An ACO/VNS Hybrid Approach for a Large-Scale Energy Management Problem

Challenge ROADEF/EURO 2010

Roman Steiner, Sandro Pirkwieser, Matthias Prandtstetter
Vienna University of Technology, Austria
Institute of Compter Graphics and Algorithms

June 13th 2010

overview

fitness function

fitness function

- without violations:

  objective function

- with violations:

$$f = v \cdot f_0 + \frac{q}{q_0} \cdot f_0$$

v number of violations

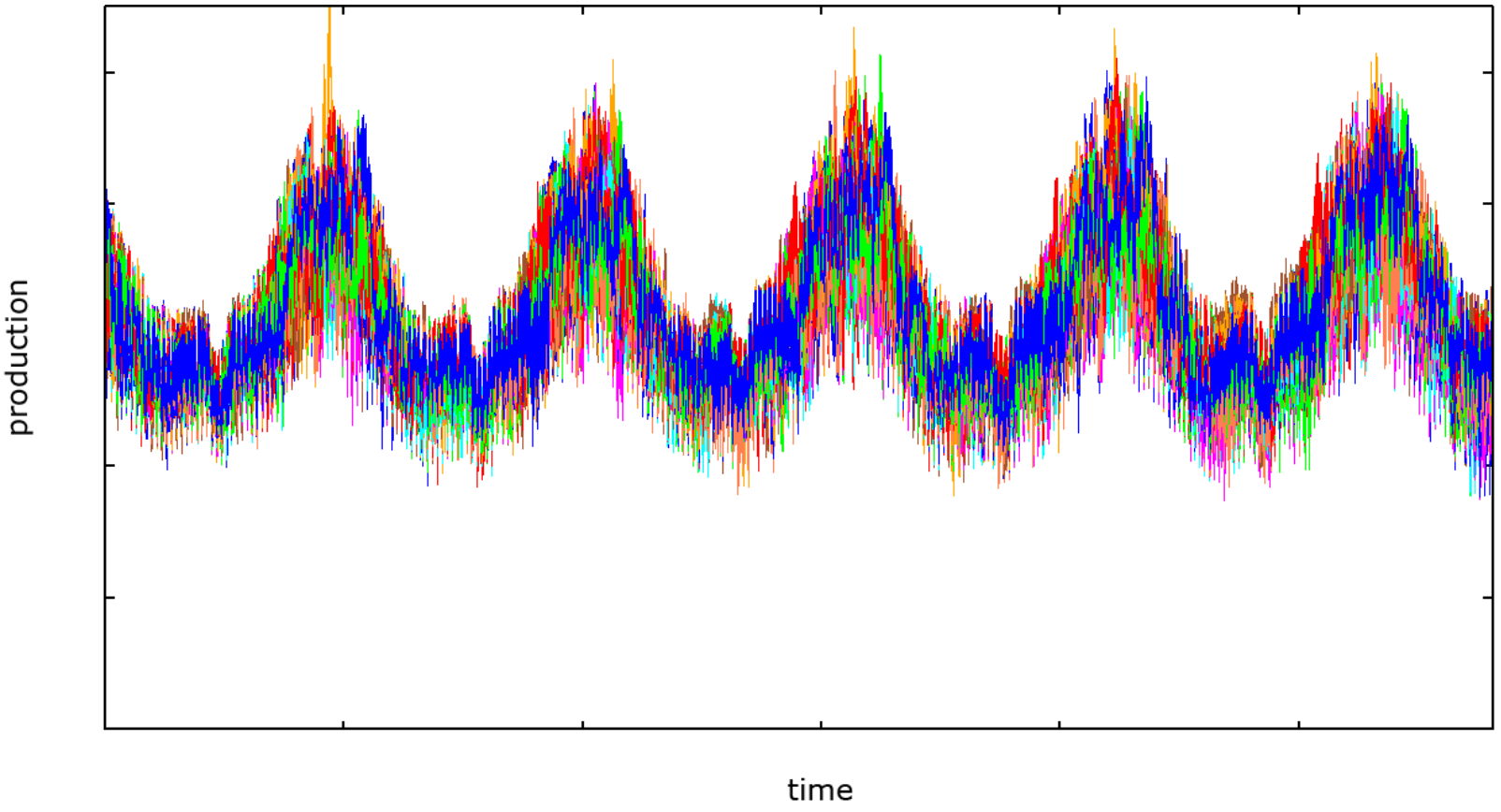
q quality measure

f_0 cost of trivial (infeasible) solution

q_0 quality measure of trivial solution

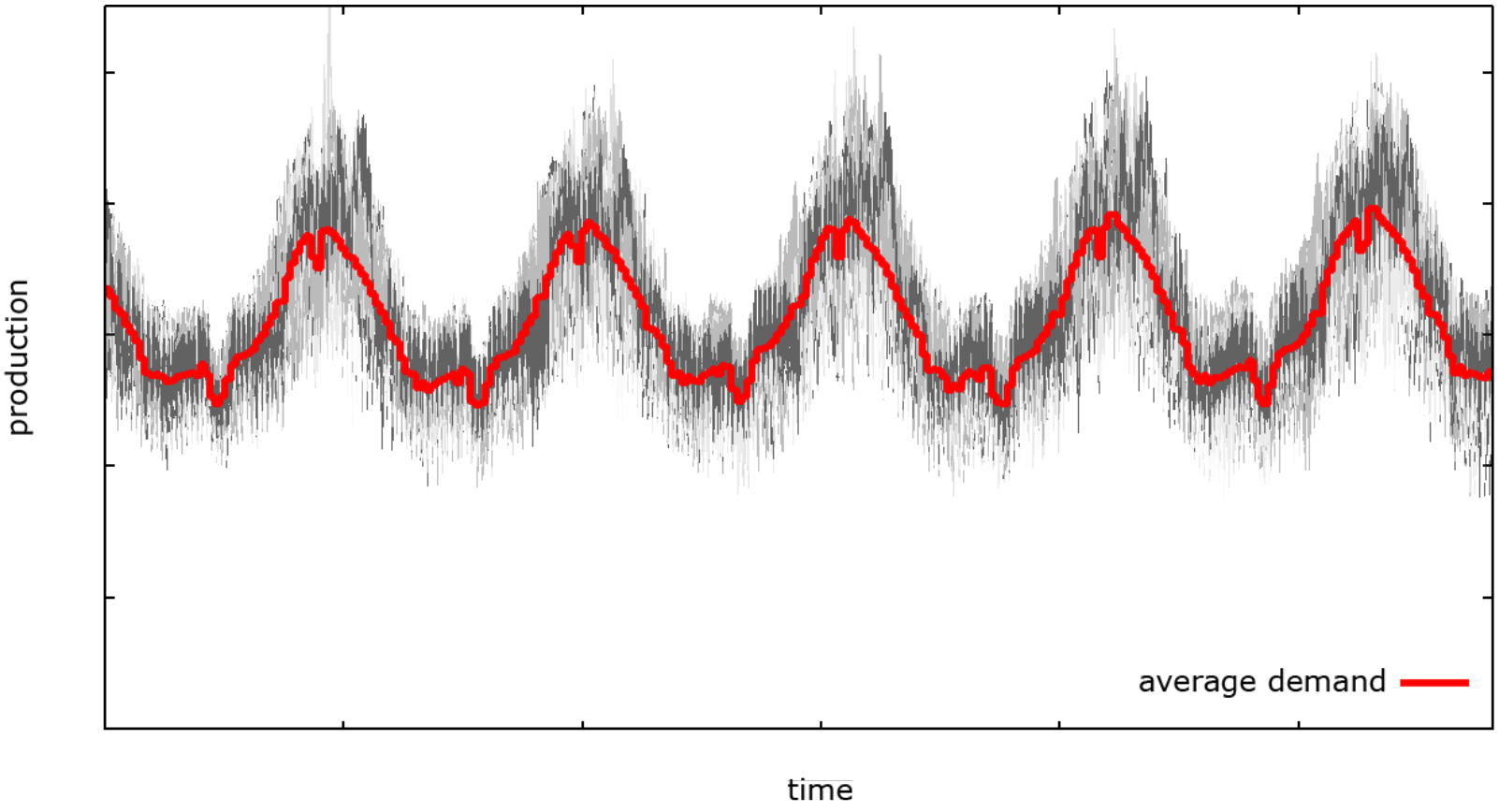
fitness function

quality measure



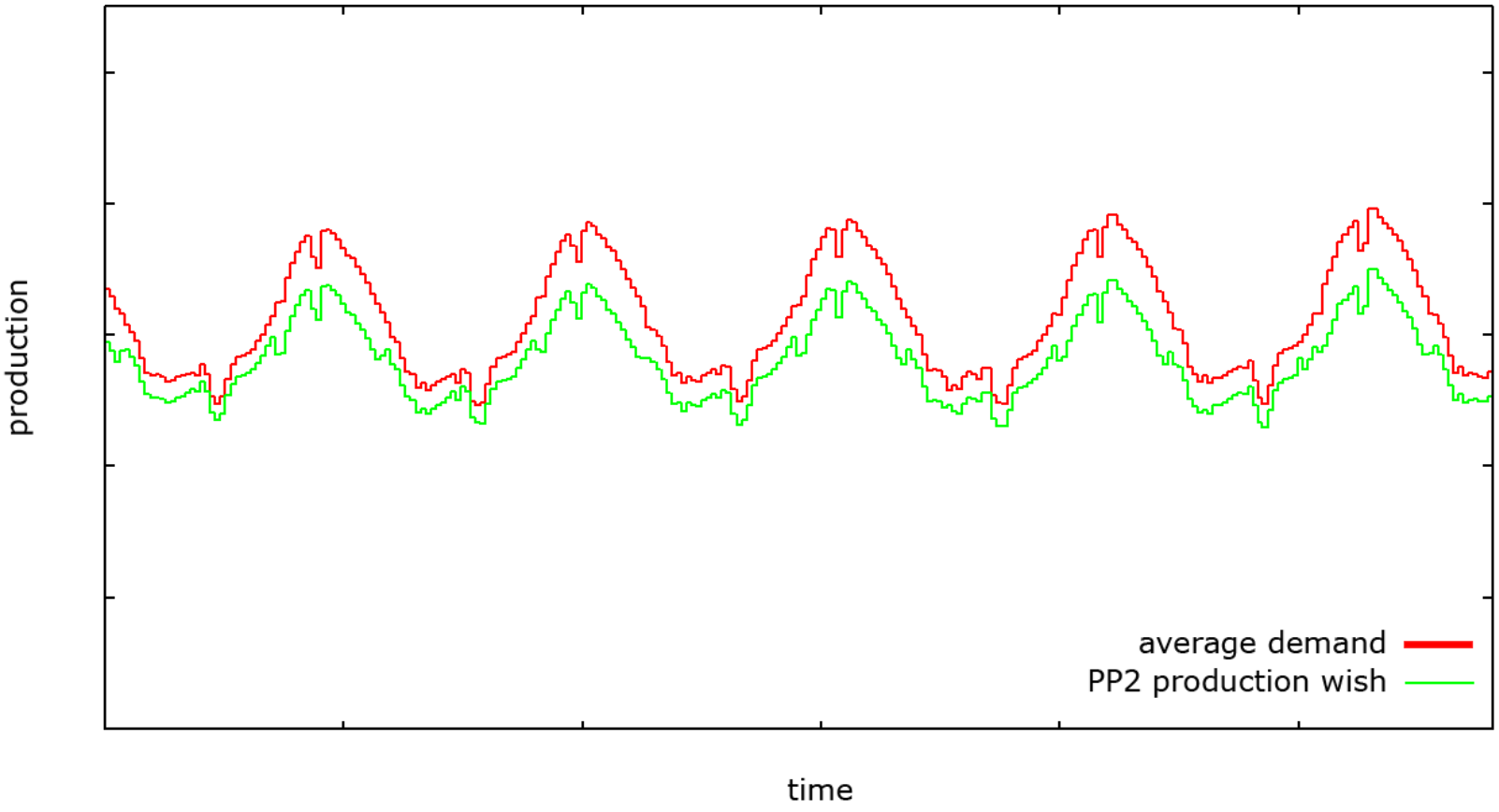
fitness function

quality measure



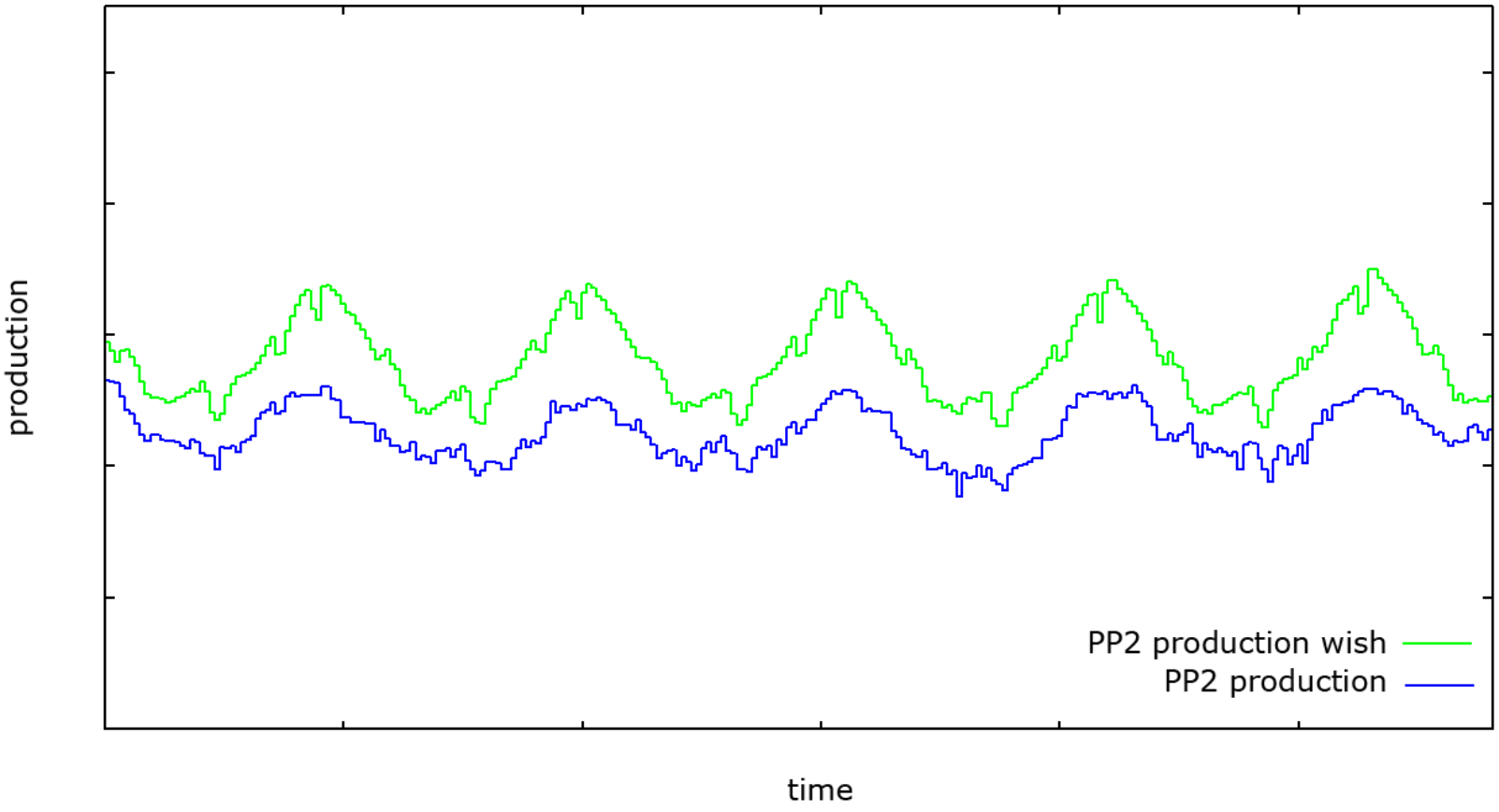
fitness function

quality measure



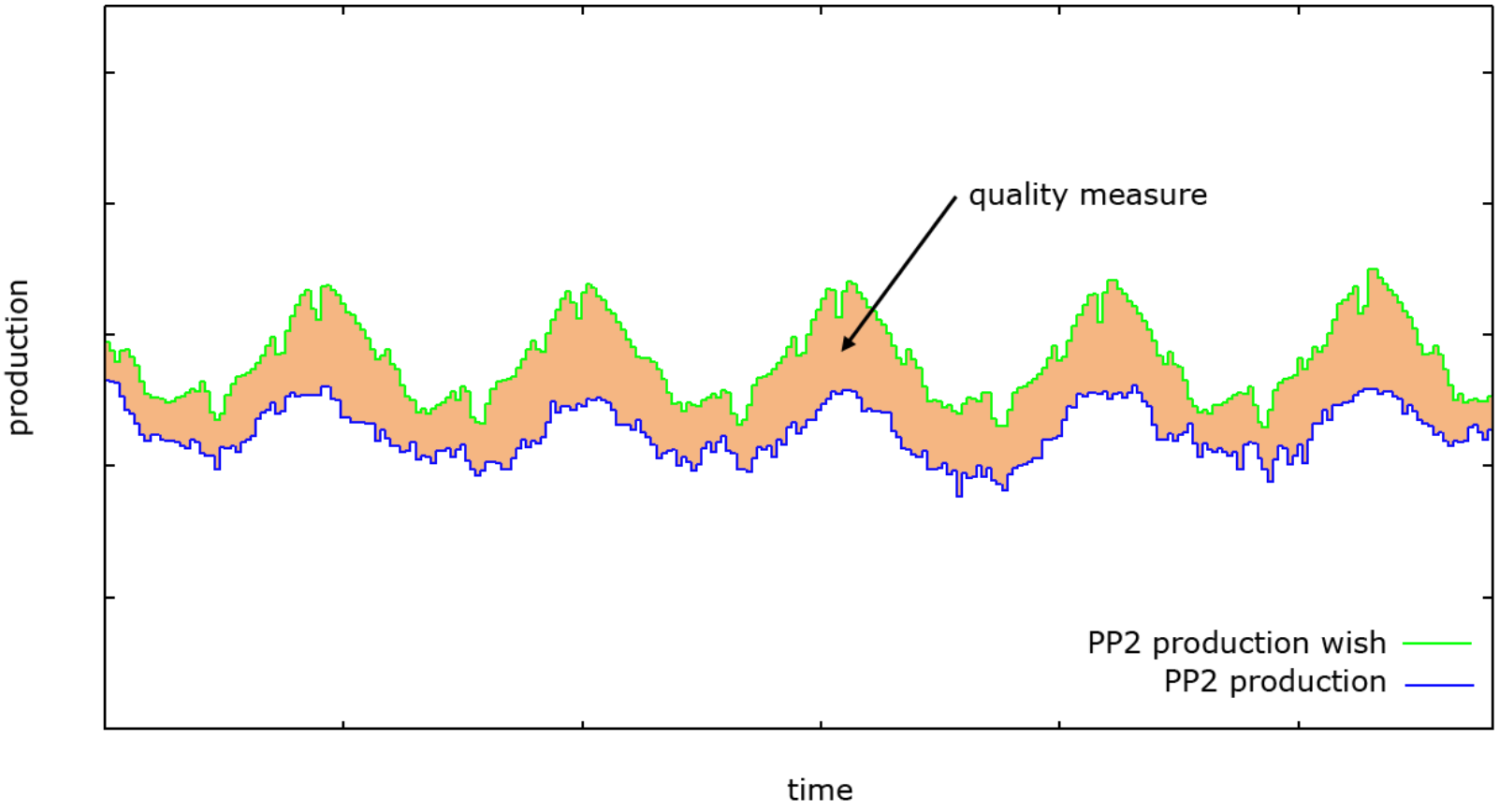
fitness function

quality measure



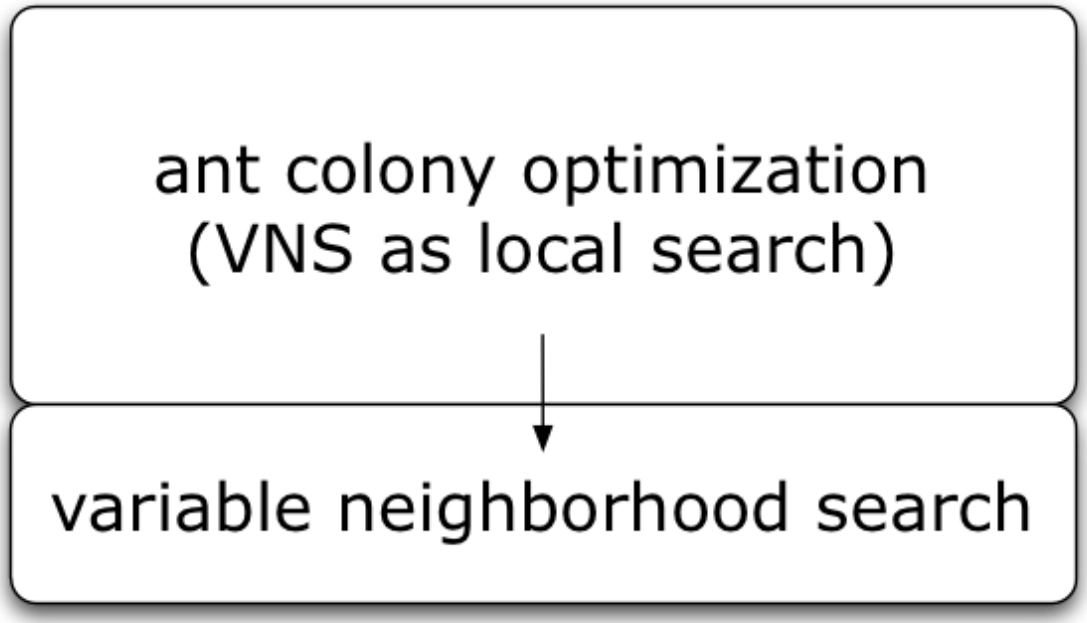
fitness function

quality measure



metaheuristic

the algorithm



ant colony optimization

- pheromone model: date selection probability for each outage
- apply VNS on best ant
- only best ant deposits pheromones
- no pheromones on outages with violations

ant colony optimization

solution generation

solution generation

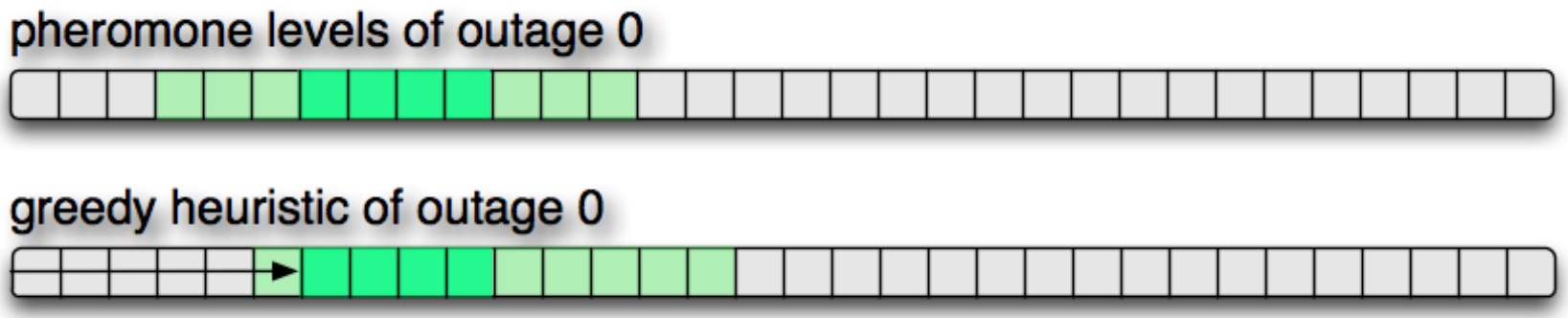
- outages are set in consecutive order
- selection based on combination of pheromone level and greedy heuristic

pheromone level: gives probability of dates for each outage

greedy heuristic: next outage when power plant empty

ant colony optimization

solution generation

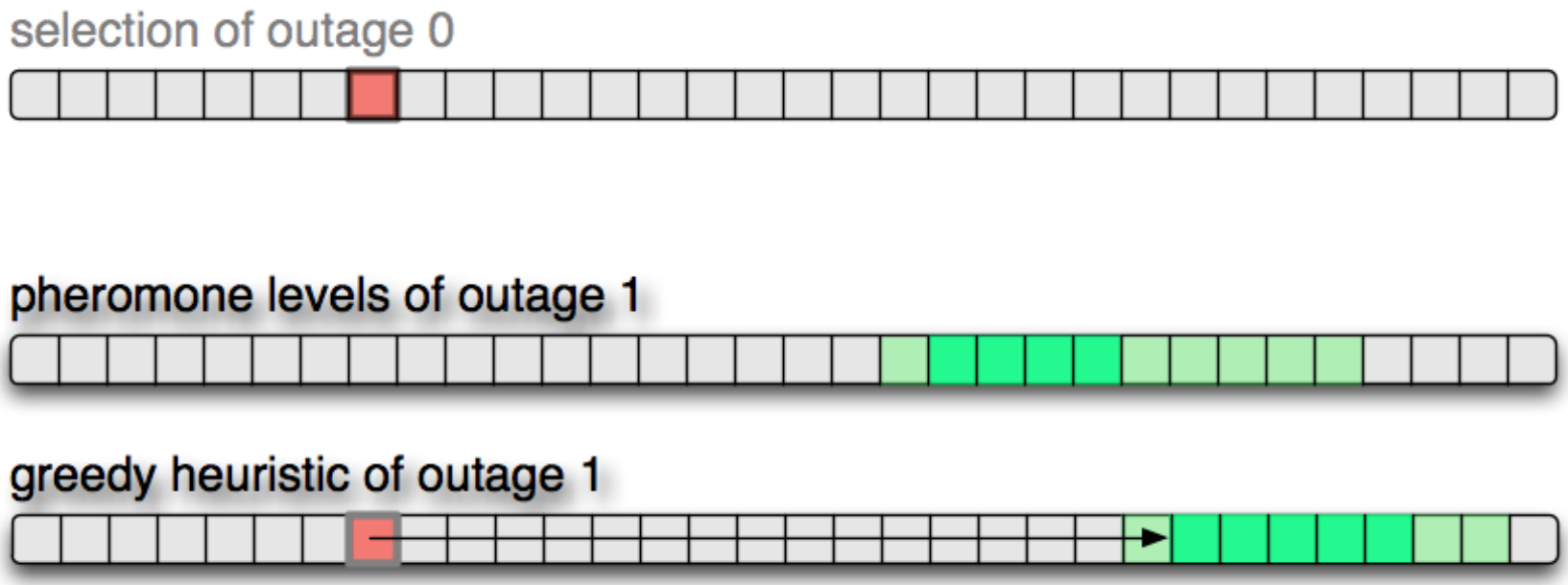


ant colony optimization

solution generation

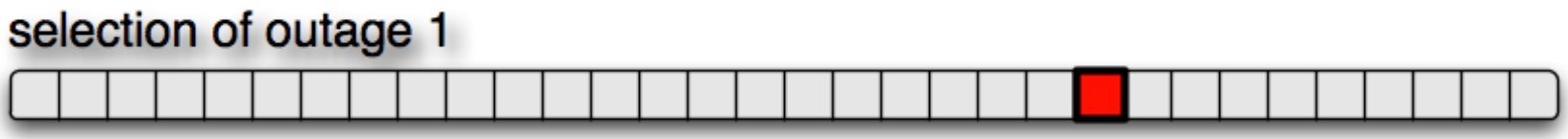
ant colony optimization

solution generation



ant colony optimization

solution generation



variable neighborhood search

- reduced VNS
- only shaking operators
- no local search
- different shaking operations in ACO and stand alone VNS

variable neighborhood search

shaking operations

used in the ACO

- **do outage** adds a new outage at the end
- **move outage** moves one outage to a different date
- **swap outages** swaps two outages of different power plants
- **no outage** removes the last outage
- **mix** select randomly one shaking operation

variable neighborhood search

shaking operations

used in the stand alone VNS

move outage moves one outage to a different date

spread outages distance between two consecutive outages is increased

smooth outages smoothes the average distance of consecutive outages

production levels

production levels

- deterministic algorithm
- basic idea:
  - use as much PP2 production as possible
  - reduce where demand is exceeded
  - fill up with cheapest PP1
- not optimal

production levels
algorithm

1. phase 1
   - primitive PP2 production levels
     (all PP2 run on full production as long as possible)
2. phase 2 (iterated)
   - reduce over production uniformly on all PP2
   - check constraint violations
3. phase 3
   - fill up with PP1 production
     (cheapest first)

production levels
algorithm

1. phase 1
   - primitive PP2 production levels
     (all PP2 run on full production as long as possible)
2. phase 2 (iterated)
   - reduce over production uniformly on all PP2
   - check constraint violations
3. phase 3
   - fill up with PP1 production
     (cheapest first)

still over production? $\longrightarrow$ mark as infeasible

production levels

results

	ACO_VNS	ACO	VNS
data 6	8.680.756	8.830.216	*350.582.969* (60%)
data 7	8.404.446	8.662.380	*285.198.769* (60%)
data 8	10.041.115	9.991.102	*963.477.563* (12%)
data 9	9.492.713	9.834.747	*549.329.564* (50%)
data 10	8.337.593	8.654.830	*183.872.131* (82%)

Table: Average results with factor 1e6.

thank you

thank you for your attention