Tutorial for

# Introduction to Computational Intelligence in Winter 2011/12

Günter Rudolph, Nicola Beume

http://ls11-www.cs.tu-dortmund.de/people/rudolph/teaching/lectures/CI/WS2011-12/lecture.jsp

## Sheet 6, Block C

11.01.2012

**Return: 23.01.2012**
**Discuss: 25.01.2012**

### Exercise 6.1: Scheduling with Constraints (5 Points)

Develope and decribe (only conceptually, no implementation required) at least two mutation and at least two crossover operators for scheduling problems with precedence constraints. Demonstrate your operators at the following example problems a) and b).

Consider jobs A, B, . . . , L that shall be reordered such that a scheduling function is optimal. A valid solution has to fulfil the conditions:

a) A before C

b) A before C, D before E, G before L

### Exercise 6.2: Basic Probability Theory (5 Points)

Consider standard-bit-mutation on a bitstring of length $n$ where the probability of flipping is $p = 1/n$ for each bit.

a) Calculate the expected number of flipping bits per mutation.

b) Calculate the probability that exactly $k$ bits of the bitstring are flipped in one mutation.

c) Calculate the probability that a certain bit is flipped at least once within $t$ mutations.

d) Given a bitstring $x$, calculate the probability that a certain bitstring $y$ is the result of one mutation of $x$. Hint: Use the Hamming distance to relate bitstrings to each other.

**Exercise 6.3: Runtime Analysis** (10 Points)

Use the method of fitness-based partitions to calculate an upper bound for the expected runtime of the (1+1)-EA (with standard-bit-mutation) on the following maximization problems. Give the partitioning and comment your calculations detailedly. Describe what kind of steps are helpful (which bits shall flip) in what kind of situations (state of the individual).

Let $\mathbf{x} = (x_1, \ldots, x_n)$ with $x_i \in \{0, 1\}$ for $i \in \{1, 2, \ldots, n\}$.

a)

$$f(\mathbf{x}) = \sum_{i=1}^{n} \prod_{j=i}^{n} (1 - x_j)$$

b)

$$g(\mathbf{x}) = \begin{cases} n + i & \text{if } x = 1^i 0^{n-i} \text{ with } i \in \{1, 2, \ldots, n\} \\ n - \sum\limits_{i=1}^{n} x_i & \text{otherwise} \end{cases}$$

c)

$$h_k(\mathbf{x}) = \begin{cases} n - \sum\limits_{i=1}^{n} x_i & \text{if } n - k < \sum\limits_{i=1}^{n} x_i < n \\ k + \sum\limits_{i=1}^{n} x_i & \text{otherwise} \end{cases}$$

$k \in \{1, \ldots, n\}$ is a parameter of the function family and shall be fixed before the optimization starts. Hint: Choose a constant value, e.g. $k = 5$ to make the function more accessible, then consider the general case with arbitrary $k$.