Artificial Immune Systems for Optimisation

Thomas Jansen
Aberystwyth University
Aberystwyth SY23 3DB, UK
t.jansen@aber.ac.uk

Christine Zarges
University of Birmingham
Birmingham B15 2TT, UK
c.zarges@cs.bham.ac.uk

Copyright is held by the author(s).
ACM 978-1-4503-1964-5/13/07.

Introduction
Natural Immune System
AIS as Classifiers
AIS as Optimisers
Analysing AIS
Conclusion & References

Artificial Immune Systems are . . .

• a model of the natural immune system
  if you are interested in the natural immune system

• computational systems inspired by the natural immune system
  with natural applications in anomaly detection & classification
  if you are interested in solving a classification problem

• nature-inspired algorithms using the natural immune system
  as metaphor for problem-solving
  if you are interested in solving difficult problems

• nature-inspired randomised search heuristics
  like many others, e.g., evolutionary algorithms, ACO, SA, . . .
  if you are interested in randomised search heuristics

• a fascinating area of research
  in any case

Good News We cover all these aspects. (⇝ structure governed by this)

Plans for Today

1 Introduction
   • AIS are . . .
   • Overview

2 AIS as Model of the Natural Immune System

3 AIS as Classifiers

4 AIS as Optimisers

5 Analysing AIS
   • Analysing Operators and Meta-Dynamics
   • Analysing Complete AIS

6 Summary and Conclusions
   • AIS Tutorial Summary
   • AIS as Future Research Topics

Biological Inspiration: The Immune System of Vertebrates

“The immune system recognizes infection and induces protective responses.” [30]

Main Tasks

• Immunological recognition
• Immune effector functions
• Immune regulation
• Immunological memory
**Introduction**

**Natural Immune System**

**AIS as Classifiers**

**AIS as Optimisers**

**Analysing AIS**

**Conclusion & References**

---

**Multilayer Structure of the Immune System**

- **Pathogens**
- **Skin**
- **Biochemical Barriers**
- **Innate Immune Response**
- **Adaptive Immune Response**

(Reproduced from [6, 35])

---

**Innate vs. Adaptive Immunity**

**Innate Immunity**

- Non-specific response against large number of bacteria
- First line of defense: Controls infection before adaptive immune response kicks in
- Initiates and controls adaptive immune response
- Dendritic cells form bridge between innate/adaptive immunity

**Adaptive Immunity**

- Specific and preventive immune response
- Mediated by lymphocytes in the lymph nodes
- Two main types: B cells and T cells
- Develops immunological memory
- Described by the Clonal Selection Theory

---

**Adaptive Immunity – Immunological Recognition**

- Naïve lymphocytes: not yet involved in an immune response
- Carry antigen receptors of single specificity
- Receptor diversity due to
  - Random recombination of gene fragments from several libraries
  - Somatic hypermutation to increase antigen-antibody affinity
- Become active due to interaction with antigenic stimulus
- Recognition based on complementarity between binding region of receptor and epitope of antigen on molecular level
- Antigens may present several epitopes
- Require co-stimulatory signals
- B cell receptor interacts directly while T cell receptor requires preprocessing and presenting by other cells

---

**Clonal Selection Theory (1) [2]**

- Describes basic properties of adaptive immune response
- Only cells recognizing an antigen proliferate and differentiate into effector cells
- B cells:
  - Subject to somatic hypermutations
  - B effector cells secrete antibodies
- T cells:
  - Not subject to mutation
  - T effector cells secrete lymphokine
- B cell clonal selection similar to natural selection
- Learning through increasing population size and affinity
- Immune repertoire evolved from a random base to reflection of actual antigenetic environment
Introduction

Natural Immune System

AIS as Classifiers

AIS as Optimisers

Analysing AIS

Conclusion & References

Clonal Selection Theory (2) [2]

Clonal Selection Theory (3) [2]

• Diversity via somatic hypermutations, receptor editing and newcomer cells
• Non-functional and harmful anti-self specificities are eliminated
• Variants with higher affinity dominate immune response and enter immune memory
• Some low affinity cells enter repertoire to maintain diversity
• Hypermutations
  • Point mutations allow for exploring local regions
  • On average one mutation per cell division
  • Short burst of somatic hypermutation followed by a pause to allow for selection and clonal expansion
  • Regulation of the hypermutation process by selection depending on receptor affinity
• Receptor editing
  • Instead of clonal deletion development of new receptors
  • Allows for larger steps through the landscape

Immune Network Theory [27]

• Remember Clonal Selection Theory:
  Immune system \( \triangleq \) set of discrete cells and molecules originally at rest; triggered only by a foreign antigenic stimulation
• Now: different perspective Immune Network Theory
  Immune system \( \triangleq \) regulated network of cells and molecules that recognize one another even in the absence of antigens
• Network is autonomous, self-regulated and aims at maintaining a specific range of activity
• Immune tolerance, learning and memory as inherent global properties

Clonal Selection Theory

Immune Network Theory
Danger Theory [29]

Idea  Immune system rather detects danger than nonself

Self-Nonself Discrimination

Consider  crude, unrealistic, partly misleading example

Artificial Immune Systems as Classifiers

Remember  artificial immune systems
inspired by natural immune system
→ perform self-nonself discrimination
and react accordingly most natural application

Observation  self-nonself discrimination
≡ two-class classification problem

Fact  many different AIS for this task
based on different immune principles

Today  consider three examples
• negative selection (inspired by self-nonself discrimination)
• receptor density algorithm (inspired by T cell signalling)
• dendritic cell algorithm (inspired by the danger theory)

Simple Negative Selection Algorithm

Problem  formal formulation
Input  finite alphabet $\Sigma$, string space $\Sigma^l$, training set $S \subseteq \Sigma^l$ of self strings
Output  set of detectors $D$ that match only self by means of a partial match of length $r$

Algorithm  works in two phases (outline)
Learning  randomly generate detectors
keep those that do not match any $s \in S$
Detection  mark everything that matches some $d \in D$ as nonself
(early algorithm, see e.g., Forrest et al. (1994) [12])

Fact  very inefficient (for different types of detectors)
(see e.g., Stibor et al. (2004) [34])
Efficient Negative Selection

**Algorithm outline, main ingredients**
- use prefix trees as main data structure
- efficiently build finite automaton to represent detectors
  - (note: no explicit detector set)
- construction of automaton works in time $O(|S| \cdot l \cdot r)$
- classification works in time $O(l)$

(for details see Elberfeld, Textor (2011) [11])
(for implementation see http://bioinformatics.bio.uu.nl/textor/negativeselection.html)

**Lesson Learned**
- immune metaphor useful for ideas
- algorithmic implementation following the IS may be very far from optimal
- immune-ideas can be implemented efficiently using ‘classical’ algorithmic ideas

More Modern AIS-Approaches to Classification

**Remember**
- self-nonself discrimination
  - based on a simplistic understanding of the immune system
  - can be implemented efficiently using clever algorithms/data structures

**Fact**
- many more AIS-approaches to classification exist
  - based on different aspects of immunology
  - too many to cover all here

**Today**
- two current approaches
  - based on different immunological theories
  - current $\cong$ both considered and further developed in current publications

1. based on T cell signalling: receptor density algorithm
2. based on danger theory: dendritic cell algorithm

Receptor Density Algorithm

**Motivation**
- two-class classification performed by T cells depending on history

**Basis**
- model of T cell receptor called receptor
  - having a state $c$, position $p$, negative feedback $n$, a negative feedback barrier $\beta$, length $l > \beta$
  - reacts to input $u$ by updating the position (adding $u$, subtracting $n$), increasing negative feedback for positions above $\beta$, decaying negative feedback otherwise
  - combining receptors spatially in form of a grid
  - with a stimulation kernel function
  - yields receptor density algorithm
  - (for details see Owens (2010) [32], Owens et al. (2013) [33])

Dendritic Cell Algorithm

**Motivation**
- danger theory specifically dendritic cells model immune systems responds to danger/safe signals (does not perform self-nonself discrimination)

**Basis**
- model of dendritic cells being either immature, semi-mature or mature, having a lifespan
  - processing input classified as either danger, safe or PAMP, computes two values:
  - DCM, indicating the amount of processed information, $K$, indicating the classification as normal or anormal
  - a collection of such cells (with different lifespans) forms dendritic cell algorithm performing classification
  - (for details see Greensmith (2007) [13])
  - fully formalised, simplified deterministic version available

(for details see Gu (2011) [14], Gu et al. (2013) [15])
Artificial Immune Systems as Optimisers

Remember
- most natural application
  \( \equiv \) classification, pattern recognition, IT security

Observation
- some algorithms also suitable for optimisation tasks

In particular
- clonal selection and immune network theory

Consider
- minimisation/maximisation of
  \( f : \{0, 1\}^n \rightarrow \mathbb{R} \) or
  \( f : \mathbb{R}^n \rightarrow \mathbb{R} \)

Observation
- structure similar to evolutionary algorithms
- concrete implementation very different

Today
- characteristics and concrete algorithms
  - Mutation and Metadynamics in AIS
  - Clonal Selection Algorithms: CLONALG, B-Cell Algorithm, opt-IA
  - Immune Network: opt-aiNet

Additionally
- simple Java applet to ‘play along’
  http://www.cs.bham.ac.uk/~zargesc/ais-tutorial/

Mutation in Artificial Immune Systems (1)

Usually
- Mutations at high rate \( \Rightarrow \) Hypermutations

Inverse Fitness: “Proportional” Hypermutation
- Idea
  Apply mutations with lower mutation rate to good search points
- Usually
  - Normalised fitness value \( \hat{f} \in [0, 1] \) used
    - Using optimal function value \( \hat{f}_{opt} \):
      \( \hat{f}(x) = f(x) / f_{opt} \)
    - Using best known function value \( \hat{f}_{best} \):
      \( \hat{f}(x) = f(x) / f_{best} \)
- Examples
  - CLONALG: \( p_m = \exp(-\rho \cdot \hat{f}) \)
  - opt-aiNet: \( p_m = (1/\rho) \cdot \exp(-\hat{f}) \)
- Remark
  In continuous optimisation \( p_m \) equals the mutation strength

Mutation in Artificial Immune Systems (2)

Contiguous Hypermutations
- Idea
  Perform mutations with probability \( r \in (0, 1] \) only in contiguous region.

1. Choose hotspot \( p \in \{0, \ldots, n-1\} \), length \( \ell \in \{0, 1, \ldots, n\} \).
   \( p = 8, \ell = 4 \)
   \[
   \begin{array}{cccccccc}
   0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
   \end{array}
   \]
   not wrapping around

2. Choose two hotspots \( a, b \in \{0, \ldots, n-1\} \).
   \( a = 8, b = 5 \)
   \[
   \begin{array}{cccccccc}
   0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
   \end{array}
   \]
   wrapping around

Mutation in Artificial Immune Systems (3)

Hypermutation with Mutation Potential
- Idea
  Determine number of local mutation steps during a single hypermutation.
- Different classes: static, inversely proportional, proportional
- Example
  - for some constant \( c \in ]0, 1[ \), minimisation
    \( M_c(v) = \left(1 - \frac{f_{best}}{v}\right) \cdot c \cdot n \)
  - Variants for the hypermutation of \( x \in \{0, 1\}^n \):
    \( \text{tabu, stop at first constructive mutation} \)

1. Set \( y := x \), Set \( v := f(x) \).
2. Repeat the following \( M_c(v) \) times:
   - If \( \text{tabu} = 0 \) select \( i \in \{1, \ldots, n\} \) uniformly at random
     else select \( i \in \{1, \ldots, n\} \) uniformly at random, \( i \) not previously chosen.
   - \( y[i] := 1 - y[i] \)
   - If \( f_c m = 1 \) and \( f(y) < f(x) \) Then break
Metadynamics in Artificial Immune Systems

Sometimes worst search points replaced by new random ones.

Popular mechanism: Ageing

- Idea: Increase diversity by removing old and non-improving points
- General implementation:
  - Assign age to each search point
  - Increase age in each round
  - If new search point improves over parent
    Then assign age 0 to new search point
    Else inherit age of parent
- Optional: Fill up population with new random search points
- Variants for some parameter $\tau_{\text{max}}$
  - Static Pure Ageing remove search points older than $\tau_{\text{max}}$
  - Stochastic Ageing remove each search point with probability $1 - 2^{-1/\tau_{\text{max}}}$

The B-Cell Algorithm (BCA) [28]

Parameters $\mu$: population size $\lambda$: offspring population size

1. Initialisation Create an initial population $P = \{x_1, x_2, \ldots, x_\mu\}$.
2. Clonal Selection and Expansion For all $i \in \{1, 2, \ldots, \mu\}$:
   a) Create $\lambda$ clones of $x_i$ and place them in a clonal pool $C_i = \{y_i^1, \ldots, y_i^\lambda\}$.
   b) Select $j \in \{1, \ldots, \lambda\}$ uniformly at random: Flip each bit of $y_i^j$ with probability $p_m$.
   c) For all $j \in \{1, \ldots, \lambda\}$: Apply somatic contiguous hypermutation to $y_i^j$.
3. Selection for Replacement For all $i \in \{1, 2, \ldots, \mu\}$:
   If $\min(f(y_i^1), \ldots, f(y_i^\lambda)) \leq f(x_i)$:
   Replace $x_i$ by some randomly chosen $y_i^j$ with minimal $f$-value.
4. Stopping If stopping criterion not met continue at line 2.

CLONALG [8]

Parameters $\mu$: population size $d$: selection pressure $\beta$: offspring population size factor

1. Initialisation Create an initial population $P = \{x_1, x_2, \ldots, x_\mu\}$.
2. Clonal Selection and Expansion For all $i \in \{1, 2, \ldots, \mu\}$:
   a) Create $\lambda$ clones of $x_i$ and place them in a clonal pool $C_i = \{y_i^1, \ldots, y_i^\lambda\}$.
   b) For all $j \in \{1, \ldots, \lambda\}$: Apply inversely fitness-proportional hypermutation to $y_i^j$.
3. Selection for Replacement Keep the $\mu$ best search points from $P \cup C_1 \cup \ldots \cup C_\mu$, breaking ties uniformly at random.
4. Metadynamics Replace $d$ search points with lowest fitness by new random ones.
5. Stopping If stopping criterion not met continue at line 2.

Variants
1. Non-elit selection for replacement
2. Keep best search point from $x_i \cup C_i$ for all $i \in \{1, \ldots, \mu\}$

opt-IA [3, 4, 5]

Parameters $\mu$: population size $\lambda$: offspring population size

1. Initialisation Create an initial population $P = \{x_1, x_2, \ldots, x_\mu\}$.
2. Clonal Selection and Expansion For all $i \in \{1, 2, \ldots, \mu\}$:
   a) Create $\lambda$ clones of $x_i$ and place them in a clonal pool $C_i = \{y_i^1, \ldots, y_i^\lambda\}$.
   b) For all $j \in \{1, \ldots, \lambda\}$: If (H) Then Apply hypermutation with mutation potential to $y_i^j \rightarrow C_i^H$.
      Else $C_i^H = \emptyset$.
      If (M) Then Apply contiguous hypermutations to $y_i^j \rightarrow C_i^M$.
      Else $C_i^M = \emptyset$.
3. Metadynamics Apply aging to $P$, $C_i^H$, and $C_i^M$.
4. Selection for Replacement Set $P = P \cup C_1^H \cup \ldots \cup C_\mu^H \cup C_1^M \cup \ldots \cup C_\mu^M$.
   If $|P| \geq \mu$ Then Keep the $\mu$ best search points from $P$, breaking ties u.a.r. and removing duplicates.
      Else Keep all search points in $P$; fill up $P$ with random points.
5. Stopping If stopping criterion not met continue at line 2.
Analysing Artificial Immune Systems as Optimisers

Observation artificial immune systems as optimisers are randomised search heuristics used for optimisation just as evolutionary algorithms, ant colony optimisation, particle swarm optimisation, simulated annealing, iterated random local search, . . .

Consequence AIS as optimisers should be considered the same way as other RSH as optimisers applied as other RSH analysed as other RSH

Fact analysis of RSH as optimisers is important topic

Introduction to runtime analysis Pietro Oliveto, Per Kristian Lehre
Bio-inspired computation Frank Neumann, Carsten Witt
Black-box complexity Benjamin Doerr, Carola Doerr

Analysing Artificial Immune Systems

Why?

Because ’gaining a better understanding’
  • of general limitations (black-box complexity)
  • of behaviour in typical situations (example functions)
  • of impact of specific operators (operators in (1+1)-frame)
  • of parameter settings (simple algorithms with 1 parameter)
  • for particular problem classes (classes of functions; combinatorial optimisation problems)

Because ’design of better randomised search heuristics’
  • know when not to apply
  • have an idea of when to apply
  • have an idea of ’good’ operators
  • have an idea of ’good’ parameter values
  • have an idea of what kind of RSH
  • ...
Analysing Randomised Search Heuristics

Artificial Immune Systems?
Yet another class of Randomised Search Heuristics?
Why should I care?

Facts  artificial immune systems offer
• useful alternative design paradigm for RSHs
• have different operators with different properties
  ⇻ useful in different situations
• can be a simpler and at least equally efficient alternative to
crossover-based EAs

Now
• overview of three different types of AIS-specific mutation
• considering ageing as example for a ‘meta-dynamic’
• example of a complete AIS in combinatorial optimisation

Contiguous Hypermutations

Remember (here with \( r = 1 \))

1 location, no wrapping

\[
\begin{align*}
\text{Prob (1st bit mutated)} &= \frac{1}{2} \\
\text{Prob (only 1st bit mut.)} &= \frac{1}{2n} \\
\text{1-bit mut. rather unlikely}
\end{align*}
\]

1 location, wrapping

\[
\begin{align*}
\text{Prob (1st bit mutated)} &= \frac{1}{2} \\
\text{Prob (only 1st bit mut.)} &= \frac{1}{2n} \\
\text{1-bit mut. rather unlikely}
\end{align*}
\]

2 locations

\[
\begin{align*}
\text{Prob (1st bit mutated)} &= \frac{1}{2} \\
\text{Prob (only 1st bit mut.)} &= \frac{1}{2n} \\
\text{1-bit mut. rather unlikely}
\end{align*}
\]

Analysing Contiguous Hypermutations

Method
• Insert mutation operator in (1+1)-framework.
  ⇻ study of effects in isolation
• Prove general observations.
  ⇻ knowledge of general properties
• Compare with (1+1) EA (with mutation probability \( 1/n \)) on
  well-known example functions.
  ⇻ assessment of effects under well-known circumstances
• Find examples with extremely differing performance.
  ⇻ clear understanding of benefits and drawbacks

Observation  method not unique to contiguous hypermutations
but generally applicable for study of operators

Remember  use Java applet, try for yourself
http://www.cs.bham.ac.uk/~zargesc/ais-tutorial/

Results About Contiguous Hypermutations (Part 1)

General Observation  \( \forall f \) with unique global optimum:
\[
\begin{align*}
\text{E (T}_{1+1}^{\text{CHM}, w. f} \text{)} &= \Omega(n) \\
\text{E (T}_{1+1}^{\text{CHM}, w. f} \text{)} &= \Omega(n^2) \\
\text{E (T}_{1+1}^{\text{CHM}, f} \text{)} &= \Omega(n^2) \\
\text{due to probability of final mutation} \\
\text{(all bounds tight)}
\end{align*}
\]

Comparison  for ONEMAX(x) = \( \sum_{i=1}^{n} x[i] \)
\[
\begin{align*}
\text{E (T}_{1+1}^{\text{EA, ONEMAX}} \text{)} &= \Theta(n \log n) \\
\text{E (T}_{1+1}^{\text{CHM}, w. f, ONEMAX} \text{)} &= O(n^2 \log n) \\
\text{E (T}_{1+1}^{\text{CHM}, w. ONEMAX} \text{)} &= \Theta(n^2 \log n) \\
\text{E (T}_{1+1}^{\text{CHM}, ONEMAX} \text{)} &= \Theta(n^2 \log n) \\
\text{due to difficulty of making 1-bit improvements}
\end{align*}
\]
Results About Contiguous Hypermutations (Part 2)

Comparison  for $\text{LEADINGONES}(x) = \sum_{i=1}^{n} x[i]$

$E\left(T_{1+1} \text{EA,LEADINGONES}\right) = \Theta(n^2)$
$E\left(T_{\text{CHM}_{1,wo,LEADINGONES}}\right) = O(n^2)$
$E\left(T_{\text{CHM}_{1,w,LEADINGONES}}\right) = \Theta(n^2 \log n)$
$E\left(T_{\text{CHM}_{2,LEADINGONES}}\right) = \Theta(n^2 \log n)$

since only position of left-most flipping bit matters

Comparison  for $n \cdot \text{LEADINGONES}(x) - \text{OneMAX}(x)$, init. in $0^n$

$E\left(T_{1+1} \text{EA,LEADINGONES}\right) = \Theta(n^2)$
$E\left(T_{\text{CHM}_{1,wo,LEADINGONES}}\right) = O(n)$
$E\left(T_{\text{CHM}_{1,w,LEADINGONES}}\right) = O(n^2 \log n)$
$E\left(T_{\text{CHM}_{2,LEADINGONES}}\right) = \Theta(n^2)$

since sequence of 0-bits at end only significant advantage if improving mutations easy to find

Summary Contiguous Hypermutations

- **difficulties** with flipping single bits
  - $\approx$ bad at locating optima precisely
  - $\Rightarrow$ combine with other operators if locating optima precisely matters
- **advantages** when huge mutations are needed
  - $\Rightarrow$ worth a try when hill-climbing not effective
- some variants with strong positional bias
  - $\approx$ advantages/disadvantages depending on function
  - $\Rightarrow$ only use variants with positional bias if known facts about objective function make that appear useful
- all noticeable effects rely on $r \approx 1$
  - $\approx$ even $r = 1 - \varepsilon$ ($\varepsilon > 0$ constant) not useful
  - $\Rightarrow$ use $r = 1 - o(1)$, e.g., $r = 1 - 1/n$

(for details see Jansen/Zarges (2011) [20])

Results About Contiguous Hypermutations (Part 3)

Demonstrate  very large performance difference
to demonstrate understanding of benefits and drawbacks

$CLOB_{b,k}(x) = n \cdot \left( \sum_{h=1}^{k} \sum_{j=1}^{n/k} x\left(h - 1 \cdot (n/k) + j\right) \right) - \text{OneMAX}(x)$

Example  with $n = 24$, $k = 3$, $b = 2$

$x = 011011101100110011101100111110111011$

$CLOB_{2,3}(x) = 24 \cdot (1 + 1 + 2) - 17 = 79$

Comparison  for $CLOB_{b,k}$ (with $n/(k \cdot b) \in \mathbb{N}, t := n/k$)

$E\left(T_{1+1} \text{EA,CLOB}_{b,k}\right) = \Theta(k \cdot b \cdot (t + \log k))$
$E\left(T_{\text{CHM,CLOB}_{b,k}}\right) = O(n^2 \log n)$  (all 3 variants)
since length of block does not matter

Consider  four variants

- $\text{MP}_{\text{no tabu, blind}}$ (as above, tabu=false)
- $\text{MP}_{\text{tabu, blind}}$ (as above, tabu=true)
- $\text{MP}_{\text{no tabu}}$ (tabu=false, evaluate and stop at first improvement)
- $\text{MP}_{\text{tabu}}$ (tabu=true, evaluate and stop at first improvement)

Hypermutations with Mutation Potential

Remember  Hypermutation($x$)  (for $x \in \{0, 1\}^n$, minimise $f$)

1. number of mutations steps $m(f(x)) := \left\lfloor \frac{1 - f(x)}{c \cdot n} \right\rfloor$
2. Repeat $m$ times
   - If tabu=false then select $i \in \{1, 2, \ldots, n\}$ u. a. r.
   - Else select $i \in \{1, 2, \ldots, n\}$ not previously chosen u. a. r.
3. local mutation: $x[i] := 1 - x[i]$

Consider  four variants

- $\text{MP}_{\text{no tabu, blind}}$ (as above, tabu=false)
- $\text{MP}_{\text{tabu, blind}}$ (as above, tabu=true)
- $\text{MP}_{\text{no tabu}}$ (tabu=false, evaluate and stop at first improvement)
- $\text{MP}_{\text{tabu}}$ (tabu=true, evaluate and stop at first improvement)
Results About Hypermut. with Mutation Potential (Part 1)

Comparison for \( \text{ZERO\_MIN}(x) = n + 1 - \text{ONE\_MAX}(x) \)
\[
E\left(T_{(1+1}\text{EA,ZERO\_MIN}}\right) = \Theta(n \log n)
\]
\[
E\left(T_{\text{MP\_no\_tabu,blind,ZERO\_MIN}}\right) = 2\Omega(n)
\]
(even with high probability)
\[
E\left(T_{\text{MP\_tabu,blind,ZERO\_MIN}}\right) = 2\Omega(n)
\]
(even with high probability)
due to drift to middle (due to blindness)
\[
E\left(T_{\text{MP\_no\_tabu,ZERO\_MIN}}\right) = \Theta(n^2 \log n)
\]
\[
E\left(T_{\text{MP\_tabu,ZERO\_MIN}}\right) = \Theta(n^2 \log n)
\]
due to additional evaluations

Summary Hypermutations with Mutation Potential

- blind variants difficulties locating specific points
  - bad at locating optima precisely
  - combine w. other operators if precise hits matter
- blind variants performs mostly blind random walk
  - hardly ever useful
  - if used at all, only in combination with other operators
- first improvement version can do local search (less efficient)
  - no replacement for local search/standard bit mutations
  - prefer local search if you want local search
- first improvement version can locate remote optimal regions
  - useful for such objective functions
  - use as costly alternative to local search/standard bit mutation if such properties are suspected
- depends heavily on actual function values
  - sensitive with respect to trivial transformations
  - prefer rank-based variants
  (for details see Jansen/Zarges (2011) [21])
Inverse Fitness-“Proportional” Hypermutations

Remember

Normalisation
\[ \text{opt } f(x) = f(x)/f_{\text{opt}} \in [0, 1] \]
\[ \text{best } f(x) = f(x)/f_{\text{current } \text{best}} \in [0, 1] \]

Mutation probabilities
\[ \text{CLONALG } e^{-f(x)/\rho} \]
\[ \text{opt-aiNet } e^{-f(x)/\rho} \]

resulting in four variants
- CLONALG_{\text{opt}}
- CLONALG_{\text{best}}
- opt-aiNet_{\text{opt}}
- opt-aiNet_{\text{best}}

Results About Inverse Fitness-“Prop.” Hypermut. (Part 1)

Remember
- CLONALG_{\text{opt}} inefficient even with \( \rho = \ln n \)

How is this possible in practice?
- under-estimating opt improves (see CLONALG_{\text{best}})
- OneMax not necessarily realistic
- bad performance empirically only for rather large values of \( n \)

Think

\[ (n/2) \pm \sqrt{n} \quad n - n/\ln n \]
\[ (n/2) - \sqrt{n} \quad n - n/\ln n \]
\[ (n/2) + \sqrt{n} \quad n - \ln n \]
\[ n = 10^5 \]
\[ n = 10 \]
\[ n = 10^2 \]
\[ n = 10^3 \]

Summary Inverse Fitness-“Proportional” Hypermutations

- can be very inefficient in simple situations
  \( \sim \) e.g., bad at hill climbing
  \( \Rightarrow \) use only when needed
- using 'current best' appears superior to 'optimal value' for normalisation
  \( \sim \) populations useful
  \( \Rightarrow \) prefer population-based approaches and 'current best' for normalisation
- CLONALG very sensitive with respect to \( \rho \)
  \( \sim \) very bad performance easy to achieve
  \( \Rightarrow \) prefer opt-aiNet
- only analytical results for OneMax
  \( \sim \) most points open
  \( \Rightarrow \) investigate more

(for details see Zarges (2008), (2009), (2011) [36, 37, 38])
Metadynamics in Artificial Immune Systems

**Remember** metadynamics influence behaviour of algorithm in a more global way

\[ \tau_{\text{max}} \]

\[ \rightarrow \text{more difficult to analyse than an operator} \]

**Example** ageing

**Remember** ageing has parameter maximal age \( \tau_{\text{max}} \)

comes in different variants (static pure, stochastic, . . .)

depends non-trivially on implementation details

**Remember** *method* for analysis/work programme

- insert in simple algorithmic framework
- prove general observations
- compare with known algorithms on known problems
- find extreme examples to understand benefits and drawbacks

### Parameter Study: The Maximal Age

**Note** maximal age \( \tau_{\text{max}} \) must not be too small

\[ \tau_{\text{max}} = \mathcal{O}(n^k \log n) \]

\[ \leadsto \text{very inefficient} \]

\[ \tau_{\text{max}} = \Omega(n^k \log n) \]

\[ \leadsto \text{efficient} \]

**See** appropriate range for \( \tau_{\text{max}} \) can be extremely narrow

\[ \tau_{\text{max}} = \mathcal{O}(n^k \log n) \text{ or } \tau_{\text{max}} = \Omega(n^k \log n) \]

\[ \leadsto \text{very inefficient} \]

\[ \tau_{\text{max}} = \omega(n^k \log n) \text{ and } \tau_{\text{max}} = O(n^k - k^2) \]

\[ \leadsto \text{efficient} \]

(for details see Horoba, Jansen, Zarges (2009) [17])

**Note** maximal age \( \tau_{\text{max}} \) must not be too large

\[ \tau_{\text{max}} = \Omega\left(\log n(n^{k_1} + \mu \log n)\right) \]

\[ \leadsto \text{very inefficient} \]

\[ \tau_{\text{max}} = O\left(n^{k_1 - k_2}\right) \text{ and } \tau_{\text{max}} = \omega(\log n(n^2 + \mu n \log n)) \]

\[ \leadsto \text{efficient} \]

### Comparing Ageing Variants

**Static Pure Ageing**

- new search points get age 0 if they improve
- superior in local optima

**Evolutionary Ageing**

- new search points get age 0 always
- superior on plateaus

**Combine** both into genotypic ageing

‘new search points get age 0 unless they are copy or worse’ combines advantages, good on plateaus and at local optima

(for details see Jansen, Zarges (2011) [22])
Understanding Specific Benefits

Observation  ageing performs restarts in a complicated way
And nothing more?

Idea  ageing can perform partial restart
      e.g., useful when crossover combines new and old search points

And this makes a difference?

Paying Attention to Details (cont.)

Ageing Variants  replace worst search point and among those
1. a random one
2. one with min. age distance from new one
3. one with most frequent age
4. one with rarest age

\[ E(T) = 2^{\Omega(n)} \] even with high probability

\[ E(T) = O((\mu + (n/\log \mu)) \cdot (\tau_{\max} + n^2 + \mu n \log n)) \]

\[ E(T) = \Omega((1 + n/\mu \log \mu) \cdot (\tau_{\max} + n^2 + \mu n \log n)) \]

\[ \Theta((1 + n/\mu \log \mu) \cdot (\tau_{\max} + n^2 + \mu n \log n)) \]

\[ E(T) = 2^{\Omega(n)} \] even with high probability

(for details see Jansen, Zarges (2010) [19])

Summary Ageing

- ageing adds new dynamics and new capabilities
  - \( \sim \) increased potential at the price of additional parameter
    \( \Rightarrow \) use with care
- ageing very sensitive with respect to maximal age
  - \( \sim \) difficult to set additional parameter
    \( \Rightarrow \) perform careful parameter study
- different ageing variants have different capabilities
  - \( \sim \) no ‘one size fits all’ solution
    \( \Rightarrow \) try different variants
- ageing very sensitive with respect to implementation details
  - \( \sim \) algorithmic details need to be reported precisely
    \( \Rightarrow \) pay attention to details, communicate choices precisely
Introduction

Natural Immune System

AIS as Classifiers

AIS as Optimisers

Analysing AIS

Conclusion & References

Analysing AIS in Combinatorial Optimisation

Fact analysis for example functions
(either commonly used or specifically designed)
important first step in understanding, not end of story

Observation more interesting, relevant, realistic
analysis on combinatorial optimisation problems
(see Neumann, Witt (2010) [31] for EAs)

Fact has been started for AIS, too
namely for the B-Cell algorithm
for Vertex Cover and Longest Common Subsequences

Summary Vertex Cover

• BCA alternative without crossover to EAs
• Ordering heuristic for encoding instead of “cheating” possible
• Known analyses for EAs reproducible for BCA
• On complete bipartite graph more efficient than (1+1) EA;
only slightly more inefficient than (1+1) EA with restarts
• On amplified complete bipartite graphs considerably more
efficient than mutation-based EAs
• No need for crossover and population on example graph
• Difficult to find hard instances with “bad approximation ratio
with high probability”
• BCA alternative to EAs with respect to efficiency;
more easier to analyse
• for details see [18]

The Vertex Cover Problem (VC)

Input undirected Graph \( G = (V, E) \)
Output smallest subset \( V' \subseteq V \) covering all edges, i.e.
\( V' \subseteq V \) with \( \forall e \in E : e \cap V' \neq \emptyset \)

Example

Facts
• “classic” NP hard optimisation problem
• simple 2-approximation algorithm
• no 1.3606-approximation (if \( P \neq NP \))
(no \( (2-\epsilon) \)-approximation under stronger assumptions)

The Longest Common Subsequence Problem

Input \( m \) sequences \( X_1, X_2, \ldots, X_m \in \Sigma^* \)
Output common subsequence \( Y \)
with \( \forall Y' \in \Sigma^* : Y' \) is common subsequence \( \Rightarrow |Y'| \leq |Y| \)

Examples
• Finite alphabet: \( \Sigma = \{0, 1\}, \Sigma = \{A, C, G, T\} \)
• Finite sequences \( \in \Sigma^* : X_1 = ACTGTGCAA \)
• Subsequences of a sequence:
  \( AGTA \) of \( ACTGTCAA \)

Facts
• General case is NP hard
• In \( P \) with fixed \( m \)
• Solvable using dynamic Programming in \( O \left( m \cdot \prod_{i=1}^{m} |X_i| \right) \)
Artificial Immune Systems are
• models of the natural IS △ tool for research in immunology
• heuristic approach to classification based on an example of complex classification from nature
• randomised search heuristics capable of optimisation
  • based on a quite different natural metaphor (compared to EAs)
  • an alternative approach to optimisation, with different characteristics and capabilities
  • an alternative solution if your favourite approach fails
• randomised search heuristics like many others
  • another field of study, worthy of analysis just like EAs/ACO/PSO/…
• another example of a complex class of RSHs △ another opportunity to study differences and similarities hopefully some day leading to useful taxonomy
• a fascinating area of research
References II


References IV


References V

