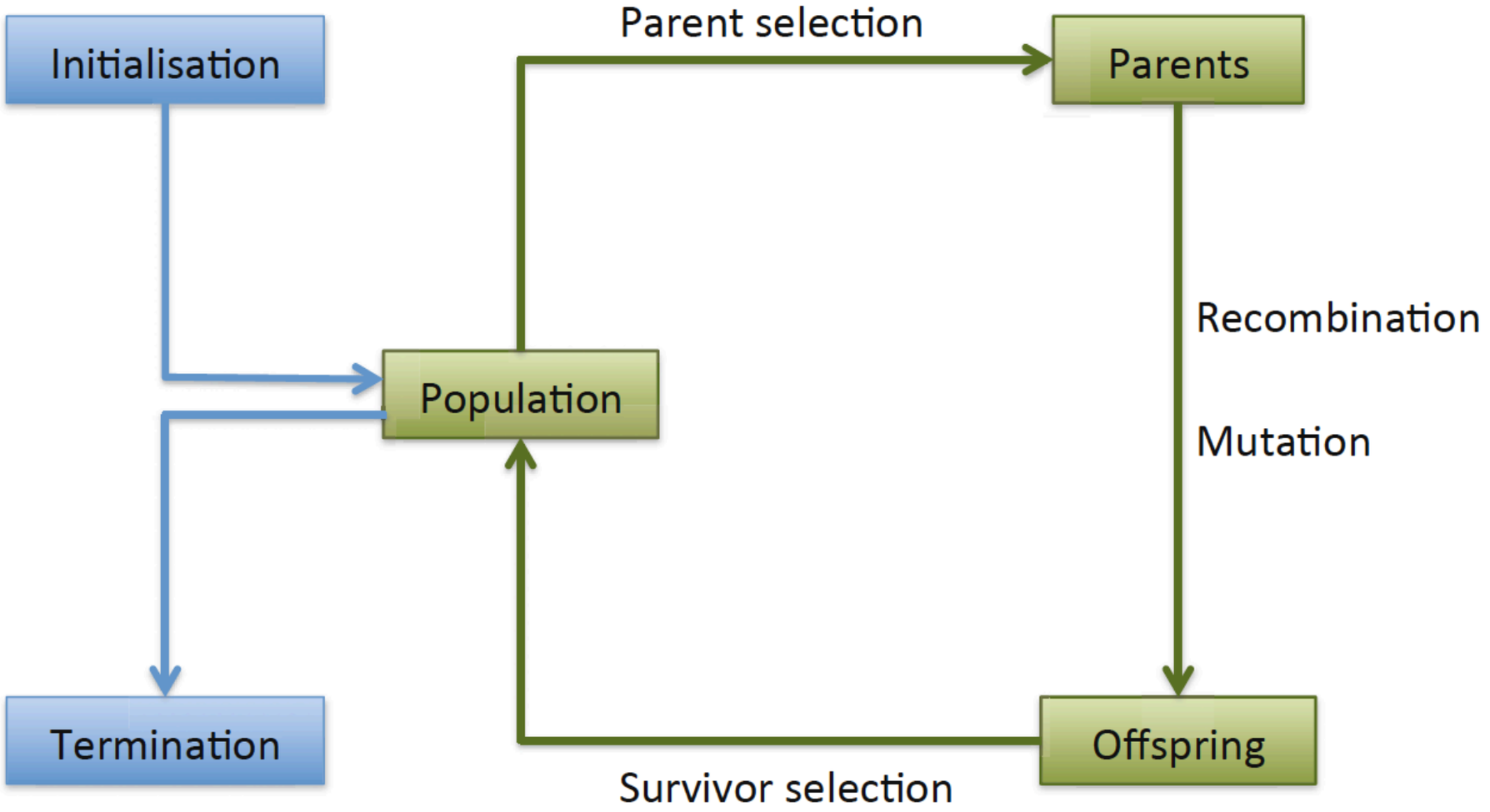
4: EA for Eight Queens Puzzle

- Eight Queens puzzle
- EA solution
- Typical EA behaviors
- EA and global optimization
- EA and neighborhood search
- Textbook Chapters 3.4.1, and 3.5

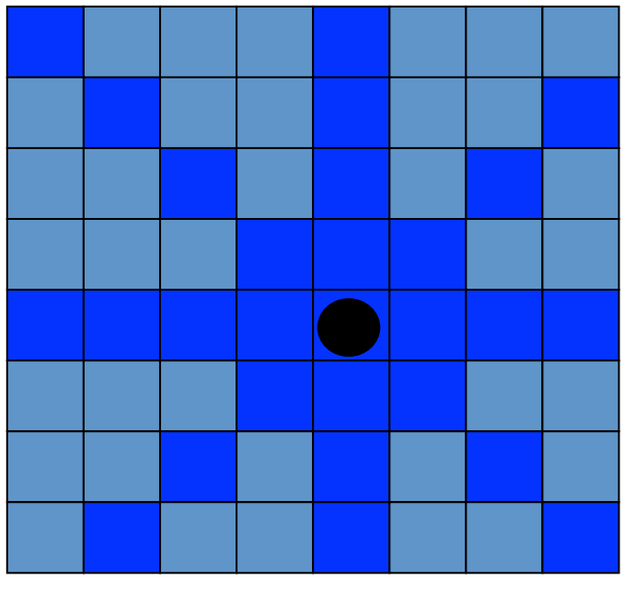
General scheme of EAs

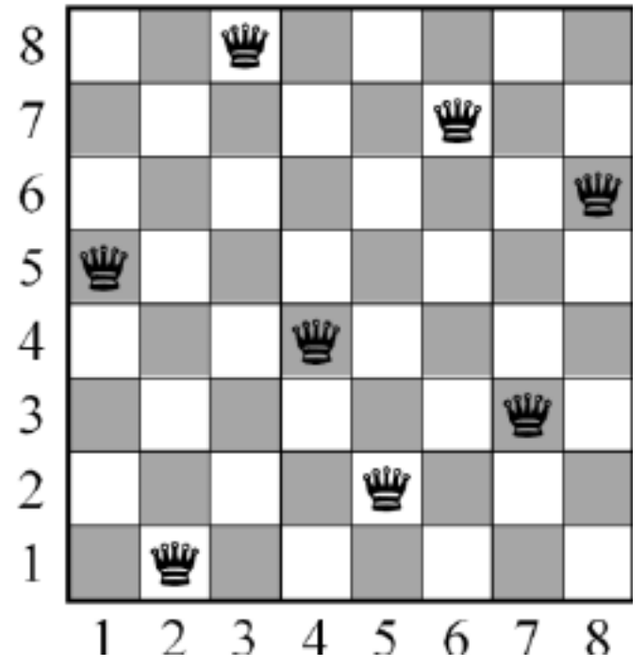


The eight queens puzzle

Task: place 8 queens on an 8 by 8 chessboard

Goal: no two queens check each other





Brute force: 64*63*62*61*60*59*58*57 = 178,462,987,637,760

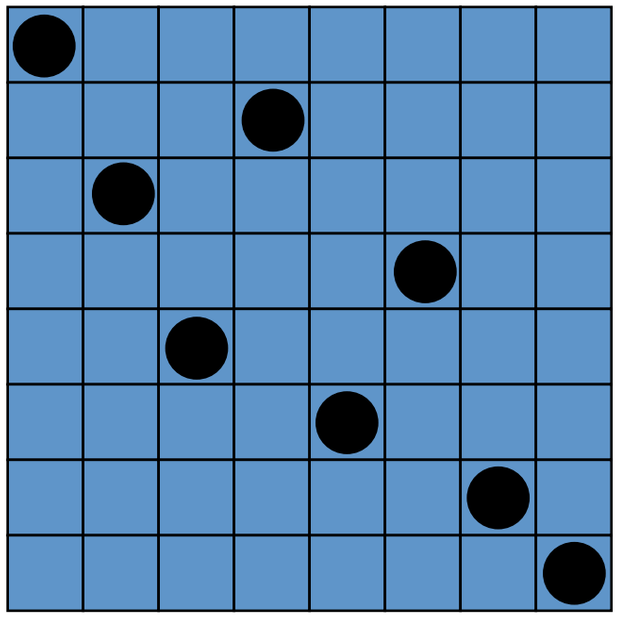
The eight queens puzzle: EA

- Represent the problem solution? phenotype? genotype?
- Evaluate fitness?
- Recombination and mutation?

The eight queens puzzle: representation

- Phenotype

  a board configuration



- Genotype

  a permutation of the numbers 1 to 8

1	3	5	2	6	4	7	8

The eight queens puzzle: fitness

- Penalty of a configuration: the number of checking queen pairs
- Penalty is to be minimized
- Fitness of a configuration: inverse penalty to be maximized

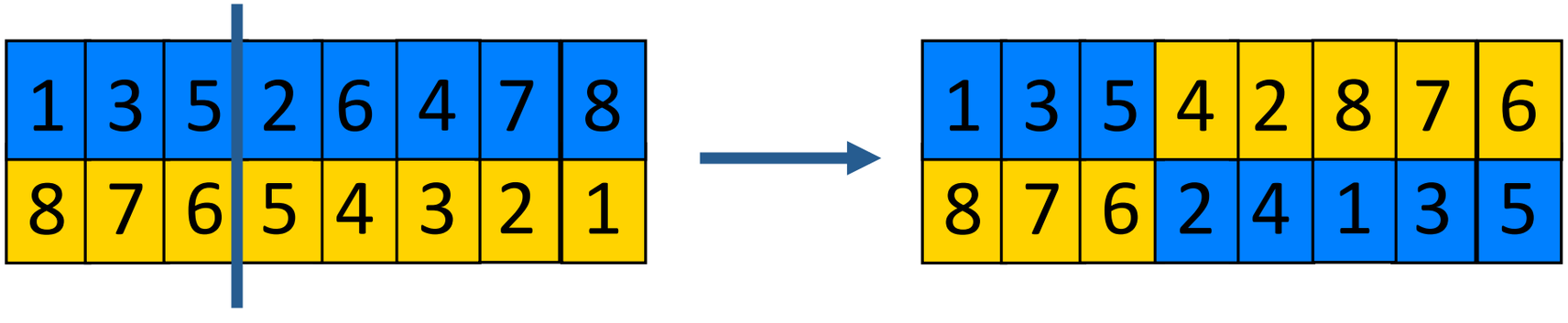
The eight queens puzzle: mutation

- Operates on the genotype
- Small variation in one permutation

- swapping values of two randomly chosen positions

The eight queens puzzle: recombination

- Operates on the genotype
- Combine two permutations into two new ones
  - choose random crossover point
  - copy first parts into offspring
  - create second part by inserting values from the other parent (in the original order, beginning after crossover point, skipping values already in the offspring)
  - "***cut and crossfill***"



The eight queens puzzle: selection

- Parent selection
  - Pick five parents and take the fittest two to undergo crossover and mutation
- Survivor selection
  - Insert a new offspring into the population to replace an existing member by
    - sorting the whole population by decreasing fitness
    - enumerating the list from high to low
    - replacing the first member with a fitness lower than the given offspring

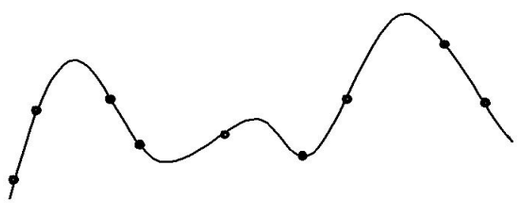
The eight queens puzzle: summary

Representation	Permutations
Recombination	'Cut-and-crossfill' crossover
Recombination probability	100%
Mutation	Swap
Mutation probability	80%
Parent selection	Best 2 out of random 5
Survival selection	Replace worst
Population size	100
Number of offspring	2
Initialisation	Random
Termination condition	Solution or 10,000 fitness evaluations

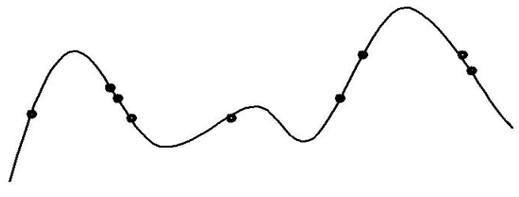
Table 3.4. Description of the EA for the eight-queens problem

Typical behavior of an EA

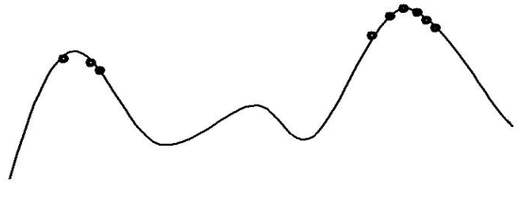
- Phrases in optimizing on a 1-dimensional fitness landscape
- Exploration vs. exploitation



Early phase:
quasi-random population distribution



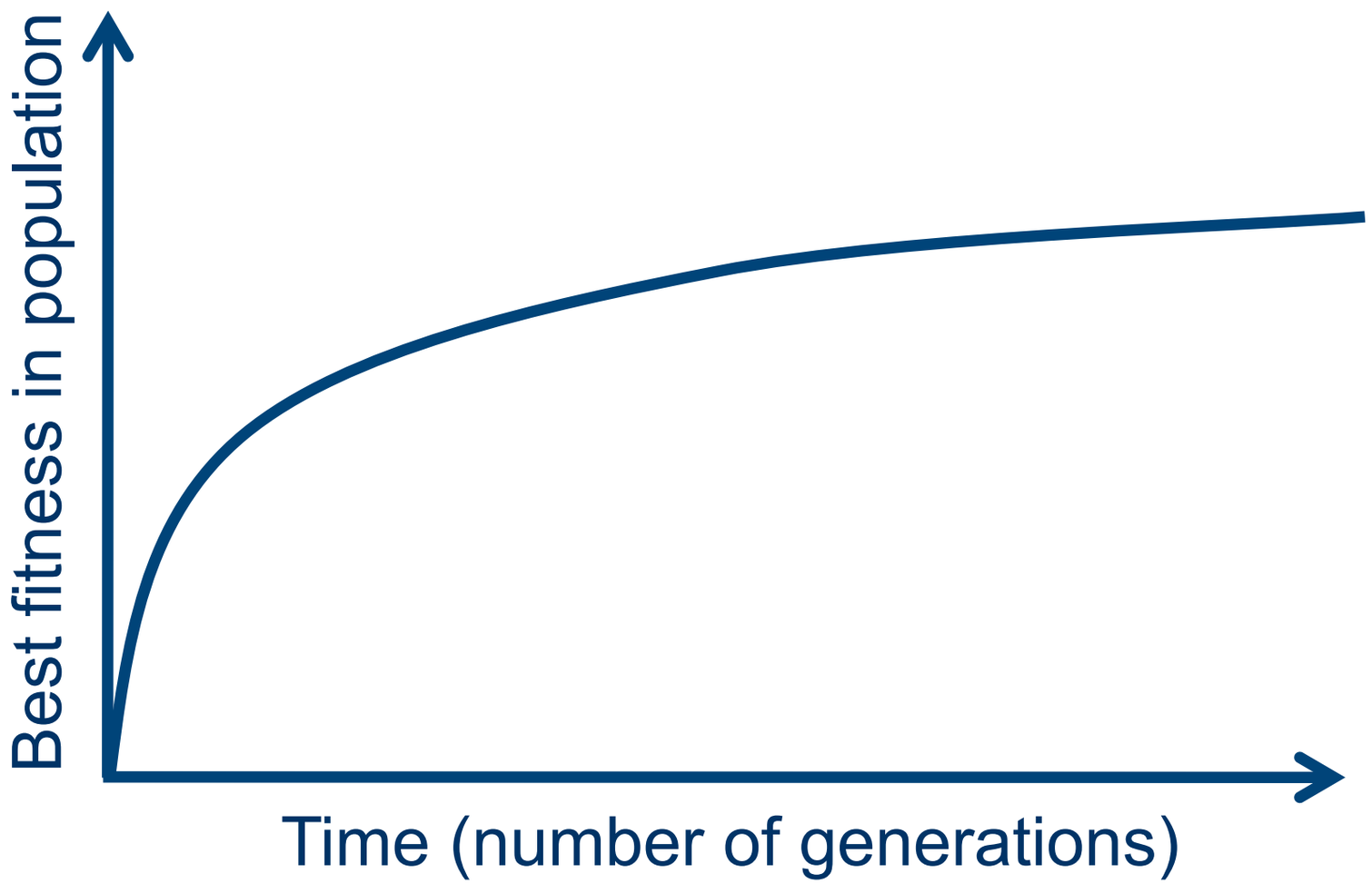
Middle phase:
population arranged around/on hills



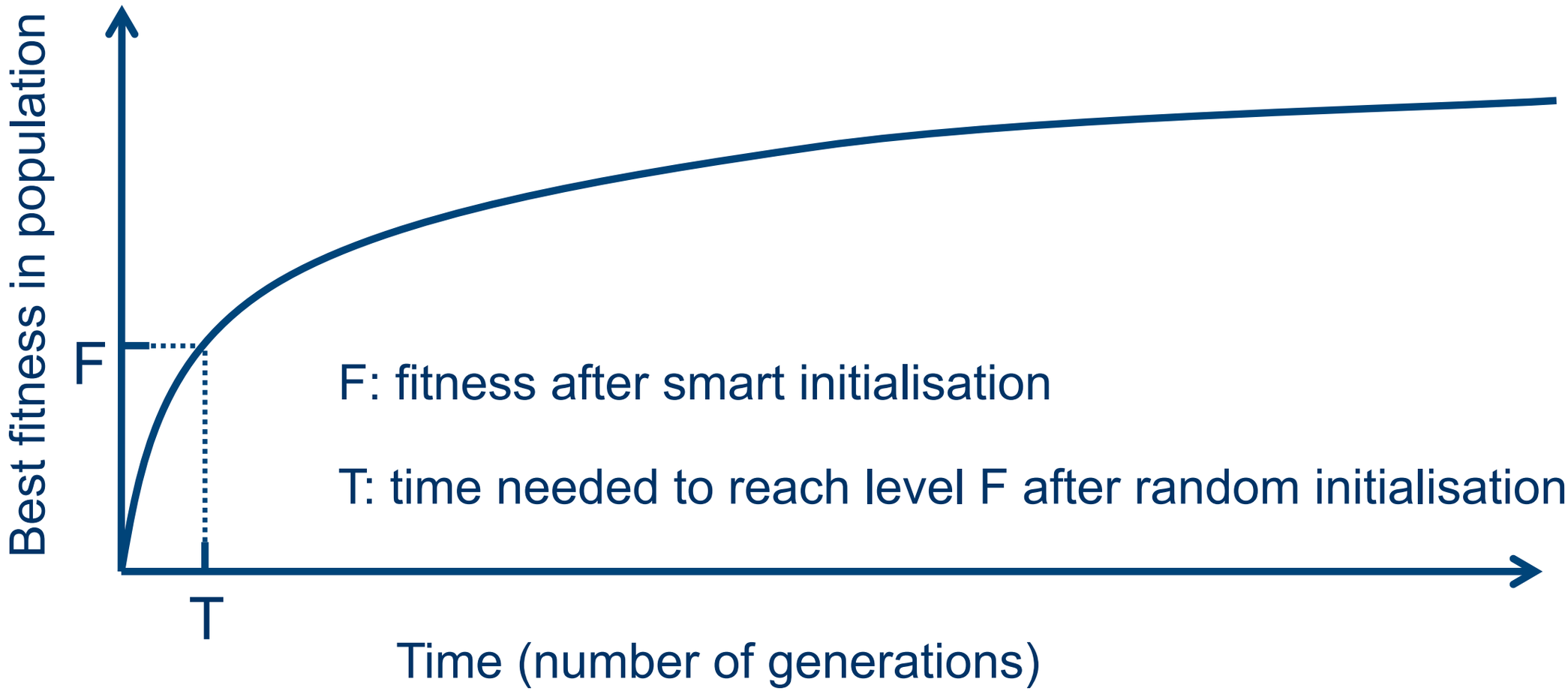
Late phase:
population concentrated on high hills

Typical run: progression of fitness

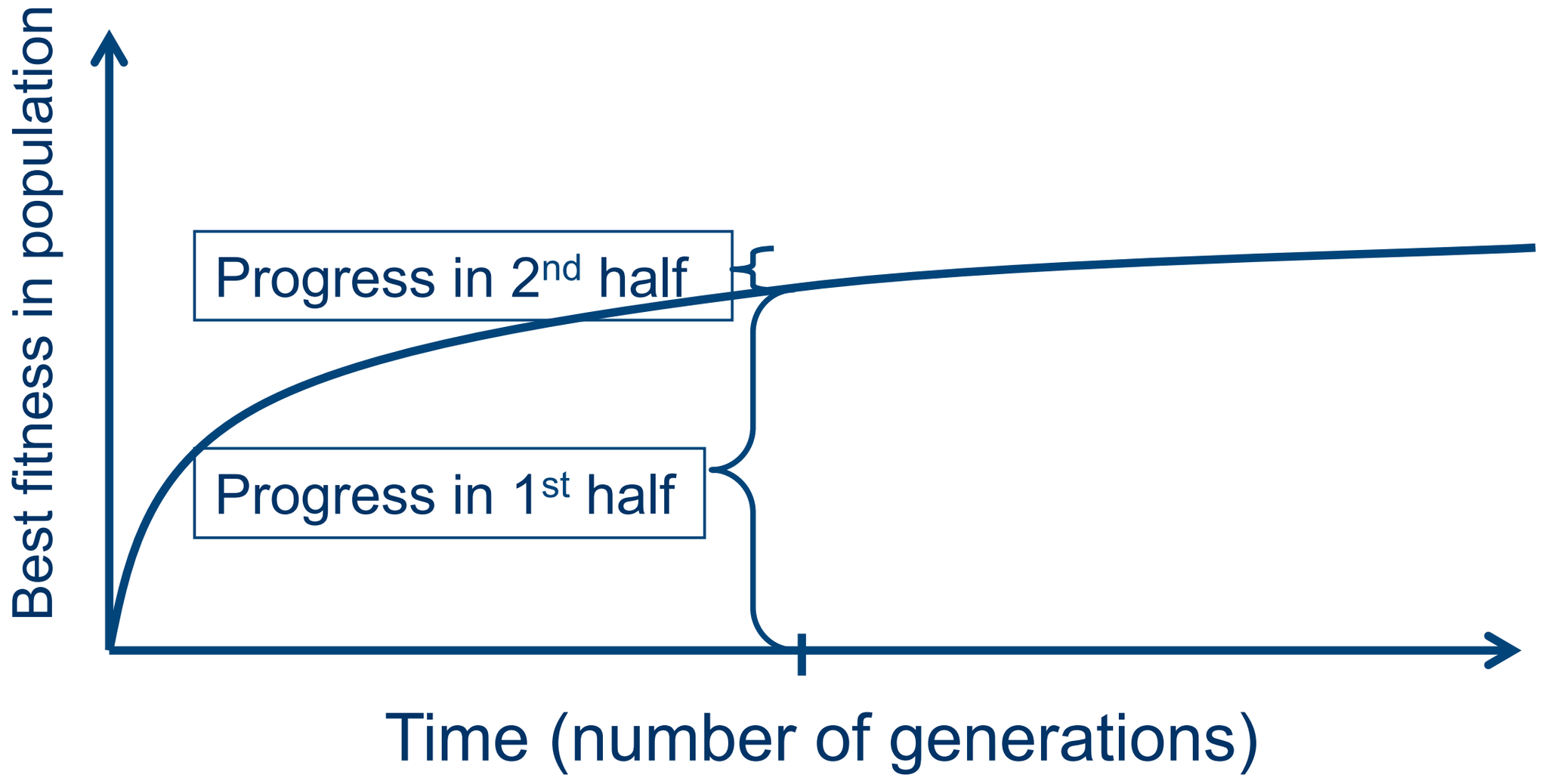
- Anytime behavior



Is it worth expending effort on smart initialization?



Are long runs beneficial?

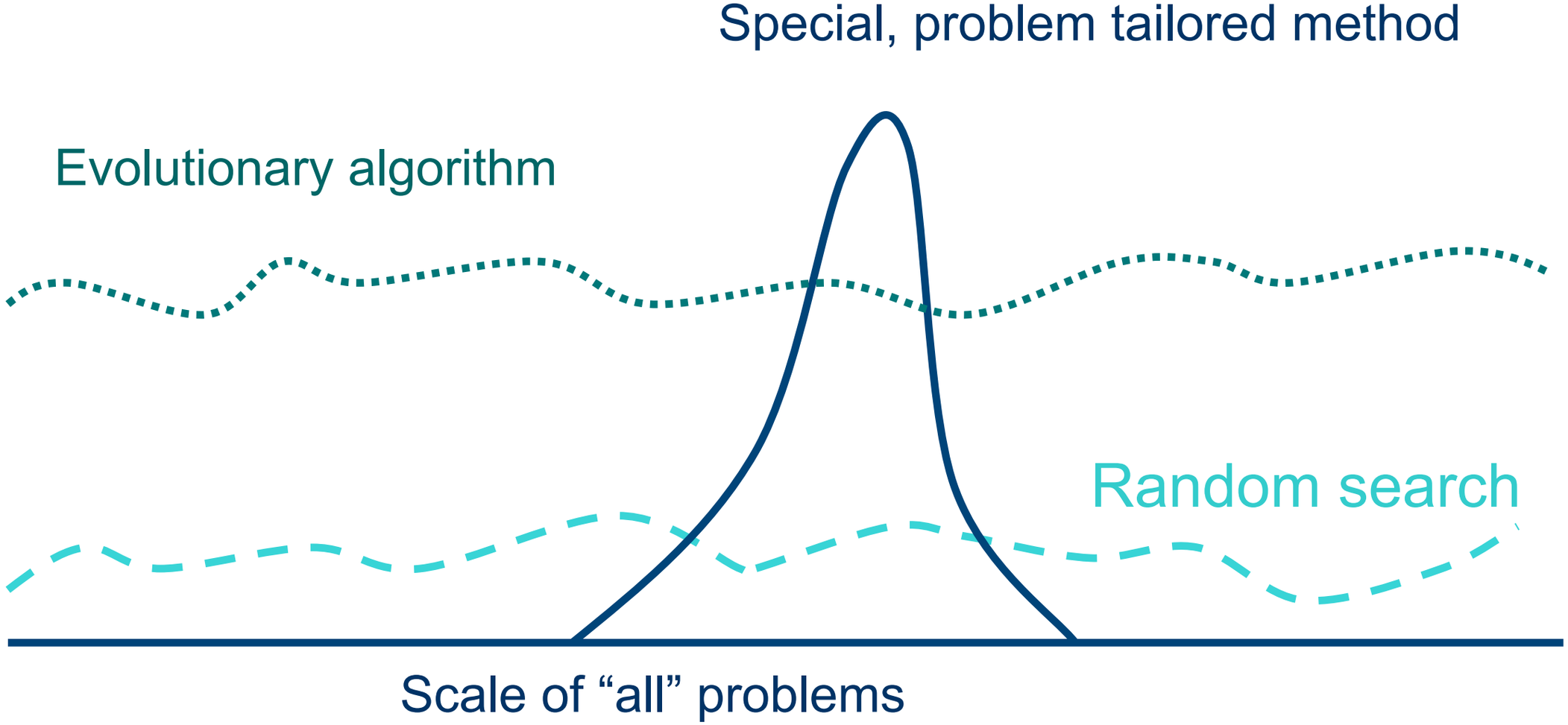


- It depends: how much you want the last bit of progress
- It may be better to do more shorter runs

EAs in context

- There are many views on the use of EAs as robust problem solving tools
- For most problems a problem-specific tool may:
    - perform better than a generic search algorithm on most instances
    - have limited utility
    - not do well on all instances
- Goal is to provide robust tools that have:
    - evenly good performance
    - over a range of problems and instances

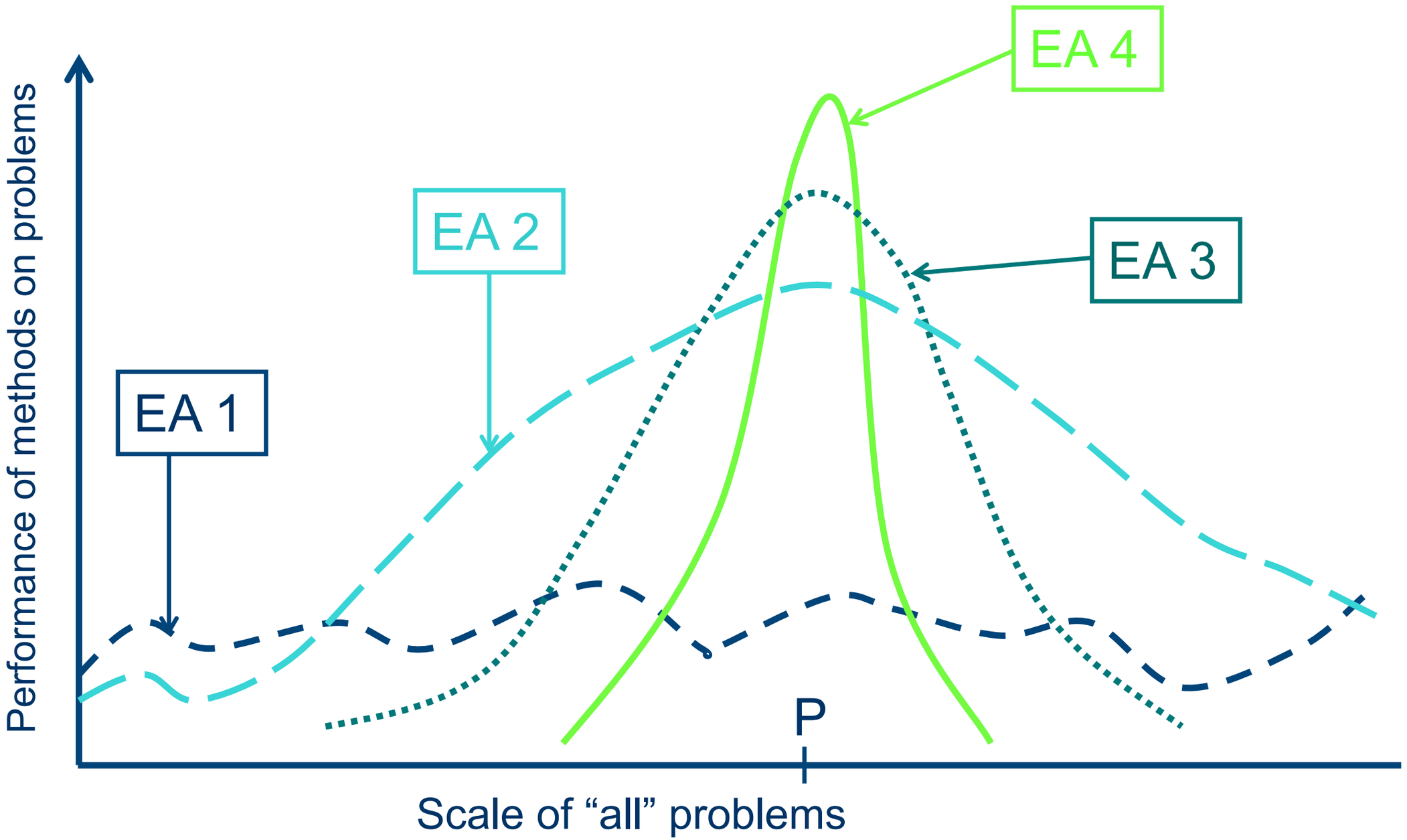
EAs as problem solvers: Goldberg's 1989 view



EAs and domain knowledge

- Trends:
    - adding problem specific knowledge to EAs
    - (special variation operators, repair, etc)
- Result: EA performance curve "deformation":
    - better on problems of the given type
    - worse on problems different from given type
    - amount of added knowledge is variable
- Theory suggests the search for an "all purpose" algorithm may be fruitless

EAs as problem solvers: Michalewicz's 1996 view



EC and global optimization

- Global optimization: search for finding best solution x^* out of some fixed set S
- Deterministic approaches allowed to run to completion
  - guarantee to find x^*, but may run in super-polynomial time
- Heuristic approaches (generate-and-test)
  - rules for deciding which x from S to generate next
  - no guarantees that best solutions found are globally optimal

EC and neighborhood search

- Many heuristics impose a neighborhood structure on *S*
- Such heuristics may guarantee that best point found is locally optimal, like hill-climbers
  - but problems often exhibit many local optima
  - often very quick to identify good solutions
- EAs are distinguished by:
  - use of population -> intrinsic parallelism
  - use of multiple search operators with different arity
  - stochastic operators of variation and selection