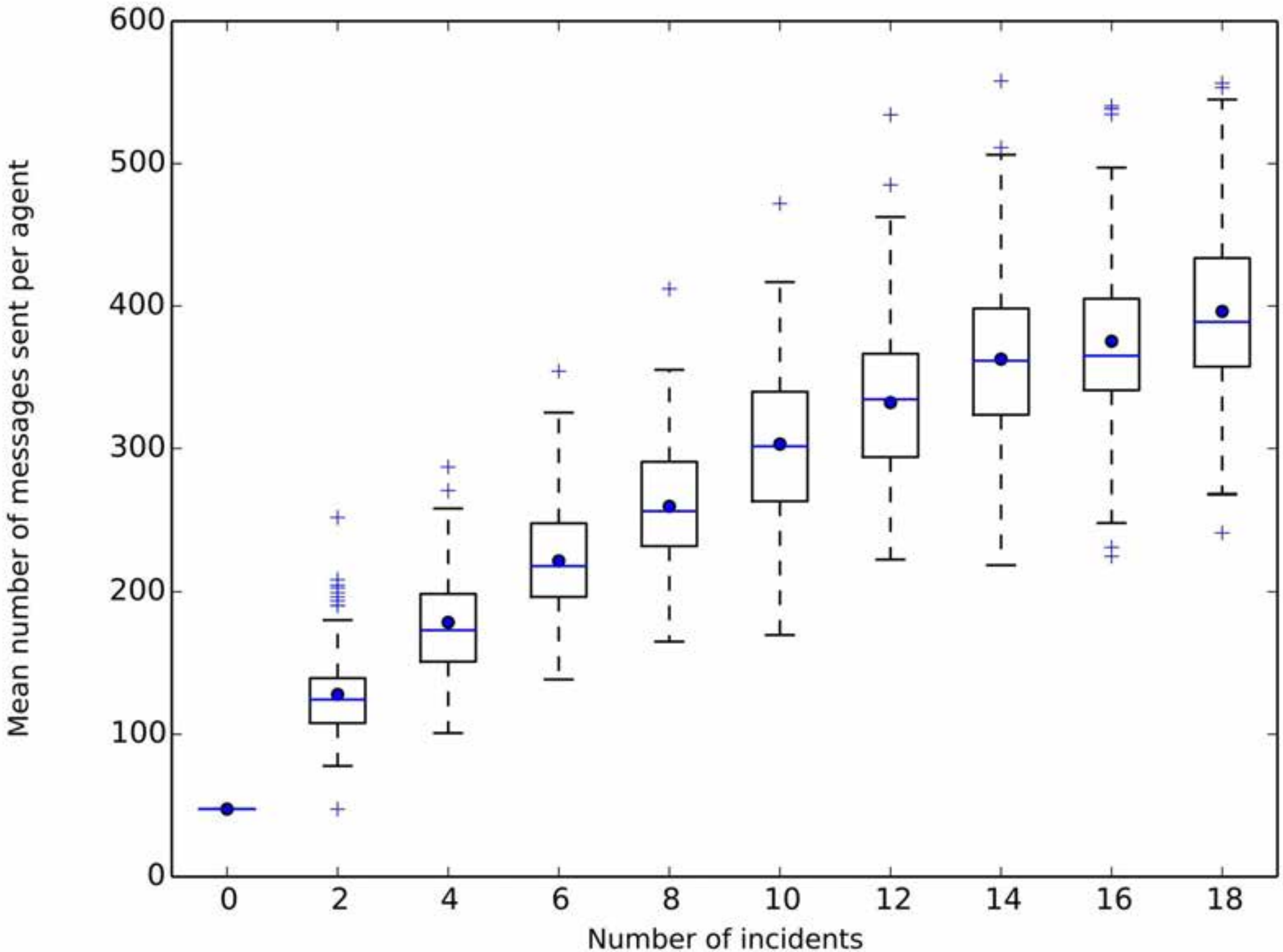
Incidents:
10 CHP shut-downs (30min)

Day in spring

Results: Rescheduling



Message overhead



Same scenario with 0 – 18 incidents

100 simulations runs for each incident amount

Random-based incident spreading

CONCLUSION

Main results

Completely self-organized virtual power plant

- coalition formation
- predictive scheduling
- reactive scheduling
- abstract representation of flexibilities

Smart grid algorithm engineering

Methodical challenges

- Engineering approach for self-organization methods
- `Well-behaviour´ of distributed self-organizing methods
  → controlled self-organization as in organic computing
- Security aspects in the engineering process
- Representative evaluation scenarios
- Computational effort for sensitivity analysis
- ...

CARL VON OSSIETZKY universität OLDENBURG

OFFIS INSTITUT FÜR INFORMATIK Energie

More information required?

http://smartnord.de/downloads/SmartNordFinalReport.pdf