Artificial Intelligence 2, Lecture 10, 20101209

Algorithm performance

Course summary

Mattias Wahde, PhD, associate professor, Chalmers University of Technology
e-mail: mattias.wahde@chalmers.se www: www.me.chalmers.se/~mwahde

$$\Psi_2(x_1,x_2) = 100\left(x_1^2 - x_2\right)^2 + (1-x_1)^2,$$

$$\Psi_3(x_1,x_2) = 0.5 - \frac{\left(\sin\sqrt{x_1^2+x_2^2}\right)^2 - 0.5}{\left(1+0.001\left(x_1^2+x_2^2\right)\right)^2},$$

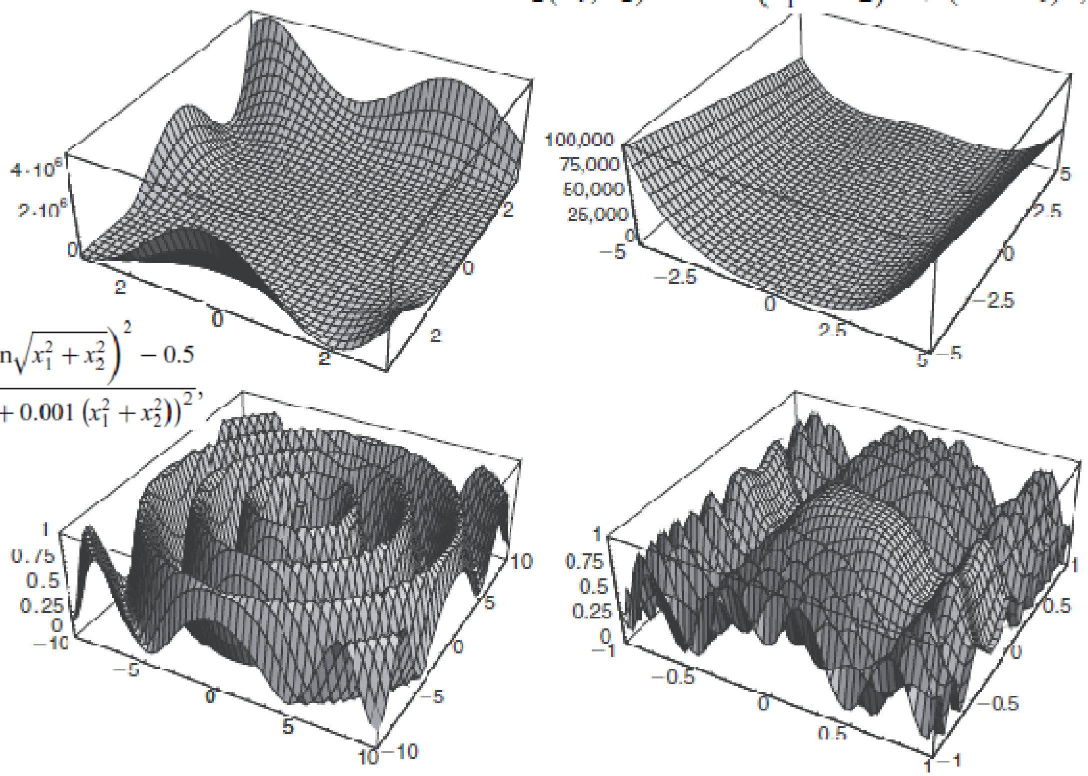


Figure D.1: *Four of the five benchmark functions defined in the text. Top left panel: the Goldstein–Price function; top right panel: the Rosenbrock function, with $n = 2$; bottom left panel: the Sine square function; bottom right panel: the benchmark function $\Psi_5^{[2]}$, with $a = 0.05$ and $b = 10$.*

Mattias Wahde, PhD, associate professor, Chalmers University of Technology
e-mail: mattias.wahde@chalmers.se www: www.me.chalmers.se/~mwahde

$\Psi_2^{[5]}(x_1,\ldots,x_5)$					
Method	Settings	N	I	Avg.	S.D.
GA	Binary	40	250	19.06	40.64
GA	Binary	100	100	3.849	7.138
GA	Binary	250	40	2.401	3.403
GA	RN, $p_{cr} = 0.8$, $C_r = 0.2$	250	40	5.404	1.682
GA	RN, $p_{cr} = 0.8$, $C_r = 0.02$	250	40	1.829	1.244
PSO	–	20	500	0.727	1.161
PSO	–	40	250	0.689	0.845
PSO	–	100	100	2.600	1.325
PSO	–	250	40	55.90	31.64
RS	–	1	10,000	95.19	48.74

$\Psi_5^{[10]}(x_1, \ldots, x_{10})$					
Method	Settings	N	I	Avg.	S.D.
GA	Binary	40	250	0.817	0.058
GA	Binary	100	100	0.875	0.048
GA	Binary	250	40	0.875	0.038
GA	RN, $p_{cr} = 0.8$, $C_r = 0.2$	250	40	0.747	0.034
GA	RN, $p_{cr} = 0.8$, $C_r = 0.02$	250	40	0.851	0.044
PSO	–	20	500	0.898	0.049
PSO	–	40	250	0.881	0.049
PSO	–	100	100	0.786	0.035
PSO	–	250	40	0.684	0.030
RS	–	1	10,000	0.687	0.026

Mattias Wahde, PhD, associate professor, Chalmers University of Technology
e-mail: mattias.wahde@chalmers.se www: www.me.chalmers.se/~mwahde

Table 6.5: *Results obtained by applying different algorithms to the instance of the TSP shown in Fig. 6.2.*

Method	Settings	N	I	Avg.	S.D.
AS	$\alpha = 1$, $\beta = 2$, $\tau_0 = 0.454$	100	100	179.68	2.81
AS	$\alpha = 1$, $\beta = 5$, $\tau_0 = 0.454$	100	100	173.84	1.89
MMAS	$\alpha = 1$, $\beta = 5$, $\tau_0 = 0.00909$	100	100	193.58	2.67
GA-R	$p_c = 0.00$, $p_{mut} = 0.01$	50	200	577.18	16.94
GA-R	$p_c = 0.00$, $p_{mut} = 0.03$	50	200	658.40	25.45
GA-R	$p_c = 0.20$, $p_{mut} = 0.01$	100	100	647.57	18.73
GA-NN	$p_c = 0.00$, $p_{mut} = 0.01$	50	200	217.89	16.04
GA-NN	$p_c = 0.00$, $p_{mut} = 0.01$	100	100	211.67	12.62
GA-NN	$p_c = 0.20$, $p_{mut} = 0.01$	100	100	216.68	10.29
GA-NN	$p_c = 0.00$, $p_{mut} = 0.01$	200	50	206.57	10.13

For each method and parameter setting, 30 runs were carried out, each run evaluating 10,000 paths. I denotes the number of iterations (i.e. generations, in the case of the GA). The two rightmost columns show the average and estimated standard deviation of the resulting distribution of path lengths. See the main text for a complete description of the parameter settings for each method.

Mattias Wahde, PhD, associate professor, Chalmers University of Technology
e-mail: mattias.wahde@chalmers.se www: www.me.chalmers.se/~mwahde

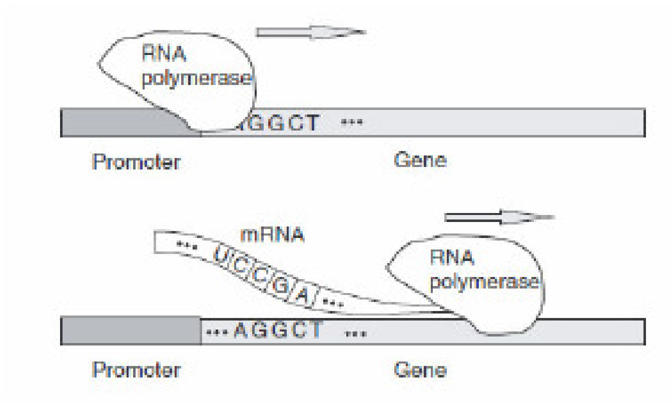
Course summary

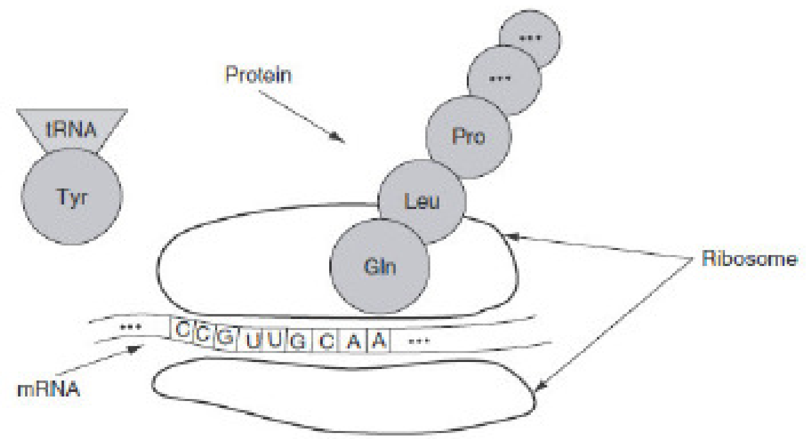
- Main topics
  - Evolutionary algorithms
    - Genetic algorithms
    - Linear genetic programming
  - Particle swarm optimization
  - Ant colony optimization
    - AS
    - MMAS
  - Neural networks
    - Delta rule

Mattias Wahde, PhD, associate professor, Chalmers University of Technology
e-mail: mattias.wahde@chalmers.se www: www.me.chalmers.se/~mwahde

Evolutionary algorithms

- Biological background
  - Terminology (fitness, populations, mutation etc.)
  - Basics of the functionality of genes in biological systems (transcription, translation etc.)



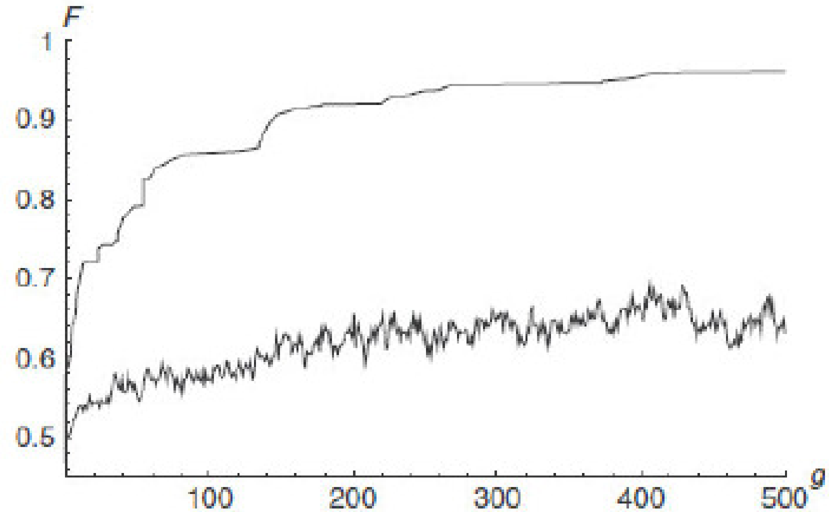


Evolutionary algorithms

- Genetic algorithms – concepts and operators
  - Encoding methods
    - Binary encoding
    - Real-number encoding
  - Fitness
  - Selection methods
    - Roulette-wheel
    - Tournament selection
  - Crossover
  - Mutation
    - Standard
    - Creep

Evolutionary algorithms

- Genetic algorithms – concepts and operators
  - Replacement
    - Generational
    - Steady-state
  - Elitism



Mattias Wahde, PhD, associate professor, Chalmers University of Technology
e-mail: mattias.wahde@chalmers.se www: www.me.chalmers.se/~mwahde

Evolutionary algorithms

- Genetic algorithms – Properties
  - The schema theorem: Derivation and interpretation

$$E(\Gamma(S,g+1)) \geq \frac{\overline{F}_S}{\overline{F}}\Gamma(S,g)\left(1-p_c\frac{d(S)}{m-1}\right)(1-p_{\text{mut}})^{o(S)},$$

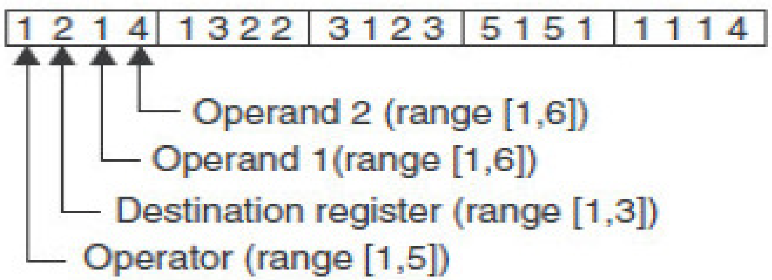
  - Infinite population models: Selection and (simplified) mutation for functions of unitation (e.g. Onemax)

$$\mathcal{G}_s(p)(j) = \frac{jp(j)}{\sum_{j=0}^{m} jp(j)}$$

  - Estimating optimal mutation rates and runtime (number of evaluations).

Evolutionary algorithms

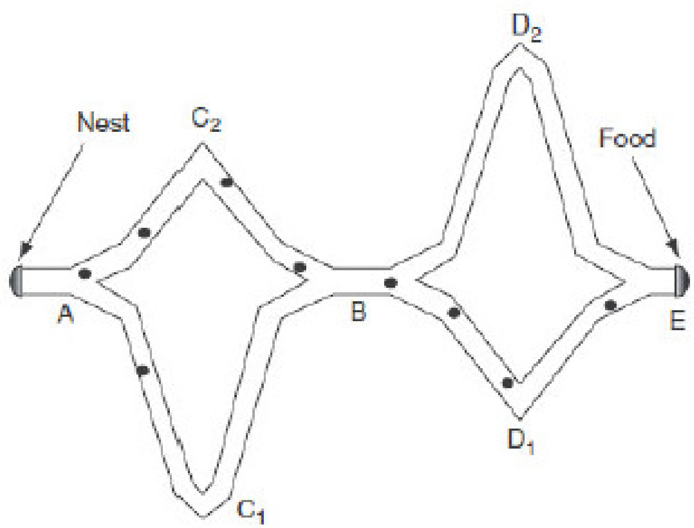
- Linear genetic programming
  - Encoding
  - Chromosome interpretation (decoding)
  - Registers
  - Crossover
  - Mutation



Ant colony optimization

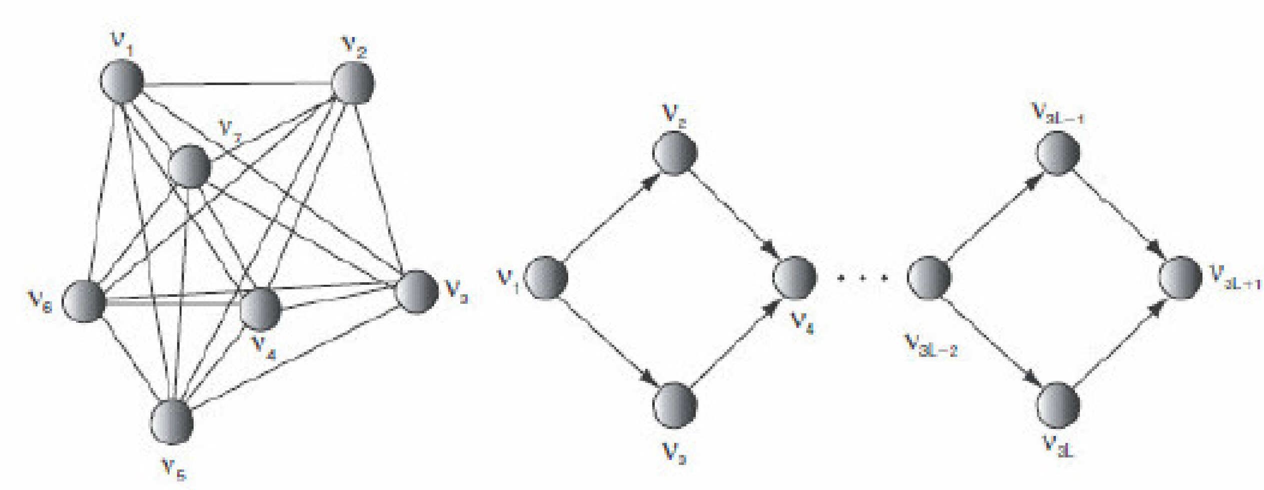
- Biological background
  - Stigmergic communication
  - The phenomenological model for ant foraging behavior.

$$p_1^S = \frac{(C+S_1)^m}{(C+S_1)^m + (C+L_1)^m},$$



Ant colony optimization

- Ant algorithms (ant colony optimization)
  - Problem formulation, construction graphs



Ant colony optimization

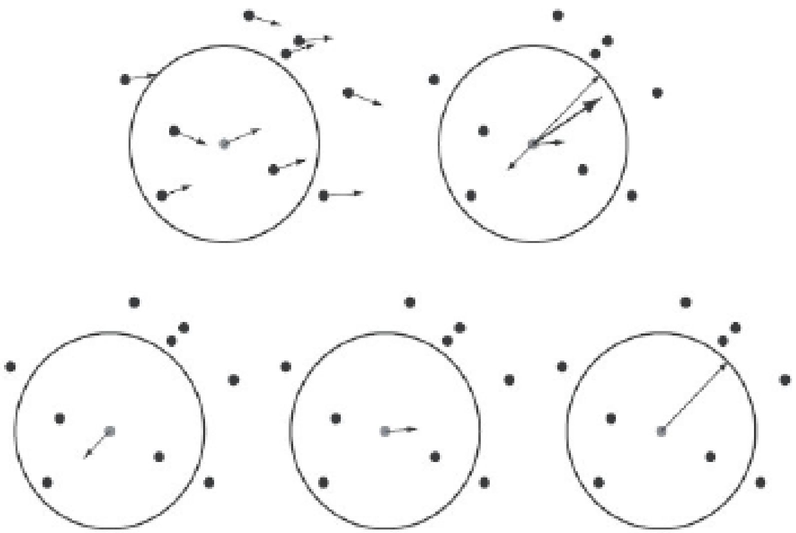
- Ant system
  - Pheromone initialization
  - Probabilistic path generation
  - Updating the pheromone level
  - Typical parameter settings

$$p(e_{ij}|S) = \frac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_{\nu_l \notin L_T(S)} \tau_{lj}^{\alpha}\eta_{lj}^{\beta}}.$$

- The traveling salesman problem (TSP)

Particle swarm optimization

- Biological background
  - Swarming in biological organisms
  - The Boids model (basic concepts)



Particle swarm optimization

- Algorithm:
  - Initialization
  - Updating the best positions (individual particles and best-in-swarm)
  - Updating velocities and positions (+ velocity restriction)

$$v_{ij} \leftarrow w v_{ij} + c_1 q \left(\frac{x_{ij}^{\mathrm{pb}} - x_{ij}}{\Delta t} \right) + c_2 r \left(\frac{x_j^{\mathrm{sb}} - x_{ij}}{\Delta t} \right), j = 1, \ldots, n.$$

  - Note: Inertia weight is included in the standard PSO.

Particle swarm optimization

- Properties:
  - Best-in-current-swarm vs. best-ever
  - Neighbourhoods
  - Maintaining coherence (velocity restriction!)
  - Inertia

Mattias Wahde, PhD, associate professor, Chalmers University of Technology
e-mail: mattias.wahde@chalmers.se www: www.me.chalmers.se/~mwahde

Particle swarm optimization

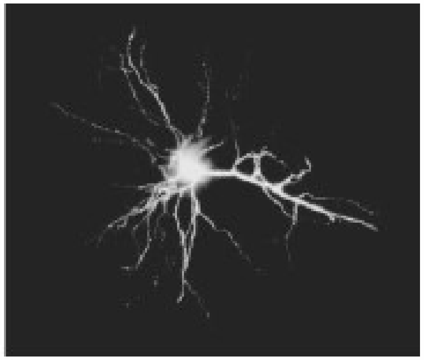
- Discrete versions:
  - Variable truncation
  - Binary PSO
    - Differences between binary PSO and ordinary PSO:

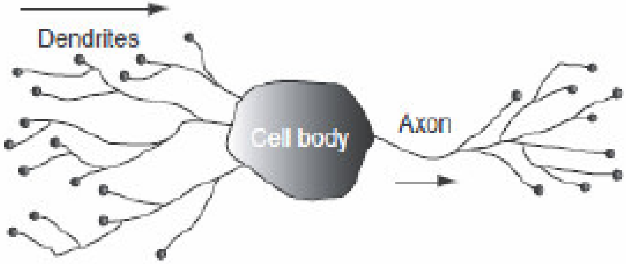
$$\sigma(v_{ij}) = \frac{1}{1 + e^{-v_{ij}}},$$

    - (Interpretation of velocities, positions limited to {0,1} etc.)

Neural networks

- Biological background (Just the basics – inspired by the human brain etc.)

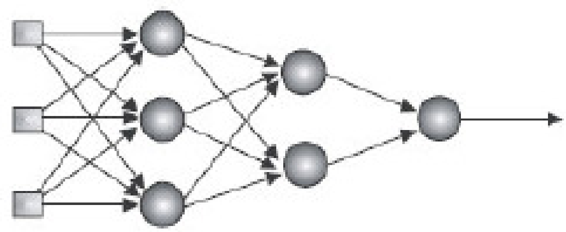




Neural networks

- Computing the output from a feedforward neural network:

$$y = \sigma\left(\sum_{j=0}^{n} w_j x_j\right),$$



Delta rule

- Use gradient descent to find the weights:

$$\Delta w_j = -\eta \frac{\partial E}{\partial w_j} = \eta \sum_{m=1}^{M} \left(o^{[m]} - y^{[m]} \right) x_j^{[m]}$$