# A First Step towards the Runtime Analysis of Evolutionary Algorithm Adjusted with Reinforcement Learning

Maxim Buzdalov    Arina Buzdalova

National Research University of IT, Mechanics and Optics
Saint-Petersburg, Russia

13 September 2013
Workshop on Theory of Randomised Search Heuristic
(ThRaSH 2013)



Introduction
- Area Overview
- EA + RL Method
- Application Example

Analyzed Algorithm
- Modified OneMax Problem
- RMHC Algorithm
- Q-learning with Greedy Strategy
- RMHC + Q-learning Algorithm
- Theorem Formulation

Proof
- Learning Lemma
- Markov Chain
- Linear Markov Chain
- Complexity Estimation

Conclusion

# Outline

Introduction

- Area Overview
- EA + RL Method
- Application Example

Analyzed Algorithm

- Modified OneMax Problem
- RMHC Algorithm
- Q-learning with Greedy Strategy
- RMHC + Q-learning Algorithm
- Theorem Formulation

Proof

- Learning Lemma
- Markov Chain
- Linear Markov Chain
- Complexity Estimation

Conclusion

Introduction
- Area Overview
- EA + RL Method
- Application Example

Analyzed Algorithm
- Modified OneMax Problem
- RMHC Algorithm
- Q-learning with Greedy Strategy
- RMHC + Q-learning Algorithm
- Theorem Formulation

Proof
- Learning Lemma
- Markov Chain
- Linear Markov Chain
- Complexity Estimation

Conclusion



Introduction
Area Overview
EA + RL Method
Application Example
Analyzed Algorithm
Modified OneMax Problem
RMHC Algorithm
Q-learning with Greedy Strategy
RMHC + Q-learning Algorithm
Theorem Formulation
Proof
Learning Lemma
Markov Chain
Linear Markov Chain
Complexity Estimation
Conclusion

# Adjusting Evolutionary Algorithms with Reinforcement Learning

- EA + RL: select fitness function for each EA population
- Other methods
  - select evolutionary operator (mutation, crossover, ...)
  - adjust real valued parameters (mutation rate, ...)
- No runtime analysis, just empirical results



Introduction
Area Overview
EA + RL Method
Application Example
Analyzed Algorithm
Modified OneMax Problem
RMHC Algorithm
Q-learning with Greedy Strategy
RMHC + Q-learning Algorithm
Theorem Formulation
Proof
Learning Lemma
Markov Chain
Linear Markov Chain
Complexity Estimation
Conclusion

# Reinforcement Learning

- The agent applies some actions $a \in A$ to the environment
- After each action the agent receives from the environment:
  - some representation of the current state $s \in S$
  - some numeric reward $R(s, a), R : S \times A \to \mathbb{R}$
- Goal: maximize the total amount of reward:

$$E[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)] \to \max$$

M. Buzdalov, A. Buzdalova

Introduction
Area Overview
EA + RL Method
Application Example

Analyzed Algorithm
Modified OneMax Problem
RMHC Algorithm
Q-learning with Greedy Strategy
RMHC + Q-learning Algorithm
Theorem Formulation

Proof
Learning Lemma
Markov Chain
Linear Markov Chain
Complexity Estimation

Conclusion

# Auxiliary Fitness Functions



Introduction
Area Overview
EA + RL Method
Application Example
Analyzed Algorithm
Modified OneMax Problem
RMHC Algorithm
Q-learning with Greedy Strategy
RMHC + Q-learning Algorithm
Theorem Formulation
Proof
Learning Lemma
Markov Chain
Linear Markov Chain
Complexity Estimation
Conclusion

- The problem: maximize target fitness function g:

$$g(x) \to \max_{x \in X}$$

- The set of auxiliary fitness functions is given:

$$H = \{h_i(x)\}$$

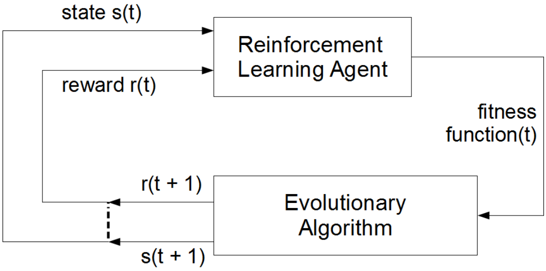
  No prior knowledge about $h_i$ properties

- The goal: adjust evolutionary algorithm using $\{h_i\}$, i. e. decrease number of populations needed to find solution

# EA + RL Method

Control EA with reinforcement learning:

- agent chooses fitness function from $\{h_i\} \cup g$
- next generation is created,
  reward and state are returned




M. Buzdalov,
A. Buzdalova

Introduction
Area Overview
EA + RL Method
Application Example
Analyzed Algorithm
Modified OneMax Problem
RMHC Algorithm
Q-learning with Greedy Strategy
RMHC + Q-learning Algorithm
Theorem Formulation
Proof
Learning Lemma
Markov Chain
Linear Markov Chain
Complexity Estimation
Conclusion

# Application Example

- Generation of tests against solutions of programming challenge tasks
- Success: test which makes the solution exceed time limit
- Fitness functions:
  - Target: running time of the solution (T)
  - Aux: counters in the solution code (Q, I, L), that correlate with T

| Algorithm | FFs | Success, % | Populations | |
|---|---|---|---|---|
| | | | Mean | $\sigma$ |
| GA | Q | 95 | 3815 | 3466 |
| GA + RL | all | 80 | 5817 | 6160 |
| GA | I | 54 | 12669 | 12873 |
| GA | L | 51 | 13755 | 14082 |
| GA | T | 0 | – | – |

M. Buzdalov, A. Buzdalova
Introduction
Area Overview
EA + RL Method
Application Example
Analyzed Algorithm
Modified OneMax Problem
RMHC Algorithm
Q-learning with Greedy Strategy
RMHC + Q-learning Algorithm
Theorem Formulation
Proof
Learning Lemma
Markov Chain
Linear Markov Chain
Complexity Estimation
Conclusion

# Requirements

| Auxiliary set type | Requirement |
|---|---|
| efficient only | method $>$ EA |
| efficient and inefficient | method $>$ EA |
| inefficient only | method $=$ EA |
| dynamically changes | method $\geq$ EA |

Notation:

- efficient fitness function $\Rightarrow$ target increases more rapidly
- inefficient – the rest
- "=" asymptotically equals, ">" outperforms

M. Buzdalov,
A. Buzdalova

Introduction
Area Overview
EA + RL Method
Application Example

Analyzed Algorithm
Modified OneMax Problem
RMHC Algorithm
Q-learning with Greedy Strategy
RMHC + Q-learning Algorithm
Theorem Formulation

Proof
Learning Lemma
Markov Chain
Linear Markov Chain
Complexity Estimation

Conclusion

Introduction
Area Overview
EA + RL Method
Application Example

Analyzed Algorithm
Modified OneMax Problem
RMHC Algorithm
Q-learning with Greedy Strategy
RMHC + Q-learning Algorithm
Theorem Formulation

Proof
Learning Lemma
Markov Chain
Linear Markov Chain
Complexity Estimation

Conclusion

# Modified OneMax Problem

M. Buzdalov,
A. Buzdalova

Introduction
Area Overview
EA + RL Method
Application Example
Analyzed Algorithm
Modified OneMax Problem
RMHC Algorithm
Q-learning with Greedy Strategy
RMHC + Q-learning Algorithm
Theorem Formulation
Proof
Learning Lemma
Markov Chain
Linear Markov Chain
Complexity Estimation
Conclusion

- Individual — bit vector of length N
- Target fitness function $f_1$ — number of ones
- Inefficient aux. fitness function $f_0$ — number of zeros
- Reinforcement learning state — number of ones

# Random Mutation Hill Climber

- X — bit vector
- Mutate(X) — inverts one random bit
- f — fitness function

## RMHC algorithm

1. Initialize X: vector of N zeros
2. Repeat until termination condition is not reached
   2.1 Y := Mutate(X)
   2.2 If $(f(Y) \geq f(X))$ X := Y



Introduction
Area Overview
EA + RL Method
Application Example
Analyzed Algorithm
Modified OneMax Problem
RMHC Algorithm
Q-learning with Greedy Strategy
RMHC + Q-learning Algorithm
Theorem Formulation
Proof
Learning Lemma
Markov Chain
Linear Markov Chain
Complexity Estimation
Conclusion

# Q-learning with Greedy Strategy



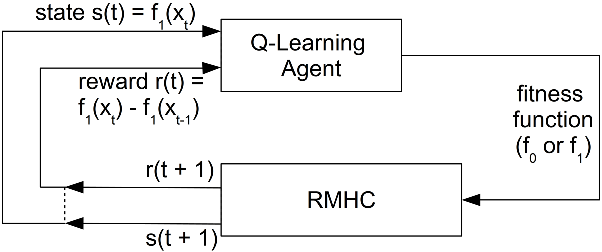
Introduction
Area Overview
EA + RL Method
Application Example
Analyzed Algorithm
Modified OneMax Problem
RMHC Algorithm
Q-learning with Greedy Strategy
RMHC + Q-learning Algorithm
Theorem Formulation
Proof
Learning Lemma
Markov Chain
Linear Markov Chain
Complexity Estimation
Conclusion

- $Q : S \times A \to \mathbb{R}$ — quality of applying action in state
- $\alpha$ — the learning rate
- $\gamma$ — the discount factor

## Q-learning algorithm

1. Initialize $Q(s, a)$: fill it with zeros
2. Repeat until termination condition is not reached
   - 2.1 Select an action: $a := \arg\max_a Q(s_t, a)$
   - 2.2 Apply selected action, get reward $R$
   - 2.3 Update $Q(s_t, a) := (1 - \alpha)Q(s_t, a) + \alpha(R + \gamma \max_{a'} Q(s_{t+1}, a'))$

# RMHC + Q-learning Algorithm



# RMHC + Q-learning Algorithm

M. Buzdalov,
A. Buzdalova

Introduction
Area Overview
EA + RL Method
Application Example
Analyzed Algorithm
Modified OneMax Problem
RMHC Algorithm
Q-learning with Greedy Strategy
RMHC + Q-learning Algorithm
Theorem Formulation
Proof
Learning Lemma
Markov Chain
Linear Markov Chain
Complexity Estimation
Conclusion

$X \leftarrow$ current individual, vector of N zeros
$Q \leftarrow$ transition quality matrix, $N \times 2$, filled with zeros
$f_1 \leftarrow$ target fitness function: number of ones in an individual
$f_0 \leftarrow$ inefficient fitness function: number of zeros in an individual
$\text{Mutate}(X) \leftarrow$ mutation operator: inverts random bit
$\alpha \in (0;1)$, $\gamma \in (0;1)$ — Q-learning parameters
while $f_1(X) < N$ do
    $S \leftarrow f_1(X)$
    $Y \leftarrow \text{Mutate}(X)$
    f, I: chosen fitness function and its index
    if $Q(S,0) > Q(S,1)$ then
        $f \leftarrow F_0$, $I \leftarrow 0$
    else if $Q(S,0) < Q(S,1)$ then
        $f \leftarrow F_1$, $I \leftarrow 1$
    else
        $I \leftarrow \text{random}(0,1)$, $f \leftarrow F_I$
    end if
    if $f(Y) \geq f(X)$ then
        $X \leftarrow Y$
    end if
    $R \leftarrow F_1(X) - S$
    $Q(S,I) \leftarrow (1-\alpha)Q(S,I) + \alpha(\gamma \cdot R + \max_j Q(S,j))$
end while

# Theorem

Random Mutation Hill Climber controlled with Q-learning algorithm with greedy exploration strategy solves modified OneMax problem in $\Theta(N \log N)$ fitness function calls.

M. Buzdalov,
A. Buzdalova

Introduction
Area Overview
EA + RL Method
Application Example

Analyzed Algorithm
Modified OneMax Problem
RMHC Algorithm
Q-learning with Greedy Strategy
RMHC + Q-learning Algorithm
Theorem Formulation

Proof
Learning Lemma
Markov Chain
Linear Markov Chain
Complexity Estimation

Conclusion

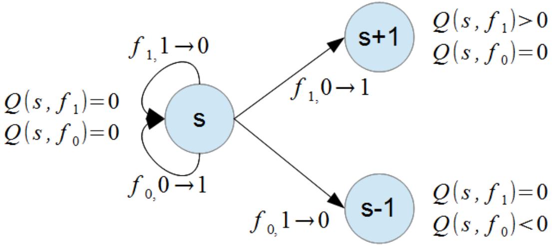
Introduction
Area Overview
EA + RL Method
Application Example
Analyzed Algorithm
Modified OneMax Problem
RMHC Algorithm
Q-learning with Greedy Strategy
RMHC + Q-learning Algorithm
Theorem Formulation
Proof
Learning Lemma
Markov Chain
Linear Markov Chain
Complexity Estimation
Conclusion

# Learning Lemma: Formulation



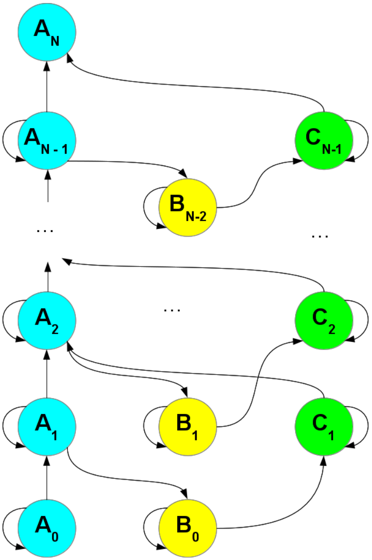
Assume that the Q-learning agent visits a state S and leaves it. Then the optimal fitness function $f_1$ will be chosen in S in all next visits.

# Learning Lemma: Proof

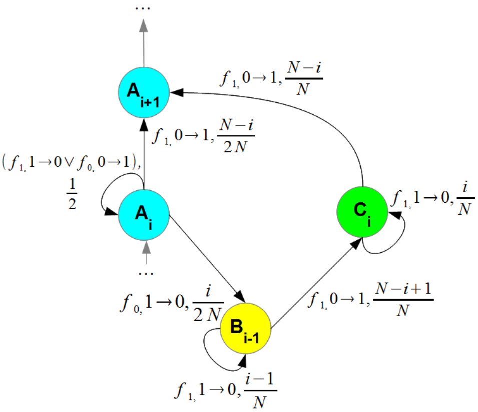



- $\mathrm{Q(S, I) := (1 - \alpha)Q(S, I) + \alpha(\gamma \cdot r + max_j\, Q(S', j))}$
- In both cases, $\mathrm{Q(S, 1) > Q(S, 0)}$
- So $\mathrm{f_1}$ will be chosen in S

# Markov Chain: Overview

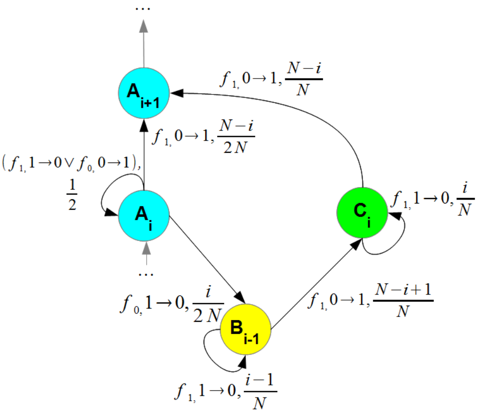


# Markov Chain: Transition Probabilities



# Expectation of Transition Number-1

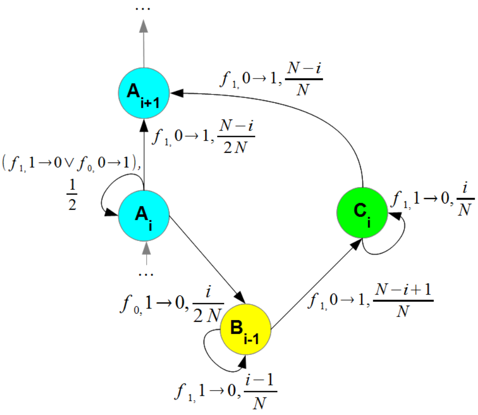





- $\mathrm{E}(\mathrm{B}_{i-1} \to \mathrm{C}_i) = 1 \times \frac{N-i+1}{N} + (1 + \mathrm{E}(\mathrm{B}_{i-1} \to \mathrm{C}_i)) \times \frac{i-1}{N}$
- $\mathrm{E}(\mathrm{B}_{i-1} \to \mathrm{C}_i) = \frac{N}{N-i+1}$

Introduction
Area Overview
EA + RL Method
Application Example
Analyzed Algorithm
Modified OneMax Problem
RMHC Algorithm
Q-learning with Greedy Strategy
RMHC + Q-learning Algorithm
Theorem Formulation
Proof
Learning Lemma
Markov Chain
Linear Markov Chain
Complexity Estimation
Conclusion

# Expectation of Transition Number-2

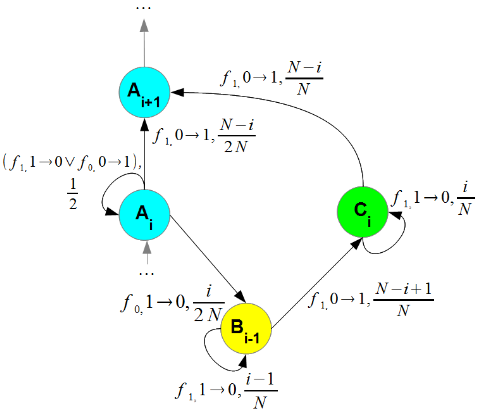



- $\mathrm{E}(\mathrm{C_i} \to \mathrm{A_{i+1}}) = 1 \times \frac{\mathrm{N-i}}{\mathrm{N}} + (1 + \mathrm{E}(\mathrm{C_i} \to \mathrm{A_{i+1}})) \times \frac{\mathrm{i}}{\mathrm{N}}$
- $\mathrm{E}(\mathrm{C_i} \to \mathrm{A_{i+1}}) = \frac{\mathrm{N}}{\mathrm{N-i}}$

M. Buzdalov,
A. Buzdalova

Introduction
Area Overview
EA + RL Method
Application Example
Analyzed Algorithm
Modified OneMax Problem
RMHC Algorithm
Q-learning with Greedy Strategy
RMHC + Q-learning Algorithm
Theorem Formulation
Proof
Learning Lemma
Markov Chain
Linear Markov Chain
Complexity Estimation
Conclusion

# Expectation of Transition Number-3

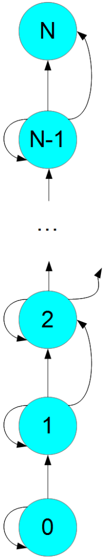



- $\mathrm{Len}(A_i \to A_{i+1}) = 1 + E(B_{i-1} \to C_i) + E(C_i \to A_{i+1})$

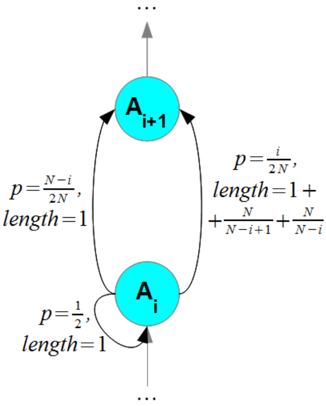



Introduction
Area Overview
EA + RL Method
Application Example
Analyzed Algorithm
Modified OneMax Problem
RMHC Algorithm
Q-learning with Greedy Strategy
RMHC + Q-learning Algorithm
Theorem Formulation
Proof
Learning Lemma
Markov Chain
Linear Markov Chain
Complexity Estimation
Conclusion

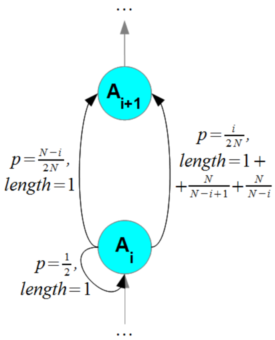
# Linear Markov Chain: Overview



# Linear Markov Chain: Lengths and Probabilities



# Complexity Estimation-1




- $Z(i) = (1 + Z(i)) \times \frac{1}{2} + \frac{N-i}{2N} + (1 + \frac{N}{N-i+1} + \frac{N}{N-i}) \times \frac{i}{2N}$
- $Z(i) = 2 + \frac{i}{N-i+1} + \frac{i}{N-i}$
- For $Z(0)$ this also holds
- $T_R(N) = \sum_{i=0}^{N-1} (2 + \frac{i}{N-i+1} + \frac{i}{N-i})$


M. Buzdalov, A. Buzdalova

Introduction
Area Overview
EA + RL Method
Application Example

Analyzed Algorithm
Modified OneMax Problem
RMHC Algorithm
Q-learning with Greedy Strategy
RMHC + Q-learning Algorithm
Theorem Formulation

Proof
Learning Lemma
Markov Chain
Linear Markov Chain
Complexity Estimation

Conclusion

# Comparison with RMHC without Inefficient Fitness Function

- RMHC without inefficient fitness function
  - $T_0(N) = \sum_{i=0}^{N-1} (1 + \frac{i}{N-i})$
  - $T_0(N) = \Theta(N \log N)$
- RMHC + Q-learning:
  - $T_R(N) = \sum_{i=0}^{N-1} (2 + \frac{i}{N-i+1} + \frac{i}{N-i})$
  - $1 + \frac{i}{N-i} < 2 + \frac{i}{N-i+1} + \frac{i}{N-i} < 2 + 2\frac{i}{N-i} = 2(1 + \frac{i}{N-i})$
- $T_0(N) < T_R(N) < 2 \cdot T_0(N)$
- $T_R(N) = \Theta(N \log N)$

Introduction
Area Overview
EA + RL Method
Application Example
Analyzed Algorithm
Modified OneMax Problem
RMHC Algorithm
Q-learning with Greedy Strategy
RMHC + Q-learning Algorithm
Theorem Formulation
Proof
Learning Lemma
Markov Chain
Linear Markov Chain
Complexity Estimation

# Conclusion



Introduction
Area Overview
EA + RL Method
Application Example
Analyzed Algorithm
Modified OneMax Problem
RMHC Algorithm
Q-learning with Greedy Strategy
RMHC + Q-learning Algorithm
Theorem Formulation
Proof
Learning Lemma
Markov Chain
Linear Markov Chain
Complexity Estimation
Conclusion

# Conclusion

- EA + RL is proved to ignore an inefficient fitness function in a model problem
- Future work
  - Generalize the obtained result
  - EA + RL with an efficient fitness fucntion outperforms EA

# Thank you! Any questions?



Introduction
Area Overview
EA + RL Method
Application Example
Analyzed Algorithm
Modified OneMax Problem
RMHC Algorithm
Q-learning with Greedy Strategy
RMHC + Q-learning Algorithm
Theorem Formulation
Proof
Learning Lemma
Markov Chain
Linear Markov Chain
Complexity Estimation
Conclusion