Evolutionary Algorithms: Dynamic Optimization Problems

Jiří Kubalík
Department of Cybernetics, CTU Prague

http://cw.felk.cvut.cz/doku.php/courses/a4m33bia/start

Dynamic Optimization Problems

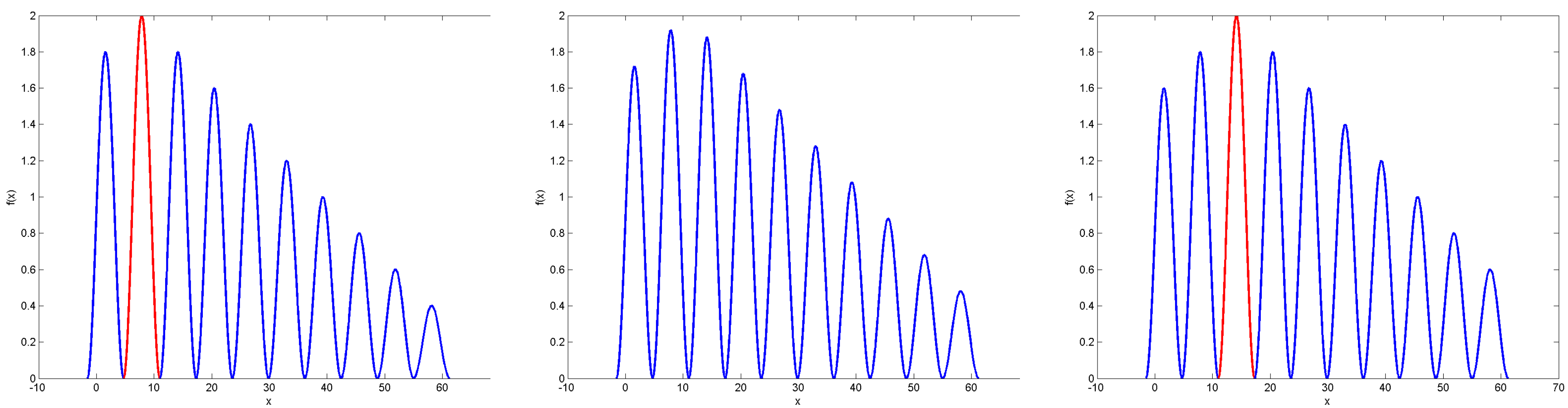
Dynamic Optimization Problems – Many real-world optimization problems are dynamic (scheduling, manufacturing), and optimization methods capable of continuously adapting the solution to a changing environment are needed.

- **Dynamism** – Only small to medium changes of the environment are considered. Continuous adaptation makes sense when landscapes before and after the change are correlated (otherwise restarted search would be an efficient option).
- **Locality** – Even a slight change in the environment might move the optimum to a totally different location; a different peak becomes a maximum peak.

Dynamic Optimization Problems

Dynamic Optimization Problems – Many real-world optimization problems are dynamic (scheduling, manufacturing), and optimization methods capable of continuously adapting the solution to a changing environment are needed.

- **Dynamism** – Only small to medium changes of the environment are considered. Continuous adaptation makes sense when landscapes before and after the change are correlated (otherwise restarted search would be an efficient option).
- **Locality** – Even a slight change in the environment might move the optimum to a totally different location; a different peak becomes a maximum peak.



Evolutionary Algorithms

Evolutionary Algorithm – due to its population-based search seems to be a natural candidate to solve dynamic optimization problems.

Problem with standard EAs is that the population eventually converges to an optimum and loses the diversity, consequently lose their ability to

- efficiently explore the search space, and
- adapt to a change in the environment.

Several ways to deal with the problem

- React on changes. As soon as a change in the environment has been detected, explicit actions are taken to increase diversity of the population to allow the shift to the new optimum.
- Maintaining diversity throughout the run. Convergence is avoided all the time; it is hoped that a spread-out population can adapt to changes more easily.
- Memory-based approaches. EAs supplied with a memory to be able to recall useful information from the past generations.

React on Changes

Restarted search

- The simplest way to deal with changes.
- Inefficient – does not utilize any knowledge acquired during the previous populations.

React on Changes

Restarted search

- The simplest way to deal with changes.
- Inefficient – does not utilize any knowledge acquired during the previous populations.

Hypermutations

- Increases population diversiy by drastically increasing the mutation rate for some number of generations.
- Variable Neighborhood Search – increases mutation gradually after a change in the environment has been detected.

Maintaining Diversity Throughout the Run

Random immigrants

- In every generation, the population is partly replaced by randomly generated individuals.
- It introduces diversity without disrupting the ongoing search process.

Maintaining Diversity Throughout the Run

Random immigrants

- In every generation, the population is partly replaced by randomly generated individuals.
- It introduces diversity without disrupting the ongoing search process.

Genotypic and phenotypic sharing

- Tries to spread out the population over multiple peaks.
- It was shown that sharing remarkably enhances the GA's ability to track optima in slowly changing environments.

Maintaining Diversity Throughout the Run

Random immigrants

- In every generation, the population is partly replaced by randomly generated individuals.
- It introduces diversity without disrupting the ongoing search process.

Genotypic and phenotypic sharing

- Tries to spread out the population over multiple peaks.
- It was shown that sharing remarkably enhances the GA's ability to track optima in slowly changing environments.

Worst among the most similar replacement

- Crowding-like replacement scheme + selection scheme that chooses the second parent with respect to similarity to the first parent.

Memory-based Approaches

EAs supplied with a memory for storing good (partial) solutions that can be reused later as necessary.

Especially beneficial in periodically changing environments.

Implicit memory – redundant representations

- Multiploidy (diploidy), with dominance change mechanism.
- May slow down convergence and favor diversity.
- Performs comparably to a simple haploid GA with a hypermutation.
- Diploid approach is able to learn two solutions and switch between them almost instantly. If more than two targets are used, the approach fails.

Memory-based Approaches

EAs supplied with a memory for storing good (partial) solutions that can be reused later as necessary.

Especially beneficial in periodically changing environments.

Implicit memory – redundant representations

- Multiploidy (diploidy), with dominance change mechanism.
- May slow down convergence and favor diversity.
- Performs comparably to a simple haploid GA with a hypermutation.
- Diploid approach is able to learn two solutions and switch between them almost instantly.

  If more than two targets are used, the approach fails.

Explicit memory – specific information is stored and reintroduced into the population at later generations.

- When the optimum reappears at a previous location, a memory could remember that location and move the population to the new optimum.
- Useful in maintaining diversity.
- Might mislead evolution and prevent it from exploring new regions and discovering new peaks.

Explicit Memory

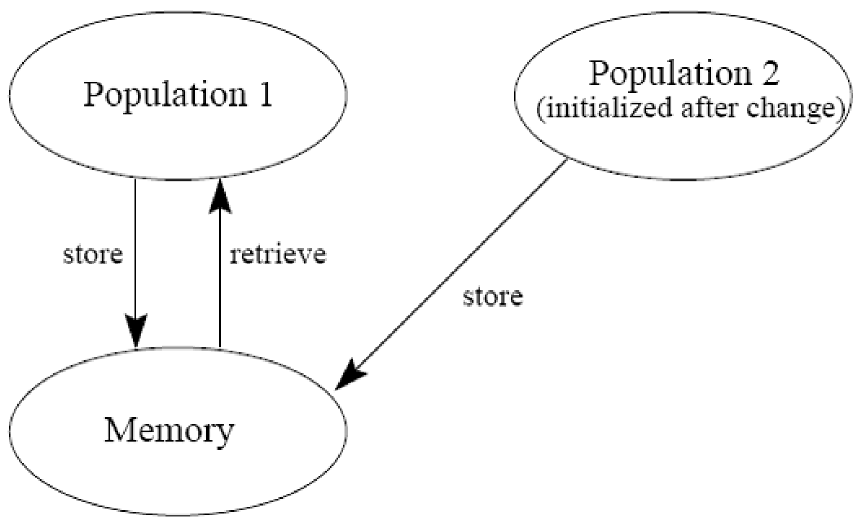
- **Which individuals should be stored in the memory?**
  - Above average fitness.
  - Not too old.
  - Well distributed across several promising areas of the search space.
- **Replacement strategies**
  - Importance value – linear combination of the individual's fitness, age and its contribution to diversity.
  - Delete the individual for which we have the maximum variance in the population. The variance for kth individual is calculated as the sum of variances of the alleles over the remaining individuals in the memory $V(k) = \sum_{i=1}^{m} \sum_{j \in P \setminus \{k\}} (x_{i,j} - \overline{x}_i)^2$
  - Crowding strategy – the new individual replaces the most similar old individual.
    Alternative: Remove the least fit individual out of the two closest old individuals.
- **Which individuals and when should be retrieved from the memory and reinserted into the population?**
  - After the environment changed – when the fitness of at least one individual in the memory has changed.
  - Merge the current population and the memory and keep the best n individuals as new population.

Restarts + Explicit Memory

Memory-based EAS – How to eliminate the risk to misguide the evolution and prevent it from exploring new regions of the search space?

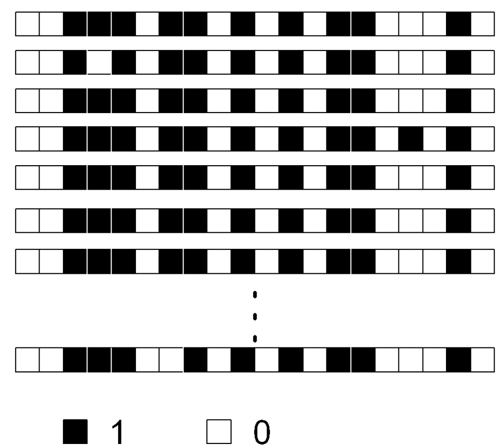
Population divided into

- **Memory-population** – works in the *store-retrieve* mode. Minimum quality and initiating jumps.
- **Search-population** – constantly searching for new peaks, adding these to the memory.
  - Does not retrieve any information from the memory.
  - Re-initialized at random after every change in the environment.



Maintaining Diversity Throughout the Run: GARB

GA with Real-coded Binary Representation



Motivation

- using redundant representation, where many different genotypes map to the same phenotype, would increase the explorative power of the EA and decrease the probability of getting stuck in a local optimum.

Realization

- real-coded (redundant) binary representation.

Effect

- population can not converge to the homogeneous state so that the premature convergence can not take place.

GARB: Representation

Pseudo-binary representation

- binary gene values coded by real numbers from the interval $\langle 0.0, 1.0\rangle$.

$$\begin{aligned} interpretation &= 1 \text{ , for } r \geq 0.5 \\ &= 0 \text{ , for } r < 0.5 \end{aligned}$$

Example: $ch_1 = [0.920.070.230.62]$, $ch_2 = [0.650.190.410.86]$

$$interpretation(ch_1) = interpretation(ch_2) = [1001]$$

Gene strength – gene's stability measure

- The closer the real value is to 0.5 the weaker the gene is.

  "one-valued genes": $0.92 >_S 0.86 >_S 0.65 >_S 0.52$

  "zero-valued genes": $0.07 >_S 0.19 >_S 0.35 >_S 0.49$

  where "$>_S$" is the "*stronger relation*".

GARB: Gene-Strength Adaptation

Vector P stores the frequencies of ones at every position in the current population.

- Example: $P = [0.820.170.350.68]$
  - 82% of ones at the first position,
  - 17% of ones at the first position,
  - 35% of ones at the first position,
  - 68% of ones at the first position,

Every offspring gene is adjusted depending on

- its interpretation,
- the relative frequency of ones at given position in the population.

GARB: Gene-Strength Adaptation cnd.

Adjustment

- Weakening – a gene at the ith position is weakened proportionally to $P[i]$ if it interprets as the binary value that is more frequently sampled at the ith position in the current population.

  The gene's value is shifted towards 0.5
- Strengthening – A gene at the ith position is strengthened proportionally to $P[i]$ if it represents the binary value that is less frequently sampled at the ith position.

  The gene's value is shifted towards the corresponding extreme 0.0 or 1.0.

Realization

- Strengthening binary one genes, weakening binary zero genes (gene at ith position):

$$gene'_i = gene_i + c * (1.0 - P[i])$$

- Strengthening binary zero genes, weakening binary one genes (gene at ith position):

$$gene'_i = gene_i - c * P[i]$$

where c stands for a maximal gene-adaptation step: $c \in (0.0, 0.2)$

GARB: Gene-Strength Adaptation cnd.

Effect

- If some allele begins to prevail in the population,
  1. the corresponding genes are weakened in subsequent generations,
  2. at some point the genes might change their interpretation (shift over the threshold 0.5) and the frequency of the allele decreases.
- Frequency of a given allele is controlled by contradictory pressures
  - the convergence to the optimal solution pressure,
  - the population diversity preservation pressure.

GARB: Boosting-up the Exploitation

Genotype of promising solutions should be stabilized for subsequent generations in order to disable rapid changes in their genotype interpretation.

Promising solutions are newly generated solutions that are better than their parents

- all genes are re-scaled (strengthened) - zero-valued genes are set to be close to 0.0 and one-valued genes are set to be close to 1.0.

  Example:

$$
\begin{aligned}
ch &= (0.71 0.45 0.18 0.57) \\
&\quad\quad\quad\quad\downarrow \\
ch' &= (0.97 0.03 0.02 0.99)
\end{aligned}
$$

Effect – genes survive with unchanged interpretation through more generations.

GARB: Algorithm

Standard generational genetic algorithm + diversity preservation related features

```
begin
initialize(OldPop)
repeat
     calculate P[] from OldPop
     repeat
           select Parents from OldPop
           generate Children
           adjust Children genes
           evaluate Children
           if Child is better than Parents
                 then rescale Child
           insert Children to NewPop
     until NewPop is completed
     switch OldPop and NewPop
until termination condition
end
```

GARB: Moving Optimum Problem

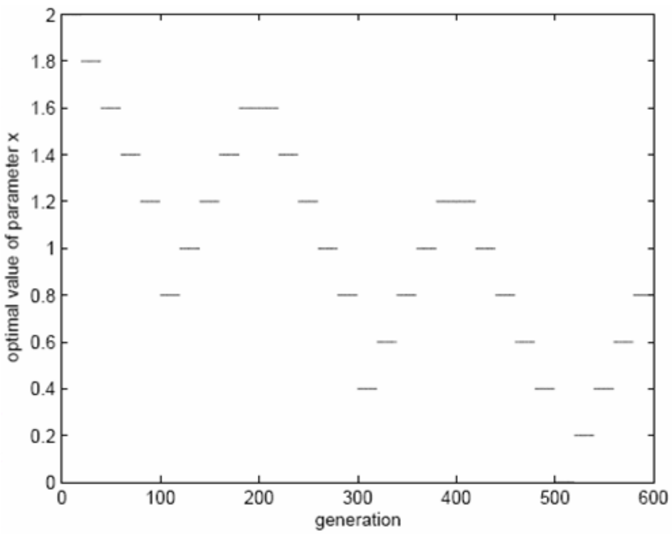
Ošmera's dynamic problem:

$$g_1(x,t) = 1 - e^{-200(x-c(t))^2}$$

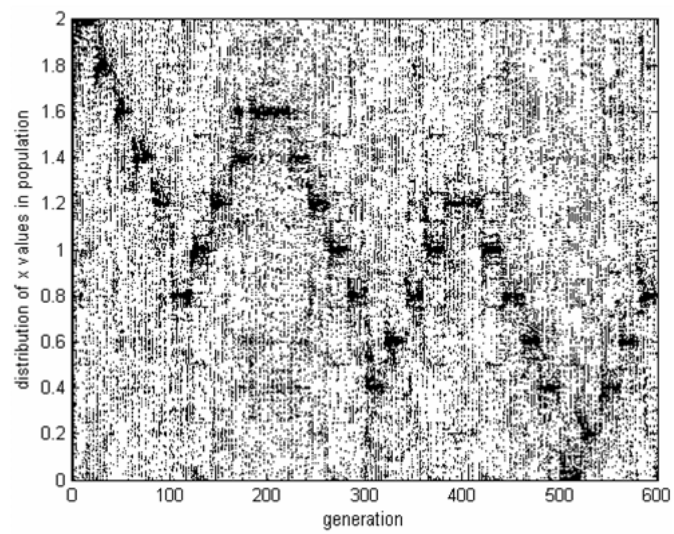
where x is represented by a bit-string of length 31 and normalized to give a value in the range $(0.0, 2.0)$, $t \in \{1, 1000\}$ is the time-step and

$$c(t) = 0.2 \cdot (||\lfloor t/100 \rfloor - 5| + |5 - (\lfloor t/20 \rfloor mod 10)|)$$

specifies the changes in the optimal bit-string.



Moving optimum

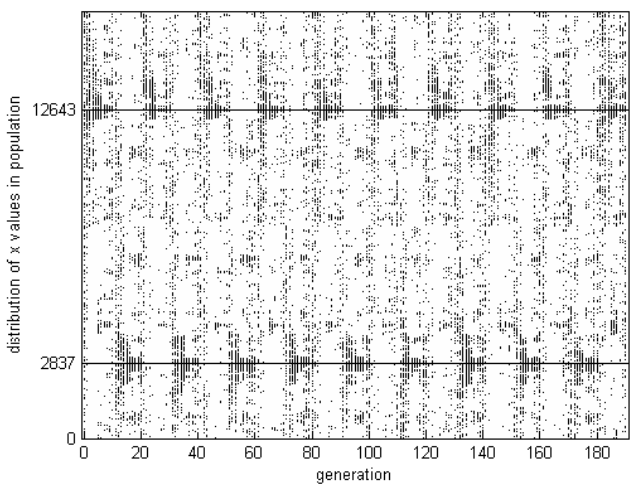


Evolution of the population diversity

GARB: Oscillating Knapsack Problem

Oscillating Knapsack Problem (String matching problem)

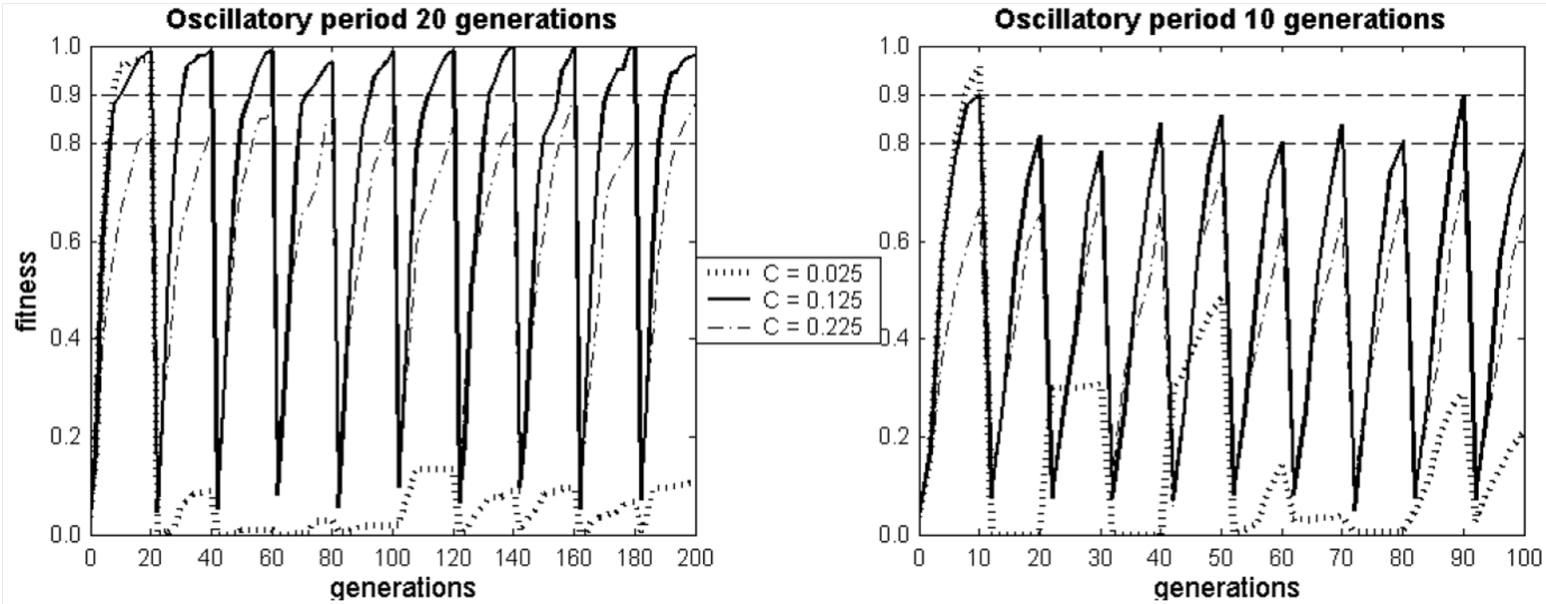
- Goal is to fill a knapsack with objects from an available set of size n, such that the sum of object weights is as close as possible to the target weight t.
- 14 objects, each of them has weight $w_i = 2^i$, where $i = 0 \ldots 13$.
- Optimum fitness is 1.0 and decreases towards 0.0 as the Hamming distance to the optimum solution string increases.
- Target weight oscillates between values 12643 and 2837 (9 bits difference).



Evolution of the population diversity

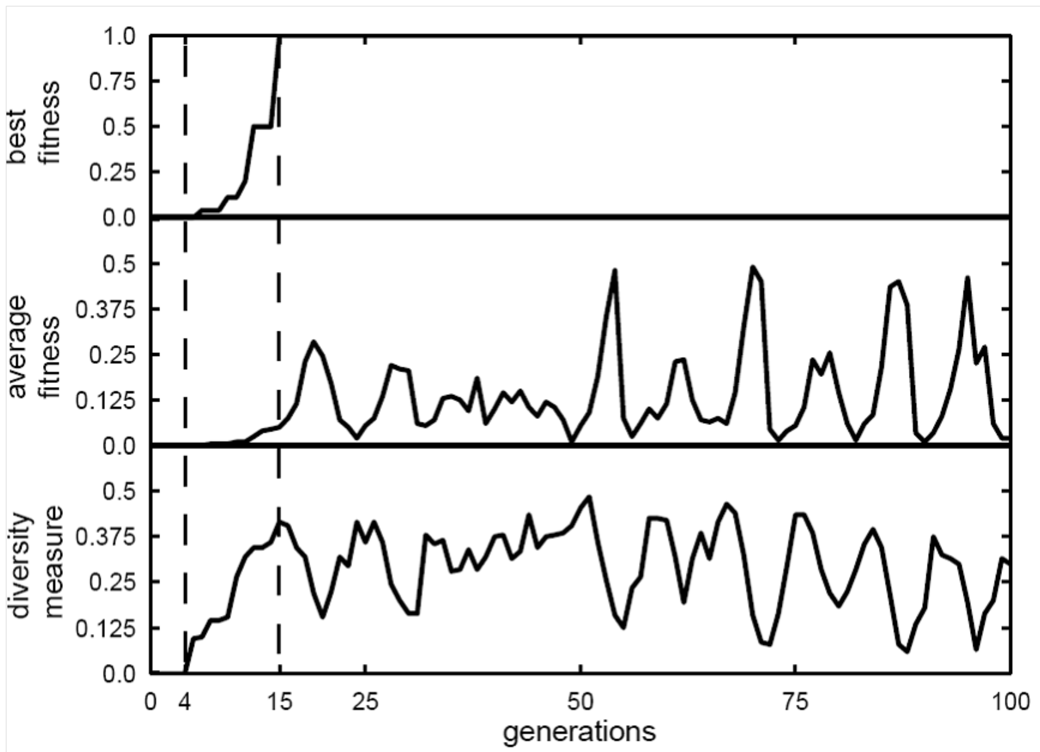
GARB: Oscillating Knapsack Problem cnd.

Average performance of GARB



GARB: Recovery from Completely Homogeneous State

GARB exhibits abilities to recover even from a completely homogeneous state, where the population consists of multiple copies of the same solution (the wrong target in this case).



GARB: Summary

- *Redundant representation* realized by real-valued genes interpreted as binary 0/1 values.
- Gene-strength adjustment plays a role of a *directed self-adaptive mutation* – effect on every gene is directly proportional to the population convergence rate at the given position.
- Strong capabilities to prevent a "premature" convergence that makes it suitable for optimizations in dynamically changing environments.

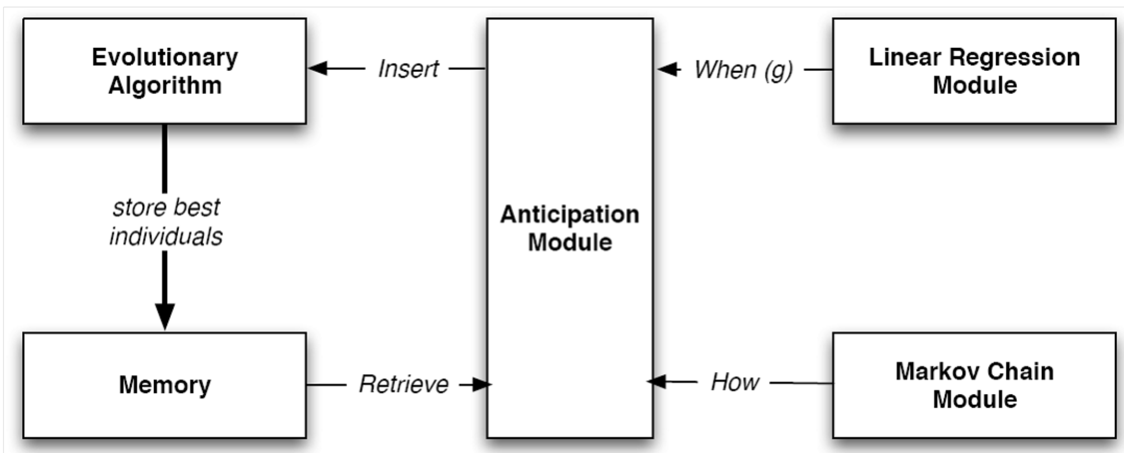
Memory-Based EAs: EA with Prediction Modules

PredEA – standard memory-based EA extended with two prediction modules

- **Linear regression module** (LR) – using information about when previous changes have occured, estimates the generation when the next change will be observed.
- **Markov chain module** (MC) – uses Markov chain to memorize information about different environments observed during the run.

  Provides predictions about to which environment the system will change.

Anticipation module (A) decides when to activate a mechanisms to prepare the main population of the EA to the next change – individuals from the memory that can be useful to the next predicted environment are inserted into the population.



PredEA: Activation of the Anticipation Module

The **activation of the module A must be done at the correct time** in order to prepare the population to the next environment.

- If the predictions of the **LR** are accurate and the correct information is introduced in the population before the change, the EA continues searching readapted to the new environment.
- When the predictions fail, the change is detected when it happens and the EA takes sometime to readapt to the new environment.

  A change is detected if some of the individuals in the memory changes its fitness.

Parameter $\triangle$ is used to decide how many generations before the predicted moment of change the **A** module is activated.

- It is used to to compensate for minor prediction errors associated to the **LR** module.

  A negative prediction error $g < g'$ means that the predicted value for the next change (g) was smaller than the real value (g').

  A positive prediction error $g > g'$ means that the change would be detected when it occurs and no effective adaptation is made.
- If $\triangle$ is greater than the positive error then the **A** module will be activated before the change.
- $\triangle$ should not be set so that the activation of **A** module is close to the change.

PredEA: Mechanisms for Adaptation of $\triangle$

Every time a change happens the prediction error is calculated and the value of $\triangle$ is updated according to the observed errors.

- $\triangle$ is initialized with the value 5,
- minimum value of $\triangle$ is 2 in order to assure that the adaptation to next changestarts at least 2 generations before it happens.

Four adaptation strategies:

1. Maximum prediction error – $\triangle$ is updated to the maximum prediction error observed so far.

$$\begin{aligned} \triangle_1(k) &= 5 && \text{if } k = 1,2 \\ &= max\{2, e_0, e_1, \ldots, e_k\} && \text{if } k > 2 \end{aligned}$$

2. Average of the positive prediction errors – $\triangle$ is updated to the average of the positive prediction errors observed so far.

$$\begin{aligned} \triangle_2(k) &= 5 && \text{if } k = 1,2 \\ &= max\{2, \frac{\sum_{i=1}^{k} e_i}{k}\} && \text{if } k > 2 \text{ and } e_i > 0 \end{aligned}$$

PredEA: Mechanisms for Adaptation of $\triangle$ cnd.

3. Average of the absolute values of the prediction errors – $\triangle$ is updated to the average of the absolute values of all prediction errors observed so far.

$$\begin{aligned}\triangle_3(k) &= 5 && \text{if } k = 1, 2 \\ &= max\{2, \tfrac{\sum_{i=1}^{k} |e_i|}{k}\} && \text{if } k > 2\end{aligned}$$

4. Average of the maximum prediction error and the average of the positive prediction errors observed so far.

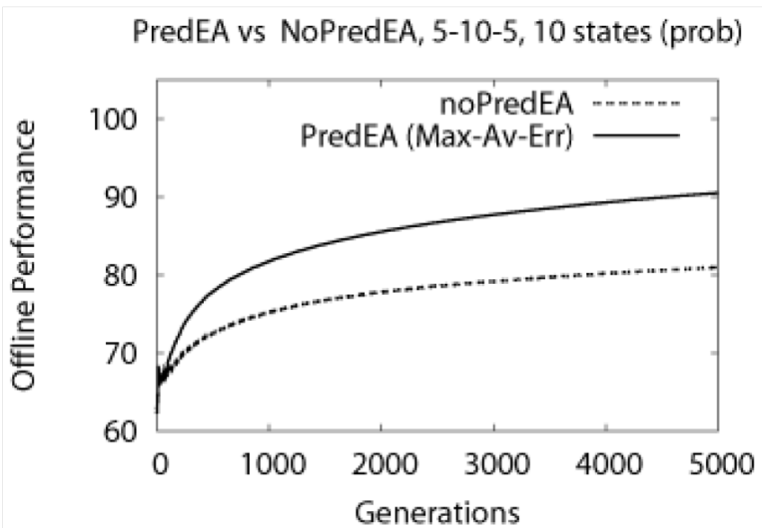
$$\begin{aligned}\triangle_4(k) &= 5 && \text{if } k = 1, 2 \\ &= max\{2, \tfrac{\triangle_1(k) + \triangle_2(k)}{2}\} && \text{if } k > 2 \text{ and } e_i > 0\end{aligned}$$

PredEA: Experimental Results

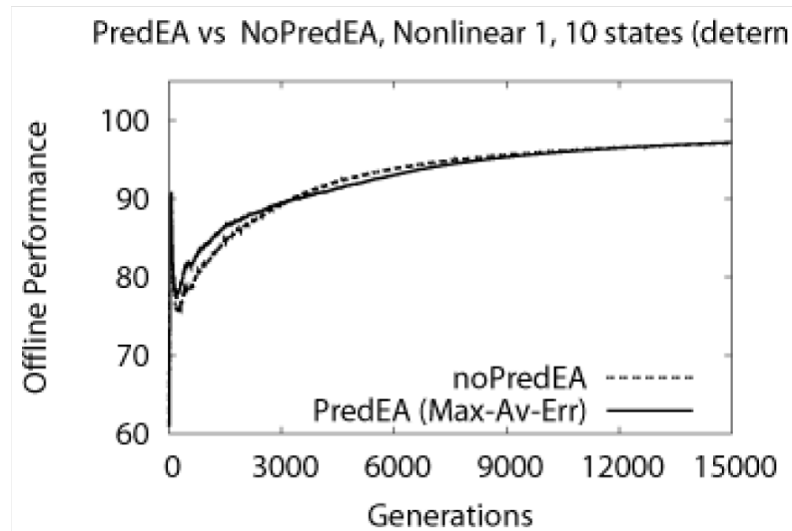
Experimental setup

- dynamic bitmatching problem – 100-bit long templates are generated for each environment.
- change period is generated through (1) the repetition of the 5-10-5 pattern and (2) the nonlinear pattern.

Comparison of the EA with and without prediction modules



Linear change period pattern



Nonlinear change period pattern

PredEA: Summary

- The prediction mechanism significantly improves the performance when the change periods follow a linear or a close to the linear trend.

  When the change period follows a nonlinear pattern, the linear predictor has a poor efficacy.

- Dynamic adjustment of the value of $\triangle$ is superior to using a constant value.

  No preliminary experimentation is needed to find the value. The optimal value is found on the fly.

Reading

- Simoes A. and Costa E.: Evolutionary Algorithms for Dynamic Environments: Prediction Using Linear Regression and Markov Chains , 2008

  http://www.springerlink.com/content/b300281m7958h712/

- Simoes A. and Costa E.: Improving prediction in evolutionary algorithms for dynamic environments, 2009

  http://portal.acm.org/citation.cfm?id=1570021

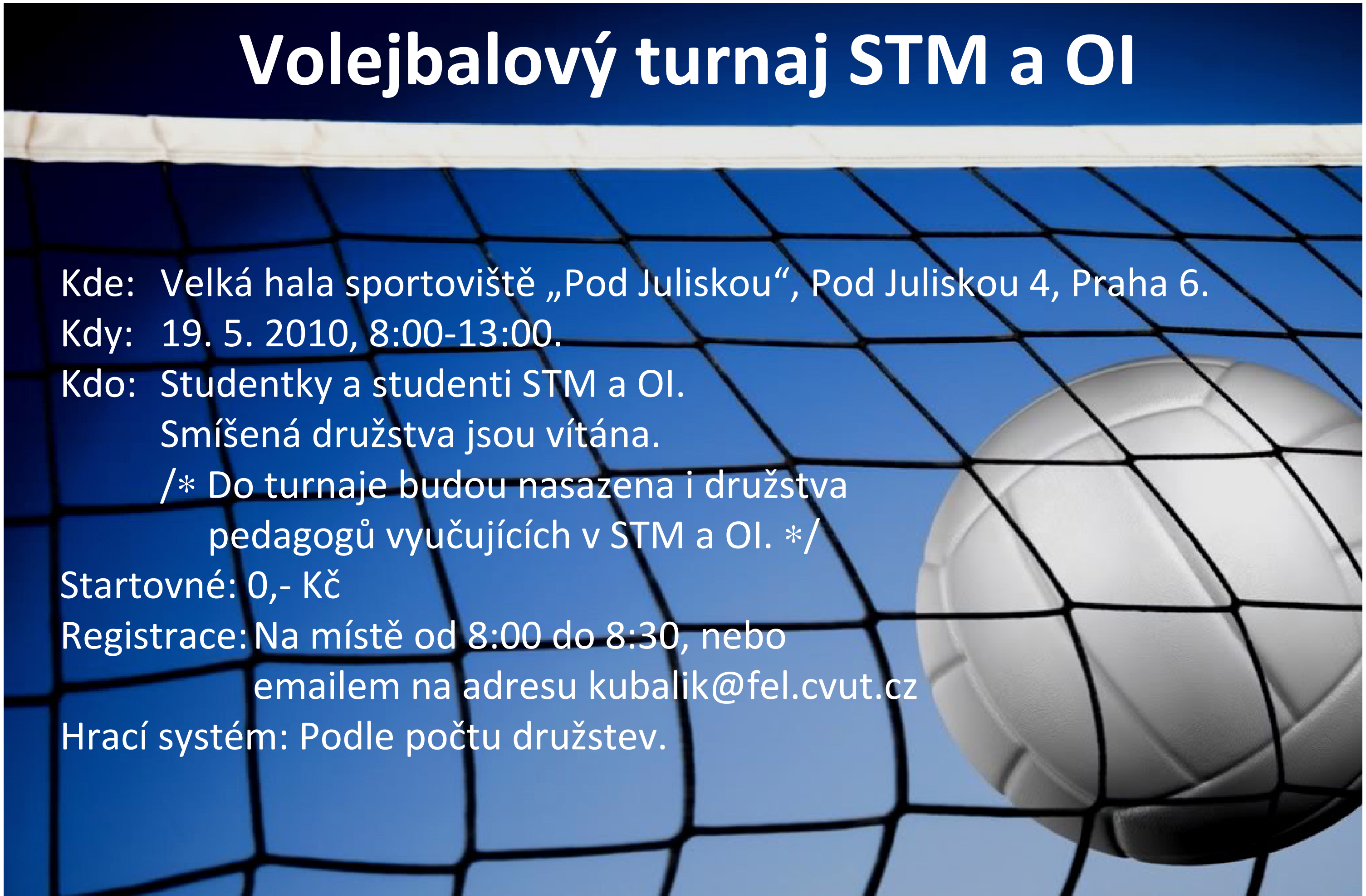
- Branke J.: Memory Enhanced Evolutionary Algorithms for Changing Optimization Problems, 1999

  http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.2.8897

- Branke J.: Evolutionary Approaches to Dynamic Optimization Problems - A Survey, 2001

  http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.10.4619

Volejbalový turnaj STM a OI



Příjďte si vyvětrat hlavu před zkouškami a změřit síly se svými kolegy a kantory!