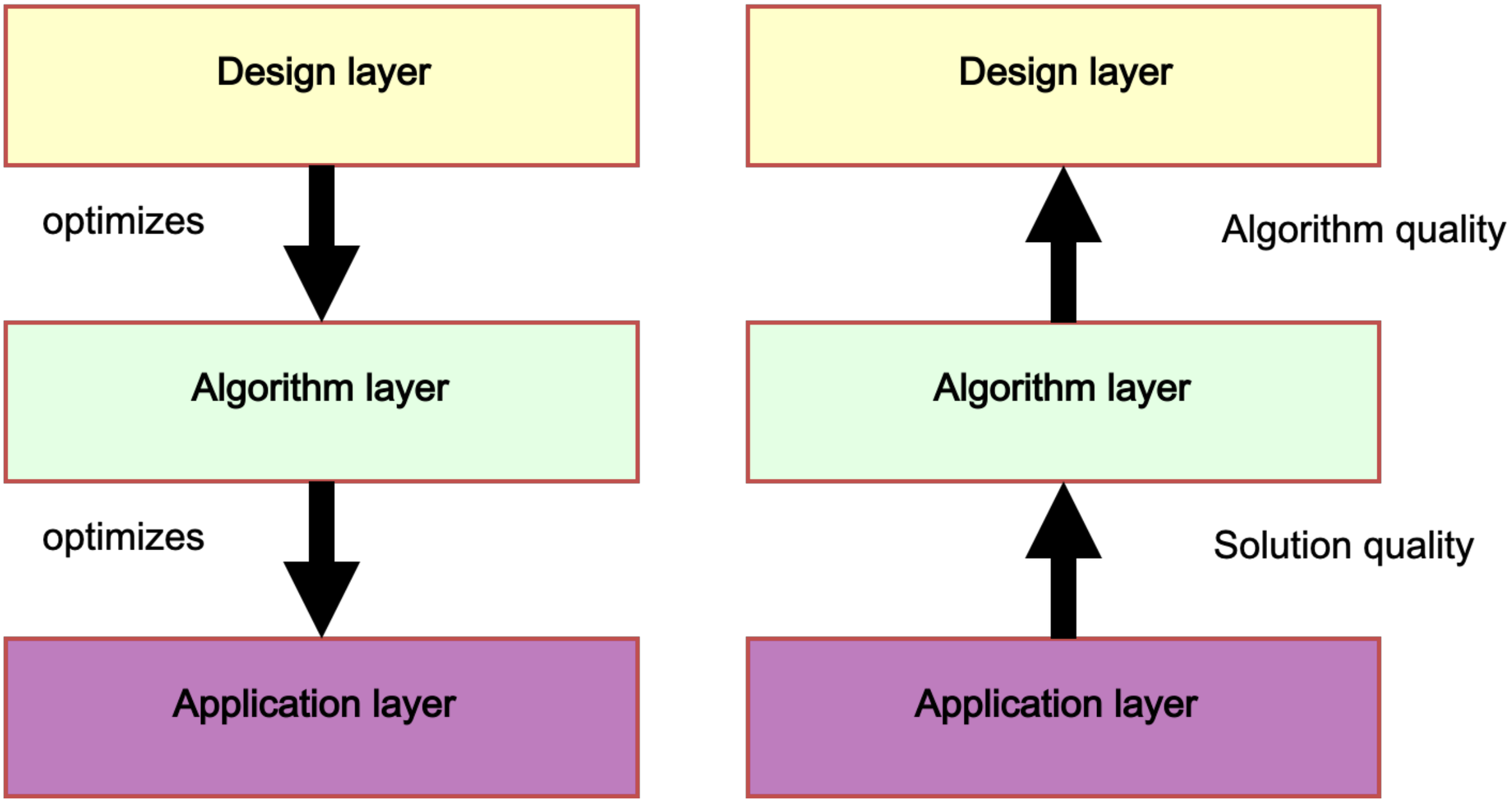
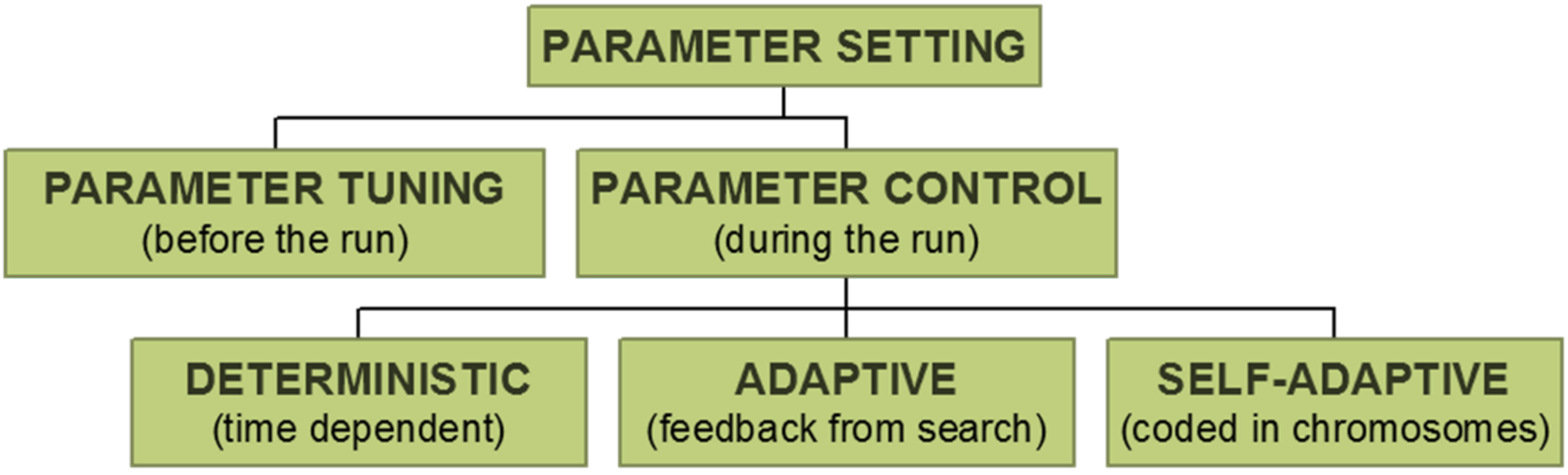
12: Parameter Tuning

- Parameter tuning vs. parameter control
- Symbolic vs. numeric parameters
- EA vs. EA instance
- Tuning by generate-and-test
- Tuning methods
- Textbook Chapter 7

Control and information flows of EA design



Taxonomy


PARAMETER SETTING
PARAMETER TUNING
(before the run)
PARAMETER CONTROL
(during the run)
DETERMINISTIC
(time dependent)
ADAPTIVE
(feedback from search)
SELF-ADAPTIVE
(coded in chromosomes)

Parameter tuning

- Test and compare different values before the “real” run
- Challenges
  - user mistakes in settings can be sources of errors or suboptimal performance
  - costs time
  - parameters interact: exhaustive search is not feasible; can’t test one at a time
  - requirements may change during a run

Parameter control

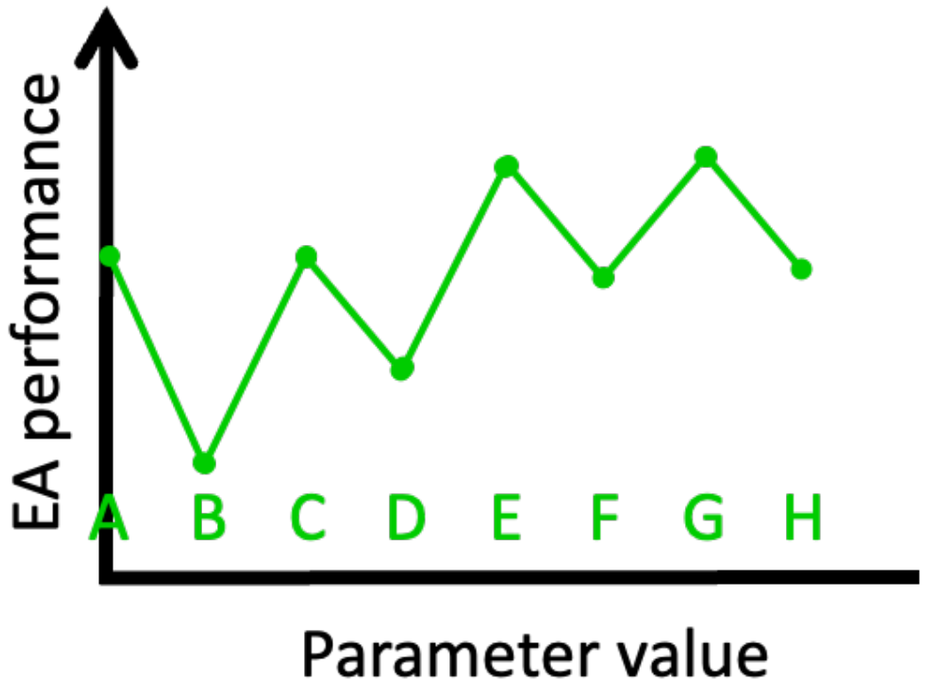
- Set values on-line, during a run
- Challenges
  - hard to optimize parameters with a time-varying schedule
  - hard to optimize user-defined feedback mechanism
  - hard to select for parameter optimization

Historical account

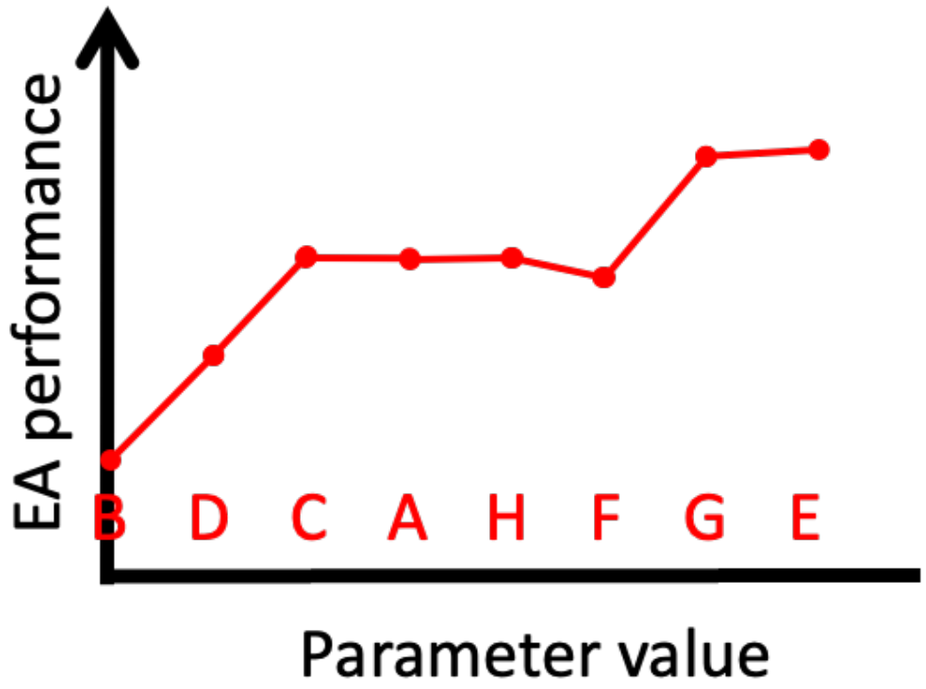
- Not much work published on parameter tuning
- Tuning efforts are often not reported
- No systematic way to tune parameters
  - based on conventions
  - ad hoc choices
  - limited experimentation with few values

Symbolic parameters

- E.g., mutation operator, xover operator, selection method
- Finite domain
- No sensible distance metric -> non-searchable, must be sampled



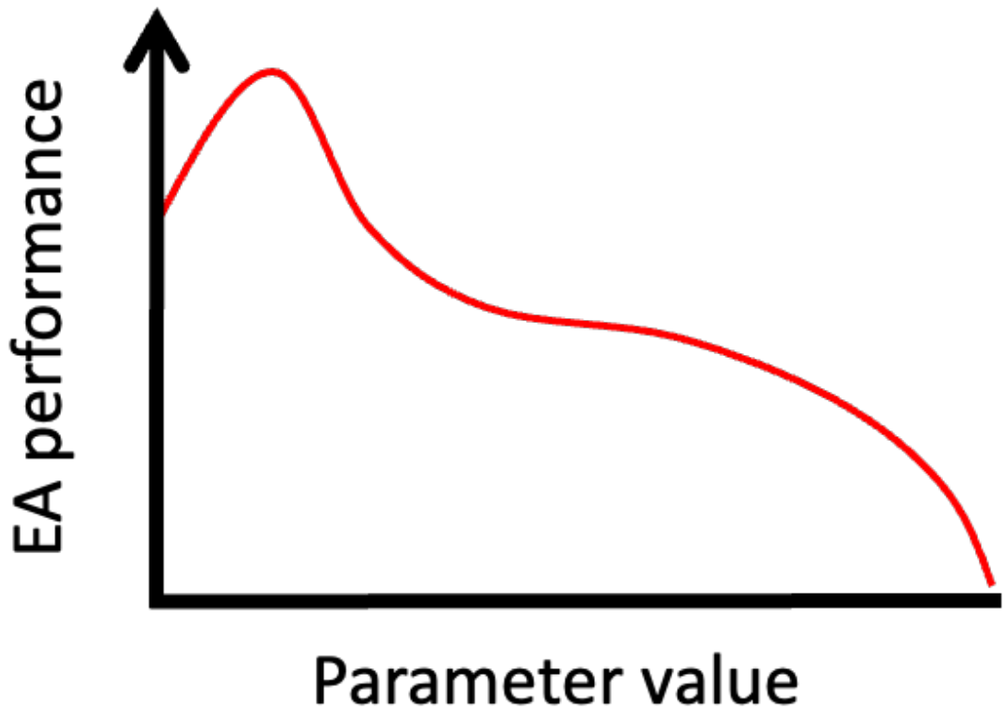
Non-searchable ordering



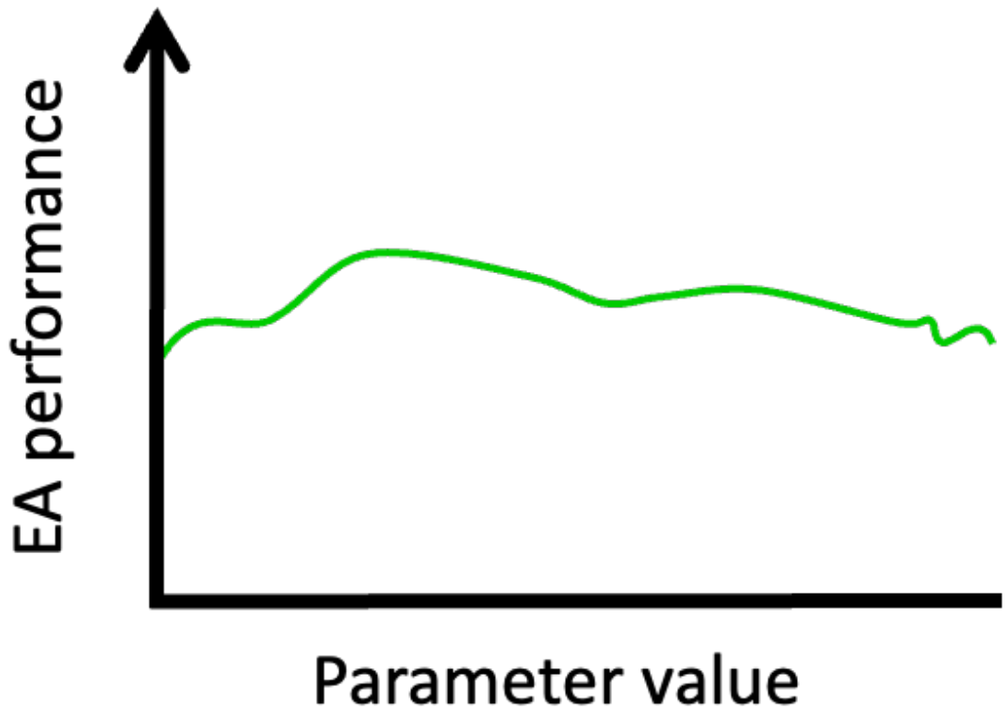
Searchable ordering

Numeric parameters

- E.g., population size, mutation rate, xover rate, tournament size
- Domain is subset of R or N
- Sensible distance metric -> searchable



Relevant parameter



Irrelevant parameter

	ALG-1	ALG-2	ALG-3	ALG-4
SYMBOLIC PARAMETERS				
Representation	Bit-string	Bit-string	Real-valued	Real-valued
Overlapping pops	N	Y	Y	Y
Survivor selection	–	Tournament	Replace worst	Replace worst
Parent selection	Roulette wheel	Uniform determ	Tournament	Tournament
Mutation	Bit-flip	Bit-flip	N(0,σ)	N(0,σ)
Recombination	Uniform xover	Uniform xover	Discrete recomb	Discrete recomb
NUMERIC PARAMETERS				
Population size	100	500	100	300
Tournament size	–	2	3	30
Mutation rate	0.01	0.1	–	–
Mutation stepsize	–	–	0.01	0.05
Crossover rate	0.8	0.7	1	0.8

EA vs. EA instance

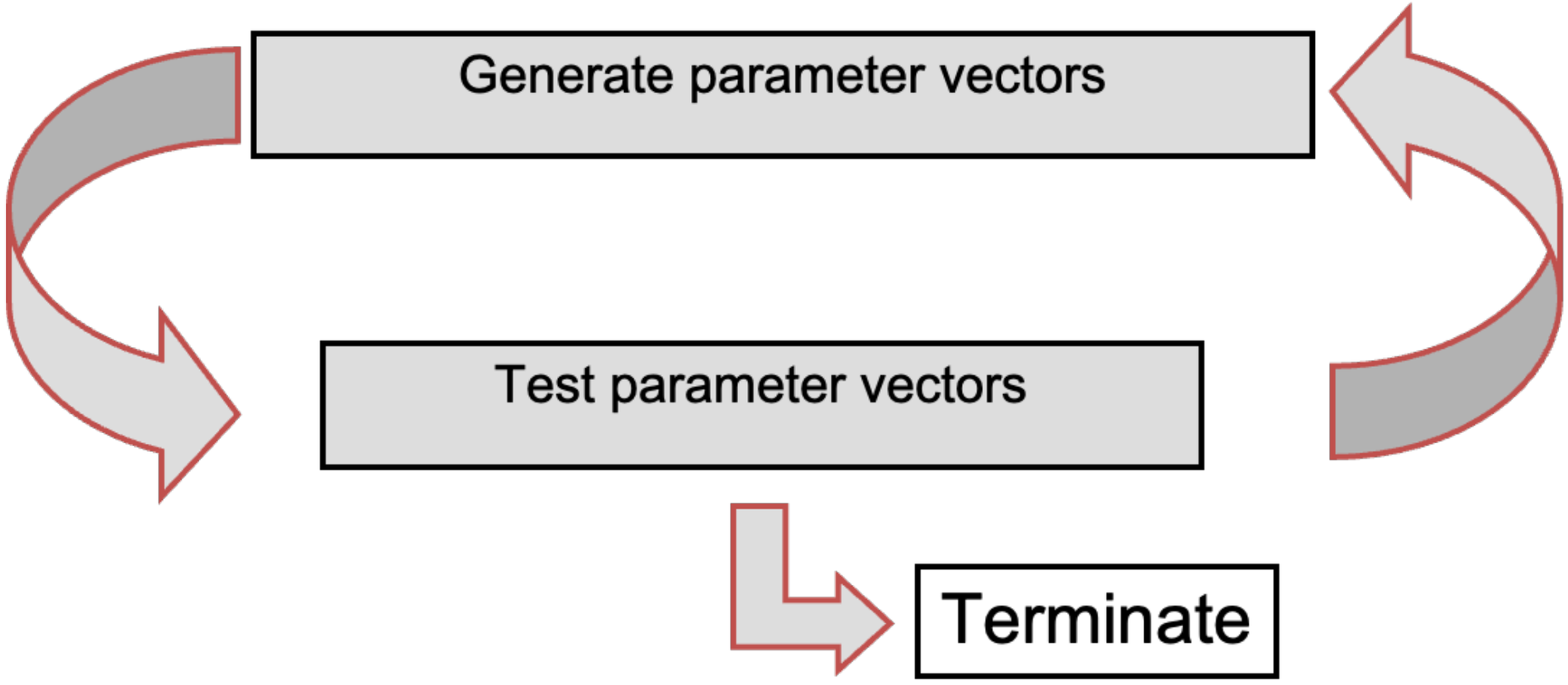
- Option 1:
    - There is only on EA, the generic EA
    - The table contains 1 EA and 4 EA instances
- Option 2: (commonly adopted)
    - An EA is a particular configuration of the symbolic parameters
    - The table contains 3 EAs with 2 instances for one of them
- Option 3:
    - An EA is a particular configuration of all parameters
    - EA = EA instance
    - The table contains 4 EAs

Why parameter tuning?

- Configure an EA by choosing parameter values that optimize its performance
- Analyze an EA by studying how its performance depends on its parameter values and/or the problems it is applied to
- Depends on
    - the problem to be solved
    - the EA used
    - the utility function (how we measure the algorithm quality)

Tuning by generate-and-test

- EA parameter tuning is a search problem itself



Meta-EA

- Run an EA to optimize the parameters of another EA
- Parameters can have a hierarchical, nested structure
    - e.g., a symbolic parameter can introduce a numeric parameter
- Utility landscape
    - abstract landscape where the locations are the parameter vectors of an EA
    - height reflects utility of a parameter vector
    - utility values are always stochastic (fitness values are typically deterministic)
    - defined in a statistical sense

Meta-EA: testing parameter vectors

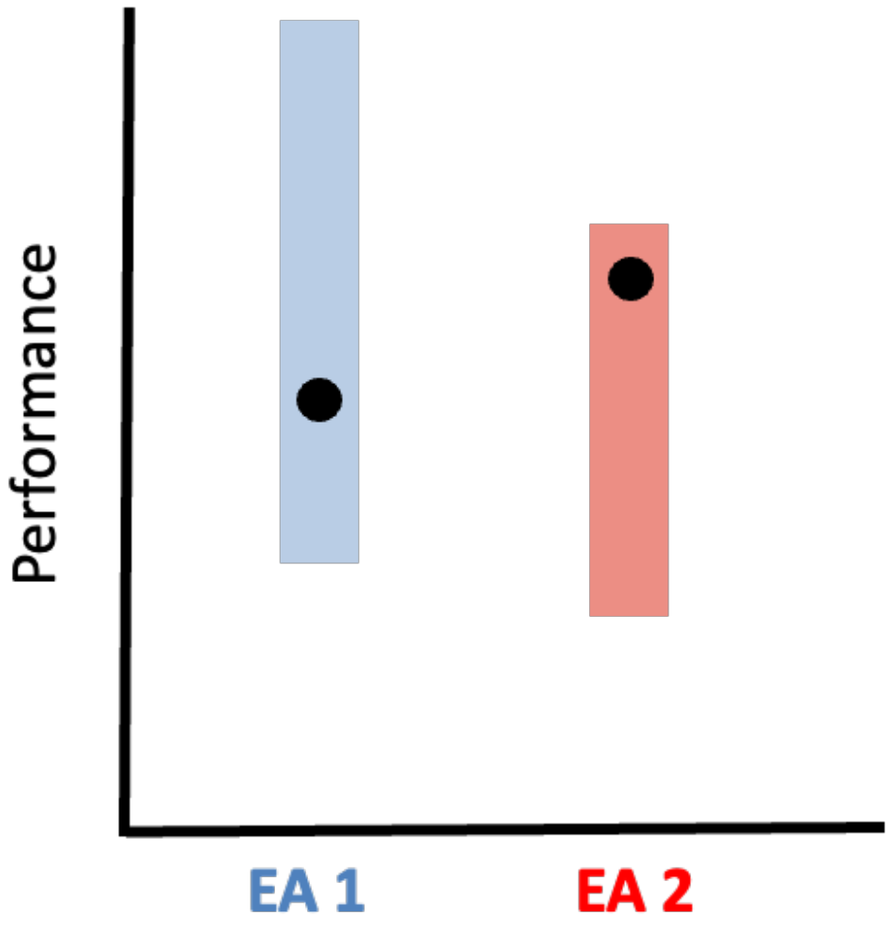
- Run EA with these parameters on the given problem(s)
- Record EA performance in that run using
  - solution quality: best fitness at termination
  - speed: time used to find required solution quality
- Multiple runs needed because EAs are stochastic
  - average performance of solution quality or speed
  - success rate
  - robustness
  - …

Tuning methods

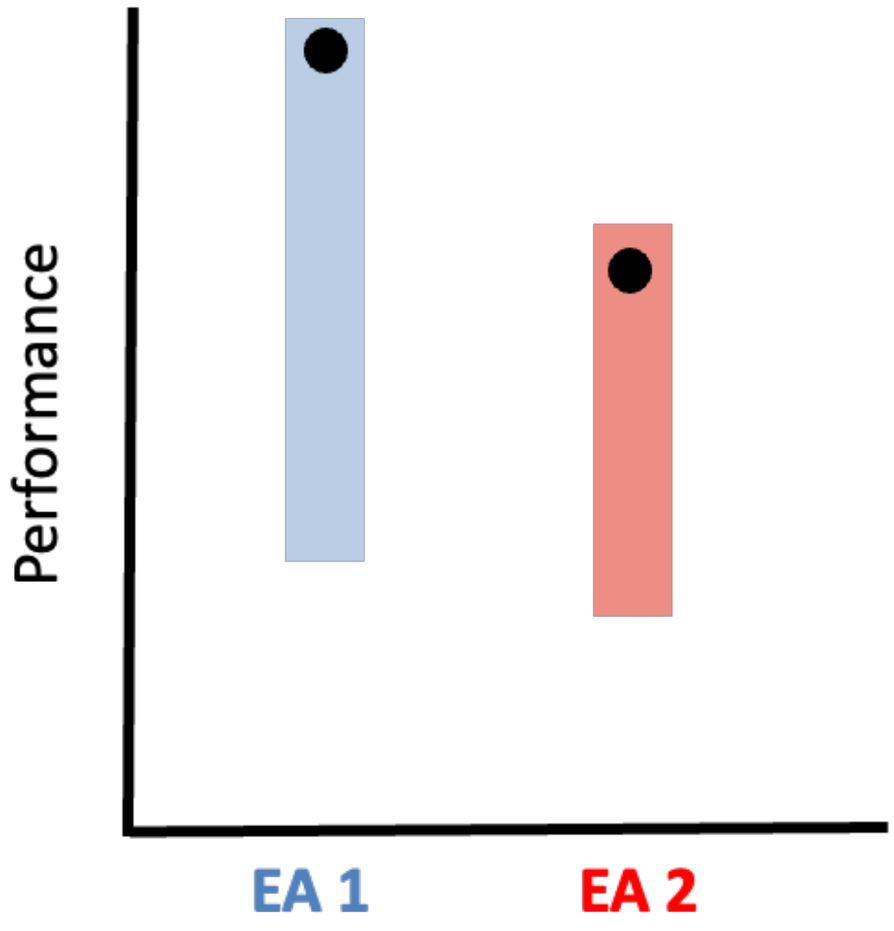
- Single-staged vs. multi-staged
  - both perform a number of tests for a reliable estimate of utilities
  - single-staged perform the same number of tests for each parameter vector
  - multi-staged select promising vectors for further tests, e.g., racing
- Model-free vs. model-based
  - optimize the utility landscape by finding parameter vectors that maximize performance
  - learn a model that estimates the performance of an EA for any given parameter vector

Eiben and Smith: Evolutionary algorithm parameters and methods to tune them, 2012

Tuning is important



EA as is (accidental parameters)



EA as it can be (“optimal” parameters)