UNIVERSITÉ CÔTE D'AZUR

# *Algorithmes Évolutionnaires (M2 MIAGE IA²)*

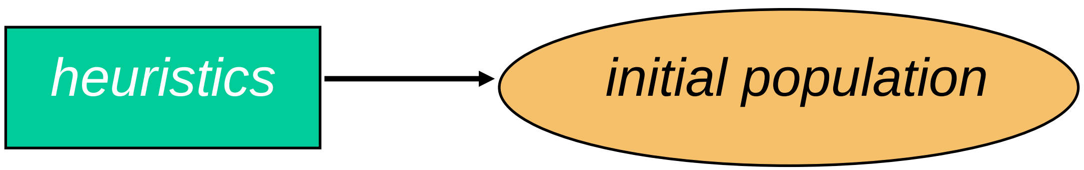
Andrea G. B. Tettamanzi
Laboratoire I3S – Équipe SPARKS
andrea.tettamanzi@univ-cotedazur.fr

univ-cotedazur.fr

# Séance 6
# Hybridation

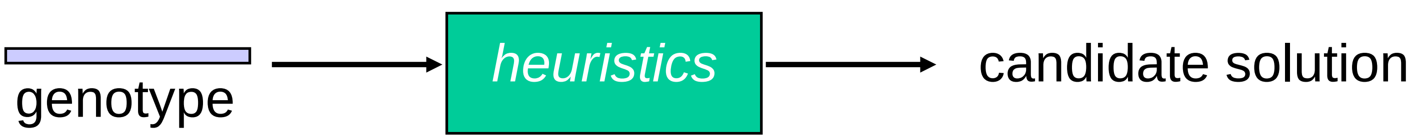
# Hybridization

1) Seed the population with solutions provided by some heuristics




2) Use local optimization algorithms as genetic operators
(Lamarckian mutation)

3) Encode parameters of a heuristics



# Seeding the Initial Population

- Given an available heuristics for the problem
- Two cases:
  - heuristics gives one solution
  - heuristics gives different solution when restarted

# Case 1: One Solution

- Insert solution in the initial population
- Fill the population with
  - random solutions (not a good idea)
  - exact copies of solution (not so good)
  - perturbations of solution (ok)
  - perturbations of solution and random solutions (ok)

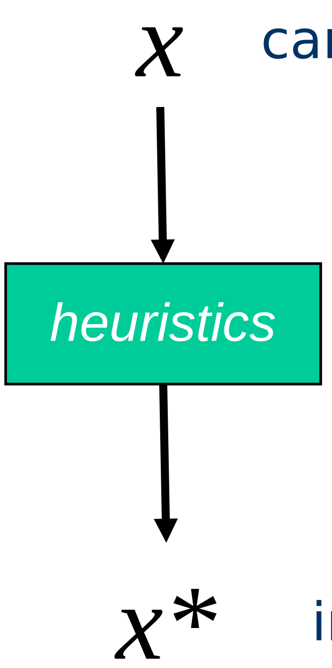
# Case 2: Many Solutions

- Run the heuristics as many times as needed to fill the initial population

# Discussion

- Win-win situation
- Results cannot be worse than the available heuristics
- Often they will be much better
- Use exploration capabilities of EAs

# Using Local Search Algorithms




candidate solution

local optimization algorithm

improved solution

# Using Local Search Algorithms

- Examples:
  - gradient descent methods
  - quasi-Newton methods (e.g., BFGS, Conjugate Gradient)
  - simulated annealing (with a limit on the number of moves)
  - etc.

# “Memetic” Algorithms

- Combination of local search and standard Eas
- “Meme” is a cultural gene
- Use an improvement operator, based on local search methods
- Use standard blind mutation as well

# Memetic Algorithms Pseudocode

```
MA() {
    Initialize(parents);
    while(gen <= MAX_GENERATIONS) {
        Evaluate(parents);
        LocalSearch(parents);
        mating_pool = Selection(parents);
        offspring = Crossover(mating_pool);
        Mutation(offspring);
        parents = Replacement(offspring, mating_pool);
        gen = gen + 1;
    }
}
```

# Sample Local Search Algorithm

```
moves = 0;
while(moves<MV) {
    old = ind;
    Mutate(ind); moves++;
    Δ = Fitness(old) - Fitness(ind);
    if(Δ > 0) // decline of solution quality
        if(Random(0, 1) > e^(-kΔ/T)) { // with probability 1 - e^(-kΔ/T)
            ind = old; // reject worse solution
            return;
        }
}
```

# Discussion

- Local search → new genetic operator
- No parallel in the natural genetic metaphor
- Attractive parallel in the cultural metaphor: “memes”
- Lamarckian mutation

# Heuristics as Decoders

- Basic assumption: parameterized heuristics

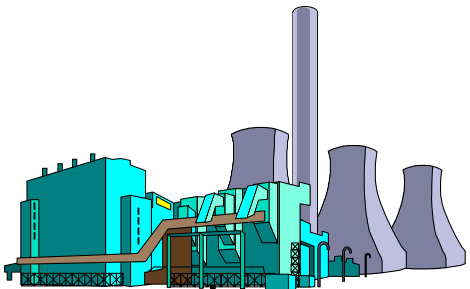
- Indirect representation:
  - encode heuristics parameters
  - the heuristics constructs the corresponding solution

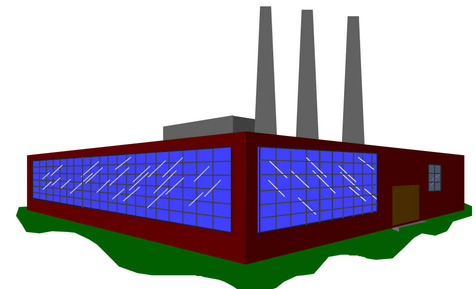
# Sample Application: Unit Commitment

- Multiobjective optimization problem: cost VS emission
- Many linear and non-linear constraints
- Traditionally approached with dynamic programming

- Hybrid evolutionary/knowledge-based approach
- A flexible decision support system for planners
- Solution time increases linearly with the problem size

# The Unit Commitment Problem

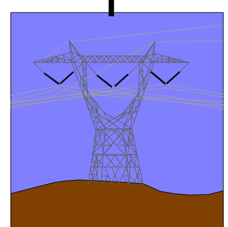




$$z_E = \sum_{i=1}^{n} E_i(P_i)$$

$$E_i(P_i) = \sum_{j=1}^{m} E_{ij}(P_i)$$

$$E_{ij}(P_i) = \alpha_{ij} + \beta_{ij} P_i + \gamma_{ij} P_i^2$$



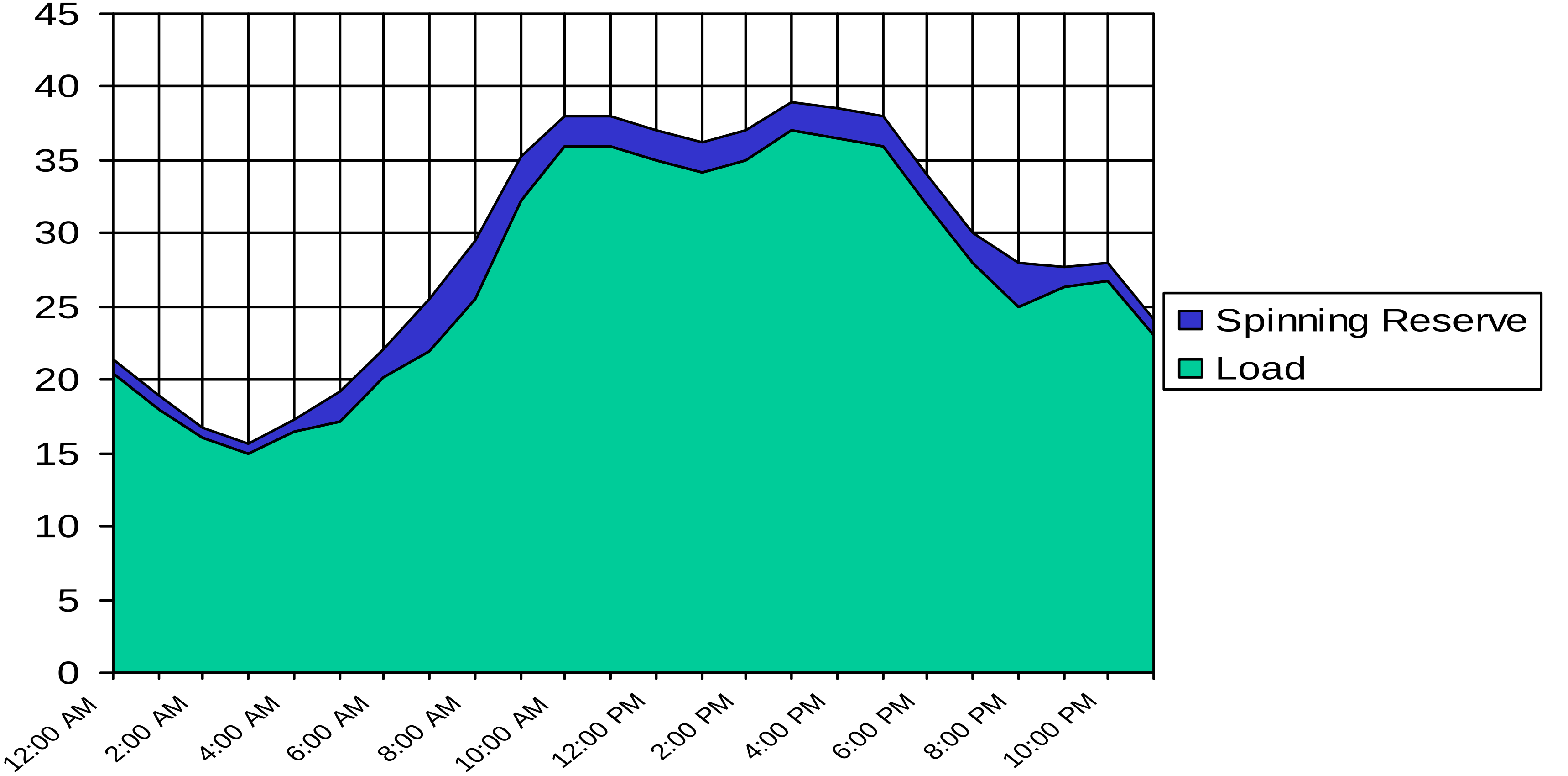
$$z_\$ = \sum_{i=1}^{n} \left( C_i(P_i) + SU_i + SD_i + HS_i \right)$$

$$C_i(P_i) = a_i + b_i P_i + c_i P_i^2$$

***Emissions***

***Cost***

# Predicted Load Curve

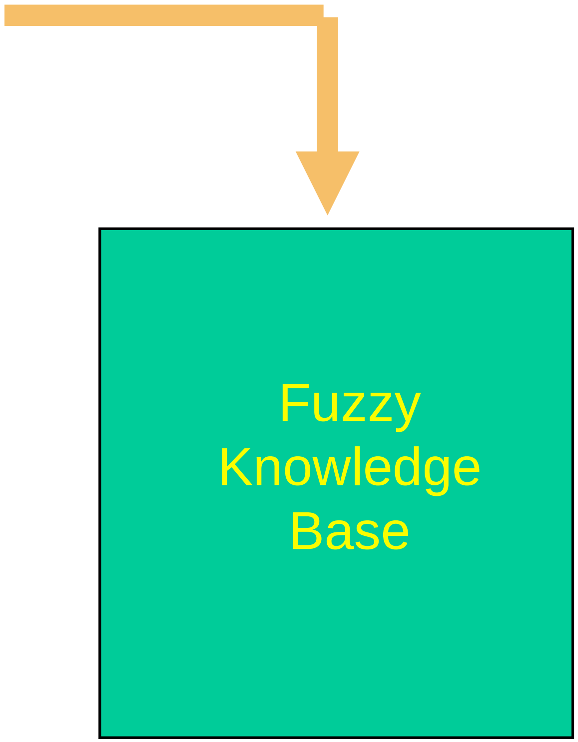


# Unit Commitment: Constraints

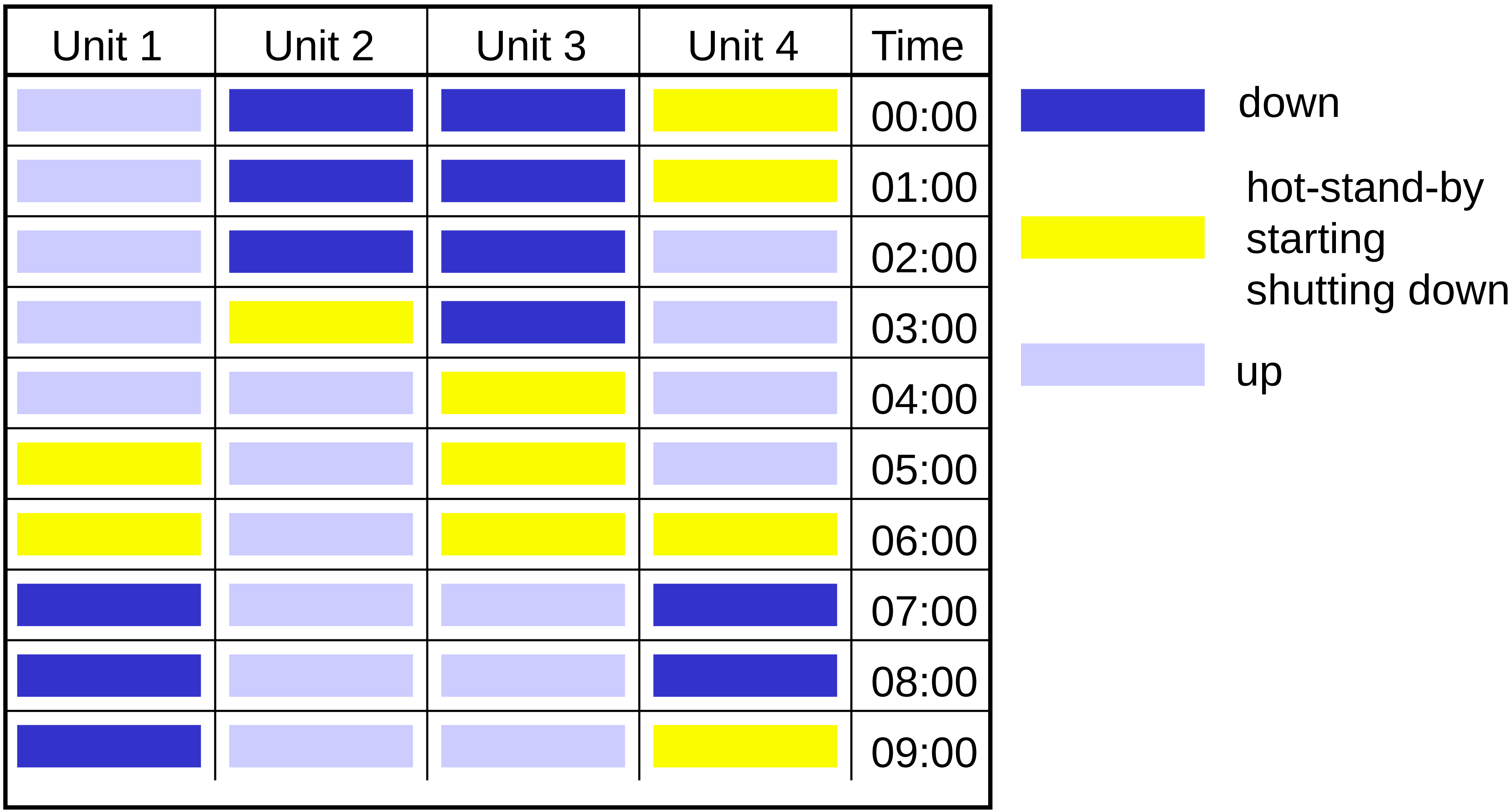
- Power balance requirement
- Spinning reserve requirement
- Unit maximum and minimum output limits
- Unit minimum up and down times
- Power rate limits
- Unit initial conditions
- Unit status restrictions
- Plant crew constraints
- ...

# Unit Commitment: Encoding

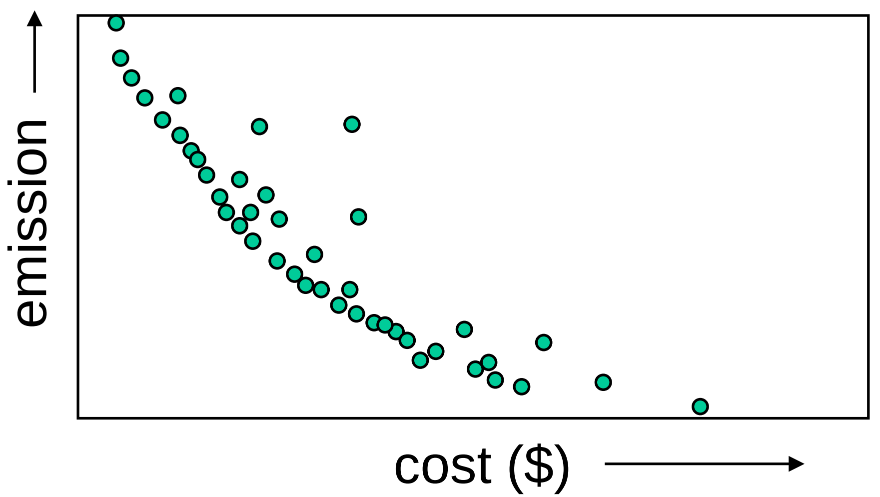
| Unit 1 | Unit 2 | Unit 3 | Unit 4 | Time |
|---|---|---|---|---|
| 1.0 | 0.8 | 0.2 | 0.15 | 00:00 |
| 0.9 | 1.0 | 0.2 | 1.0 | 01:00 |
| 0.0 | 1.0 | 0.8 | 0.2 | 02:00 |
| 0.0 | 0.5 | 1.0 | 0.8 | 03:00 |
| 1.0 | 0.65 | 0.8 | 1.0 | 04:00 |
| 0.8 | 0.8 | 0.25 | 1.0 | 05:00 |
| 1.0 | 0.4 | 0.2 | 1.0 | 06:00 |
| 0.0 | 0.0 | 1.0 | 0.75 | 07:00 |
| 0.5 | 1.0 | 1.0 | 0.8 | 08:00 |
| 1.0 | 0.5 | 0.0 | 0.0 | 09:00 |



# Unit Commitment: Solution



# Unit Commitment: Selection




**competitive selection:**

| $507,762 | ⟷ | $516,511 |
|---|---|---|
| 213,489 £ | | 60,080 £ |

# Unit Commitment References


- D. Srinivasan, A. Tettamanzi. "An Integrated Framework for Devising Optimum Generation Schedules". In *Proceedings of the 1995 IEEE International Conference on Evolutionary Computing (ICEC '95)*, vol. 1, pp. 1-4.
- D. Srinivasan, A. Tettamanzi. *A Heuristic-Guided Evolutionary Approach to Multiobjective Generation Scheduling.* IEE Proceedings Part C - Generation, Transmission, and Distribution, 143(6):553-559, November 1996.
- D. Srinivasan, A. Tettamanzi. *An Evolutionary Algorithm for Evaluation of Emission Compliance Options in View of the Clean Air Act Amendments.* IEEE Transactions on Power Systems, 12(1):336-341, February 1997.