Parameter control

Chapter 8

Motivation 1

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

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 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
- costs much time
- parameters interact: exhaustive search is not practicable
- good values may become bad during the run

Parameter 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 natural 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?

Example

Task to solve:

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

Algorithm:

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

Varying mutation step size: option1

Replace the constant σ by a function $\sigma(t)$

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

$0 \le t \le T$ 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

Varying mutation step size: option2

Replace the constant σ by a function $\sigma(t)$ updated after every n steps by the 1/5 success rule (cf. ES chapter):

$$\sigma(t) = \begin{cases} \sigma(t-n)/c & \text{if } p_s > 1/5 \\ \sigma(t-n)\cdot c & \text{if } p_s < 1/5 \\ \sigma(t-n) & \text{otherwise} \end{cases}$$

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

Varying mutation step size: option3

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

$$\sigma' = \sigma \times e^{N(0,\tau)}$$
$$x'_i = x_i + N(0,\sigma')$$

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

A.E. Eiben and J.E. Smith, Introduction to Evolutionary Computing
Parameter Control in EAs

Varying mutation step size: option4

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 1 gene of one individual

A.E. Eiben and J.E. Smith, Introduction to Evolutionary Computing
Parameter Control in EAs

Example cont'd

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

are handled by penalties:

eval(x) = f(x) + W × penalty(x)

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

A.E. Eiben and J.E. Smith, Introduction to Evolutionary Computing
Parameter Control in EAs

Varying penalty: option 1

Replace the constant W by a function W(t)

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

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

Features:
- changes in W are 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

Varying penalty: option 2

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

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

$\beta < 1$, $\gamma > 1$, $\beta \times \gamma \neq 1$ 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

Varying penalty: 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* × *penalty(x)*

while for mutation step sizes we had

eval ((x, σ)) = *f* (x)

this option is thus sensitive “cheating” ⇒ makes no sense

Lessons learned from examples

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

What

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

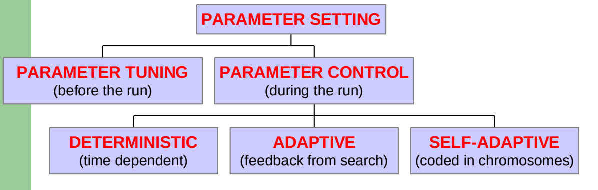
- representation
- evaluation function
- variation operators
- selection operator (parent or mating selection)
- replacement operator (survival or environmental selection)
- population (size, topology)

How

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

Global taxonomy



Evidence informing the change

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

- 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 mut. Rates
- Direction and magnitude of change is not fixed

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

Refined taxonomy

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

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

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)
- Robustness, insensivity of EA for variations assumed
  - If no. of parameters is increased by using (self)adaptation
  - For the "meta-parameters" introduced in methods