Evolutionary Computing

Chapter 8

Chapter 8:
Parameter Control

- Motivation
- Parameter setting
  - Tuning
  - Control
- Examples
- Where to apply parameter control
- How to apply parameter control

Motivation (1/2)

An EA has many strategy parameters, e.g.

- mutation operator and mutation rate
- crossover operator and crossover rate
- selection mechanism and selective pressure (e.g. tournament size)
- population size

Good parameter values facilitate good performance

Q1 How to find good parameter values ?

Motivation (2/2)

EA parameters are rigid (constant during a run)

BUT

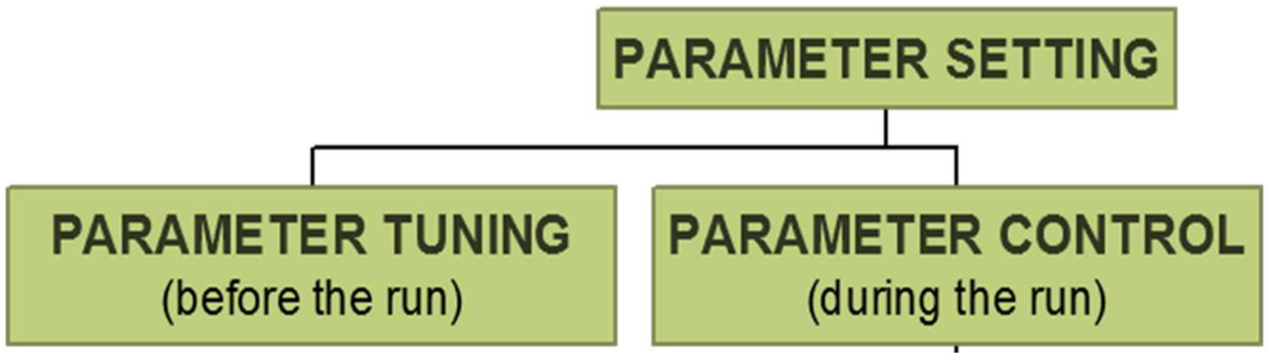
an EA is a dynamic, adaptive process

THUS

optimal parameter values may vary during a run

Q2: How to vary parameter values?

Parameter Setting



Parameter Settings: Tuning

Parameter tuning: the traditional way of testing and comparing different values before the “real” run

Problems:

- users mistakes in settings can be sources of errors or sub-optimal performance
- parameters interact: exhaustive search is not practicable
- costs much time even with “smart” tuning
- good values may become bad during the run

Parameter Settings: Control

Parameter control: setting values on-line, during the actual run, e.g.

- predetermined time-varying schedule p = p(t)
- using feedback from the search process
- encoding parameters in chromosomes and rely on selection

Problems:

- finding optimal p is hard, finding optimal p(t) is harder
- still user-defined feedback mechanism, how to ``optimize"?
- when would natural selection work for strategy parameters?

Examples: Varying mutation step size

Task to solve:

- min $f(x_1,\ldots,x_n)$
- $L_i \le x_i \le U_i$ for $i = 1,\ldots,n$ bounds
- $g_i(x) \le 0$ for $i = 1,\ldots,q$ inequality constraints
- $h_i(x) = 0$ for $i = q+1,\ldots,m$ equality constraints

Algorithm:

- EA with real-valued representation $(x_1,\ldots,x_n)$
- arithmetic averaging crossover
- Gaussian mutation: $x'_i = x_i + N(0, \sigma)$
  standard deviation σ is called mutation step size

Examples:
Varying mutation step size, option 1

Replace the constant σ by a function σ(t)

$$\sigma(t) = 1 - 0.9 \times \frac{t}{T}$$

$0 \leq t \leq T$ is the current generation number

- Features:
  - changes in σ are independent from the search progress
  - strong user control of σ by the above formula
  - σ is fully predictable
  - a given σ acts on all individuals of the population

Examples:
Varying mutation step size, option 2

Replace the constant σ by a function σ(t) updated after every n steps by the 1/5 success rule:

$$\sigma(t) = \begin{cases} \sigma(t-n)/c & \text{if } p_s > 0.2 \\ \sigma(t-n)\cdot c & \text{if } p_s < 0.2 \\ \sigma(t-n) & \text{otherwise} \end{cases}$$

- Features:
  - changes in σ are based on feedback from the search progress
  - some user control of σ by the above formula
  - σ is not predictable
  - a given σ acts on all individuals of the population

Examples:
Varying mutation step size, option 3

- Assign a personal σ to each individual
- Incorporate this σ into the chromosome: $(x_1, \ldots, x_n, \sigma)$
- Apply variation operators to x_i's and σ

$$\sigma' = \sigma \times e^{N(0,\sigma)}$$

$$x_i' = x_i + N(0, \sigma')$$

- Features:
  - changes in σ are results of natural selection
  - (almost) no user control of σ
  - σ is not predictable
  - a given σ acts on one individual

Examples:
Varying mutation step size, option 4

Assign a personal σ to each variable in each individual

Incorporate σ's into the chromosomes: $(x_1, \ldots, x_n, \sigma_1, \ldots, \sigma_n)$

Apply variation operators to x_i's and σ_i's

$$\sigma_i' = \sigma_i \times e^{N(0,\tau)}$$

$$x_i' = x_i + N(0, \sigma_i')$$

- Features:
  - changes in σ_i are results of natural selection
  - (almost) no user control of σ_i
  - σ_i is not predictable
  - a given σ_i acts on one gene of one individual

Examples:
Varying penalties

Constraints

- $g_i(x) \leq 0$ for $i = 1,\ldots,q$ inequality constraints
- $h_i(x) = 0$ for $i = q+1,\ldots,m$ equality constraints

are handled by penalties:

$$eval(x) = f(x) + W \times penalty(x)$$

where

$$penalty(x) = \sum_{j=1}^{m} \begin{cases} 1 & \textit{for violated constraint} \\ 0 & \textit{for satisfied constraint} \end{cases}$$

Examples: Varying penalties, option 1

Replace the constant W by a function W(t)

$$W(t) = (\mathrm{C} \times t)^{\alpha}$$

$0 \leq t \leq T$ is the current generation number

- Features:
  - changes in W independent from the search progress
  - strong user control of W by the above formula
  - W is fully predictable
  - a given W acts on all individuals of the population

Examples:
Varying penalties, option 2

Replace the constant W by W(t) updated in each generation

$$W(t+1) = \begin{cases} \beta \times W(t) & \text{if last k champions all feasible} \\ \gamma \times W(t) & \text{if last k champions all infeasible} \\ W(t) & \text{otherwise} \end{cases}$$

$\beta < 1$, $\gamma > 1$, $\beta \times \gamma \neq 1$ champion: best of its generation

- Features:
  - changes in W are based on feedback from the search progress
  - some user control of W by the above formula
  - W is not predictable
  - a given W acts on all individuals of the population

Examples: Varying penalties, option 3

Assign a personal W to each individual

Incorporate this W into the chromosome: $(x_1, \ldots, x_n, W)$

Apply variation operators to x_i's and W

Alert:

$$eval((x, W)) = f(x) + W \times penalty(x)$$

while for mutation step sizes we had

$$eval((x, \sigma)) = f(x)$$

this option is thus sensitive "cheating" $\Rightarrow$ makes no sense

Examples:
Lessons learned (1/2)

Various forms of parameter control can be distinguished by:

- primary features:
  - what component of the EA is changed
  - how the change is made
- secondary features:
  - evidence/data backing up changes
  - level/scope of change

Examples:
Lessons learned (2/2)

Various forms of parameter control can be distinguished by:

	$\sigma(t) = 1-0.9*t/T$	$\sigma' = \sigma/c$, if $r > 1/5$ ...	$(x_1, ..., x_n, \sigma)$	$(x_1, ..., x_n, \sigma_1, ..., \sigma_n)$	$W(t) = (C*t)^\alpha$	$W'=\beta*W$, if $b_i \in F$	$(x_1, ..., x_n, W)$
What	Step size	Step size	Step size	Step size	Penalty weight	Penalty weight	Penalty weight
How	Deterministic	Adaptive	Self-adaptive	Self-adaptive	Deterministic	Adaptive	Self-adaptive
Evidence	Time	Successful mutations rate	(Fitness)	(Fitness)	Time	Constraint satisfaction history	(Fitness)
Scope	Population	Population	Individual	Gene	Population	Population	Individual

Where to apply parameter control

Practically any EA component can be parameterized and thus controlled on-the-fly:

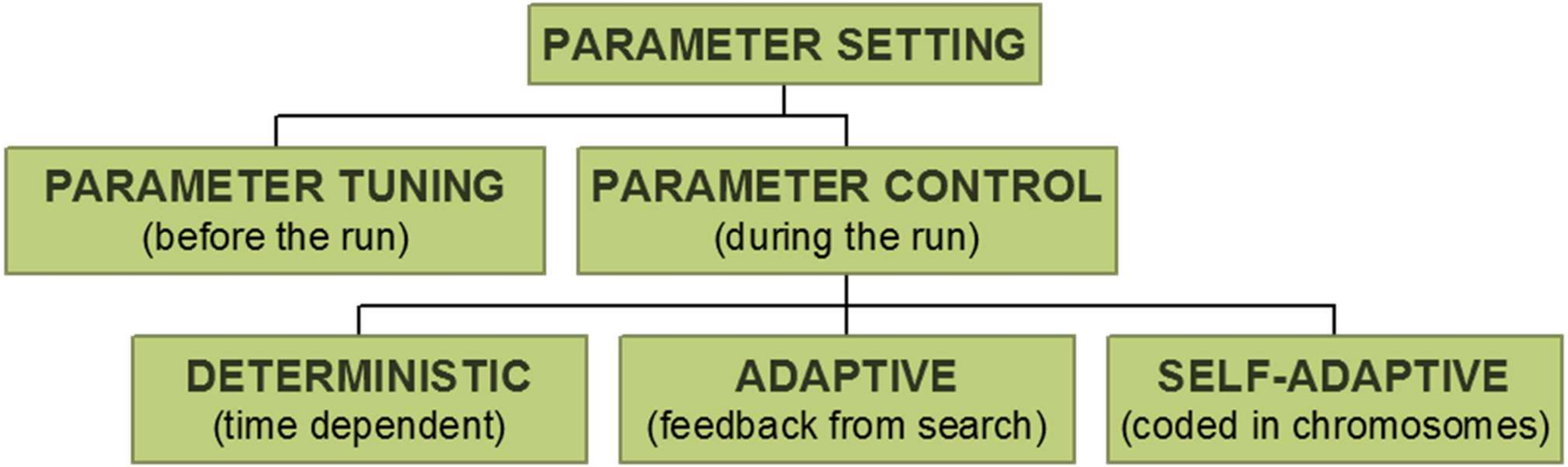
- representation
- evaluation function
- variation operators
- selection operator (parent or mating selection)
- replacement operator (survival or environmental selection)
- population (size, topology)

How to apply parameter control

Three major types of parameter control:

- deterministic: some rule modifies strategy parameter without feedback from the search (based on some counter)
- adaptive: feedback rule based on some measure monitoring search progress
- self-adaptative: parameter values evolve along with solutions; encoded onto chromosomes they undergo variation and selection

How to apply parameter control
Global taxonomy



Evidence: Informing the change (1/2)

The parameter changes may be based on:

- time or nr. of evaluations (deterministic control)
- population statistics (adaptive control)
  - progress made
  - population diversity
  - gene distribution, etc.
- relative fitness of individuals creeated with given values (adaptive or self-adaptive control)

Evidence: Informing the change (2/2)

- Absolute evidence: predefined event triggers change, e.g. increase p_m by 10% if population diversity falls under threshold x
- Direction and magnitude of change is fixed
- Relative evidence: compare values through solutions created with them, e.g. increase p_m if top quality offspring came by high mutation rates
- Direction and magnitude of change is not fixed

Evidence: Refined taxonomy

- Combinations of types and evidences
  - Possible: +
  - Impossible: -

	Deterministic	Adaptive	Self-adaptive
Absolute	+	+	-
Relative	-	+	+

Scope/level

The parameter may take effect on different levels:

- environment (fitness function)
- population
- individual
- sub-individual

Note: given component (parameter) determines possibilities

Thus: scope/level is a derived or secondary feature in the classification scheme

Evaluation/Summary

- Parameter control offers the possibility to use appropriate values in various stages of the search

- Adaptive and self-adaptive parameter control
  - offer users “liberation” from parameter tuning
  - delegate parameter setting task to the evolutionary process
  - the latter implies a double task for an EA: problem solving + self-calibrating (overhead)