The Double Digest Problem

using genetic algorithms

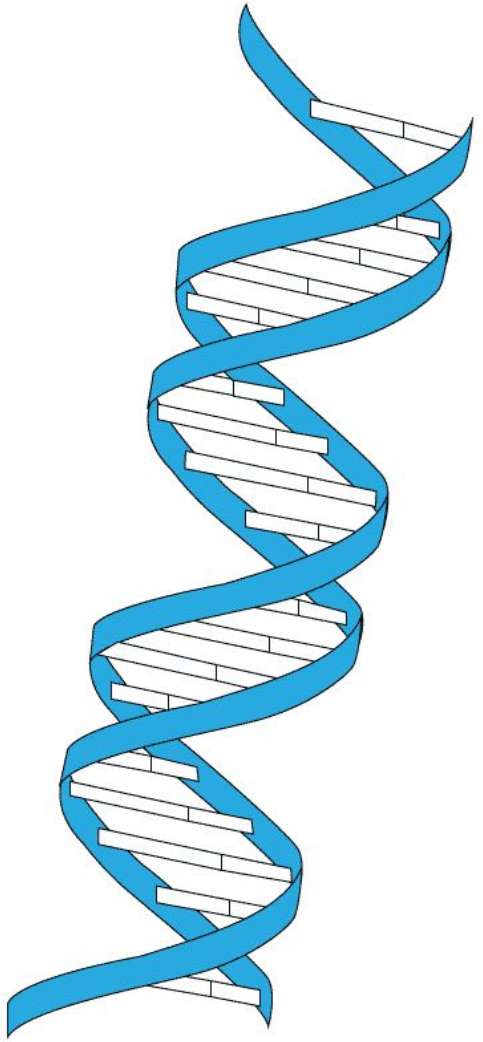
David Kuboň & Petr Martišek

Problem definition – biology

- **restriction enzyme:** enzyme that cuts the DNA at specific (short) nucleotide sequences
- **restriction site:** an occurrence of a sequence of nucleotides specific for a given restriction enzyme
- **restriction map:** positions of all the restriction sites in the DNA sequence
- given sorted set of positions $X = \{x_1, x_2, \ldots x_n\}$:
  - **partial digest:** $\delta X = \{x_j - x_i \mid 1 \leq x_i < x_j \leq x_n\}$
  - **full digest:** $\Delta X = \{x_2 - x_1, x_3 - x_2, \ldots, x_n - x_{n-1}\}$
- **restriction mapping problem:** given an experimentally obtained subset $E \subseteq \delta X$, reconstruct X

Problem definition – biology

- **Double Digest Problem:**
  - sequence S, enzymes A and B
  - input
    - ΔA … full digest using enzyme A
    - ΔB … full digest using enzyme B
    - ΔAB … full digest using both enzymes
  - output
    - A … location of the cuts for enzyme A
    - B … location of the cuts for enzyme B

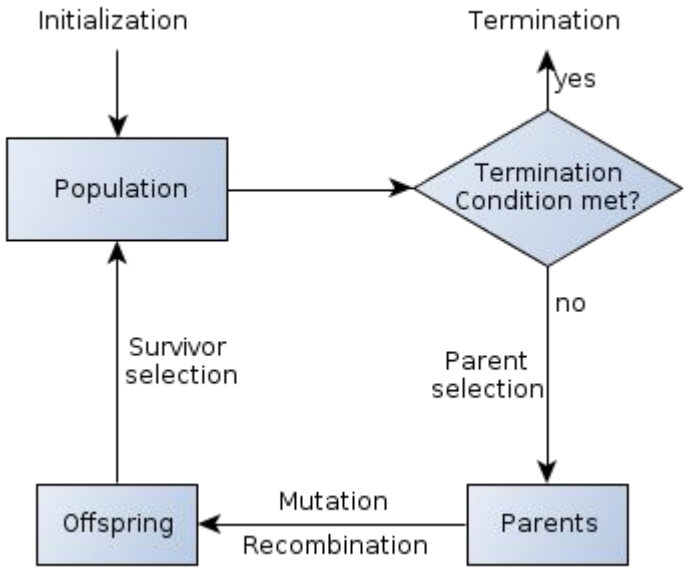


Problem definition – computer science

- input:
  - ΔA … fragment length using enzyme A
  - ΔB … fragment lengths using enzyme B
  - ΔAB … fragment lengths using both enzymes
- task:
  - find a permutation of ΔA and ΔB such that when we cut the sequence using both the obtained maps A and B simultaneously, the resulting fragments will be ΔAB
- NP-complete

Genetic algorithms

- meta-algorithm inspired by natural evolution
- solutions to a given problem are encoded as chromosomes
- the (randomly initiated) population undergoes **crossover**, **mutation** and **selection** using a defined **fitness function**



Genetic algorithms — encoding of an individual

- a candidate solution to a DDP is encoded as **a pair of permutations** of the set of fragments ΔA and the set of fragments ΔB
  - similar to the *Traveling Salesman Problem*
- example:

Input data

ΔA = {375, 282, 2205, 746, 2352, 9040}

ΔB = {3518, 1887, 389, 5916, 2017, 1273}

ΔAB = {375, 282, 2205, 656, 90, 1797, 389, 166, 5750, 2017, 1273}

One possible individual

([3, 0, 1, 5, 2, 4] , [1, 4, 3, 0, 2, 5])

Genetic algorithms – fitness function

- fitness of an individual is given by computing the combined restriction map AB and comparing the **generated** set of fragments ΔAB’ with the **given** set of fragments ΔAB
- fitness is computed as **number of matches** divided by **total number of fragments in ΔAB**

Fitness function – example

Input data

ΔA = [375, 282, 2205, 746, 2352, 9040]; ΔB = [3518, 1887, 389, 5916, 2017, 1273]

ΔAB = [375, 282, 2205, 656, 90, 1797, 389, 166, 5750, 2017, 1273]

Individual

([4, 5, 3, 1, 2, 0], [4, 2, 5, 3, 1, 0])

Generated cuts

A = [0, 2352, 11392, 12138, 12420, 14625, 15000]

B = [0, 2017, 2406, 3679, 9595, 11482, 15000]

Generated ΔAB' fragments vs. input ΔAB fragments (sorted)

ΔAB' = [54, **90**, **282**, 335, **375**, **656**, **1273, 1797, 2017, 2205,** 5916]

ΔAB = [**90**, 166, **282**, **375**, 389, **656**, **1273, 1797, 2017, 2205**, 5750]

Fitness = #matches / count(ΔAB) = 8 / 11 = 0.7272

Genetic algorithms – mutation

- **swapping mutation**
  - randomly choose two elements in the permutation and swap them

$$[4, \mathbf{5}, 3, \mathbf{1}, 2, 0] \rightarrow [4, \mathbf{1}, 3, \mathbf{5}, 2, 0]$$

- **inversion mutation**
  - randomly choose a portion of the permutation and invert it

$$[4, \mathbf{5, 3, 1, 2}, 0] \rightarrow [4, \mathbf{2, 1, 3, 5}, 0]$$

Genetic algorithms — crossover

- **Order 1 Crossover**
  - A swath of consecutive alleles from parent 1 with remaining values placed in the order of parent 2.

Parent 1: [8, 4, 7, 3, 6, 2, 5 ,1, 9, 0]

Parent 2: [0, *1, 2, 3,* 4, *5, 6,* 7, 8, 9]

Child 1: [0, 4, 7, 3, 6, 2, 5, 1, 8, 9]

- **PMX Crossover**
  - A swath is taken from parent 1 and the corresponding swath from parent 2 is sprinkled about i child. Then the remaining alleles are copied directly from parent 2.

Parent 1: [8, 4, 7, 3, 6, 2, 5 ,1, 9, 0]

Parent 2: [0, *1, 2, 3, 4, 5, 6, 7,* 8, 9]

Child 1: [0, 7, 4, 3, 6, 2, 5, 1, 8, 9]

Genetic algorithms — selection

- **Tournament**
  - Individuals chosen randomly
  - The strongest wins the right to mate

- **Roulette**
  - Individuals are assigned circle segments based on their fitness
  - A roulette is spun to choose one

Experiments

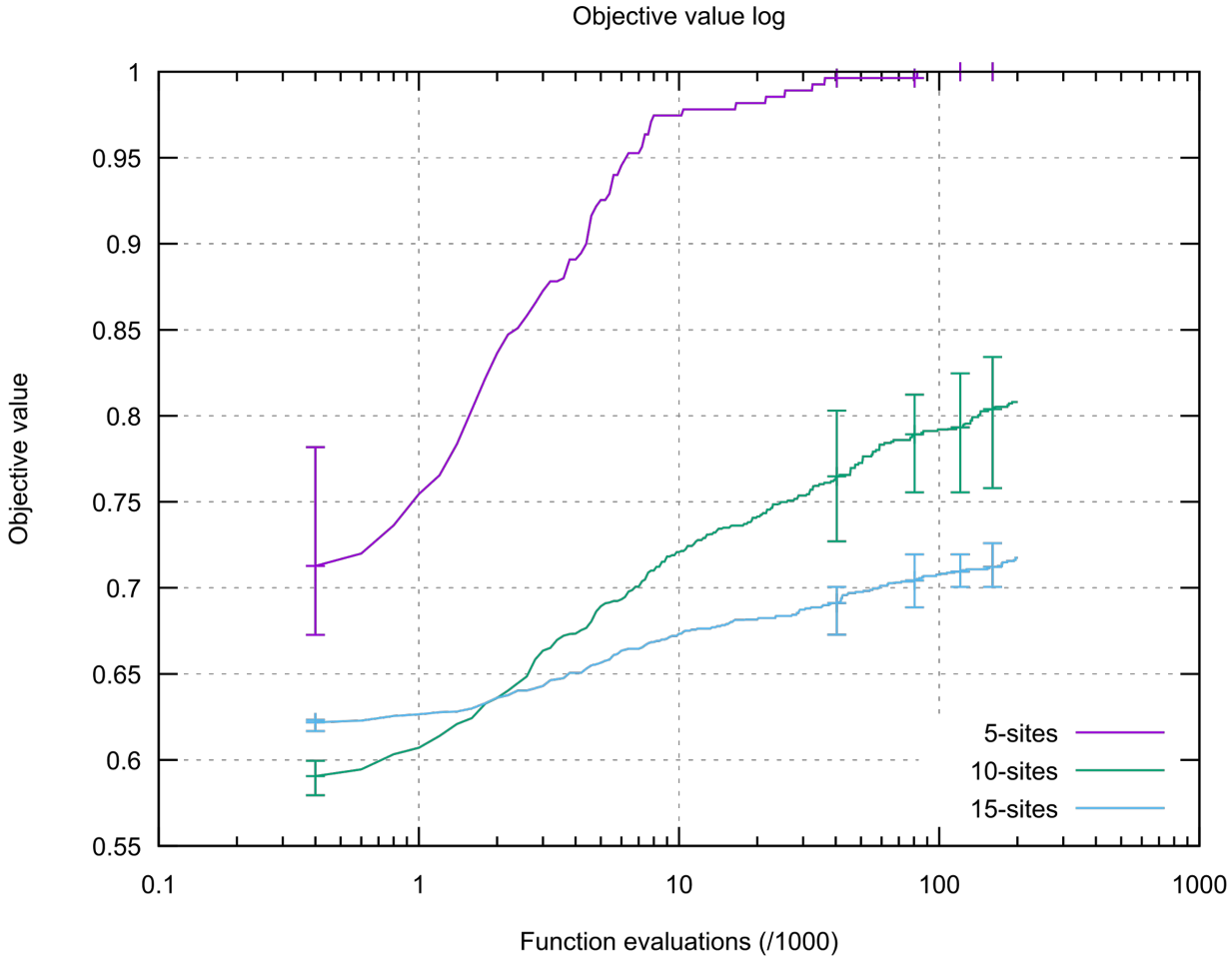
Test data

	Bombyx mori *(bourec morušový)*		**Drosophilia** *(octomilka)*		**Macaque** *(makak)*		**House mouse** *(myš domácí)*		**Chimpanzee** *(šimpanz učenlivý)*	
	enzyme	*#sites*	*enzyme*	*#sites*	*enzyme*	*#sites*	*enzyme*	*#sites*	*enzyme*	*#sites*
short	BspHI EcoRI	5 5	PacI SacI	5 5	ScaI BsgI	5 5	XbaI BtsI	5 5	SmlI Eco57I	5 5
mid	TseI HphI	10 10	TseI BbvI	9 9	FalI TstI	10 10	TatI Bdai	10 10	SspI AccI	10 10
long	PacI Hin4I	16 18	BdaI TfiI	12 13	FauI HaeIV	15 15	EcoRII SfaNI	15 15	ApoI TseI	16 16

Cuts performed by http://www.restrictionmapper.org

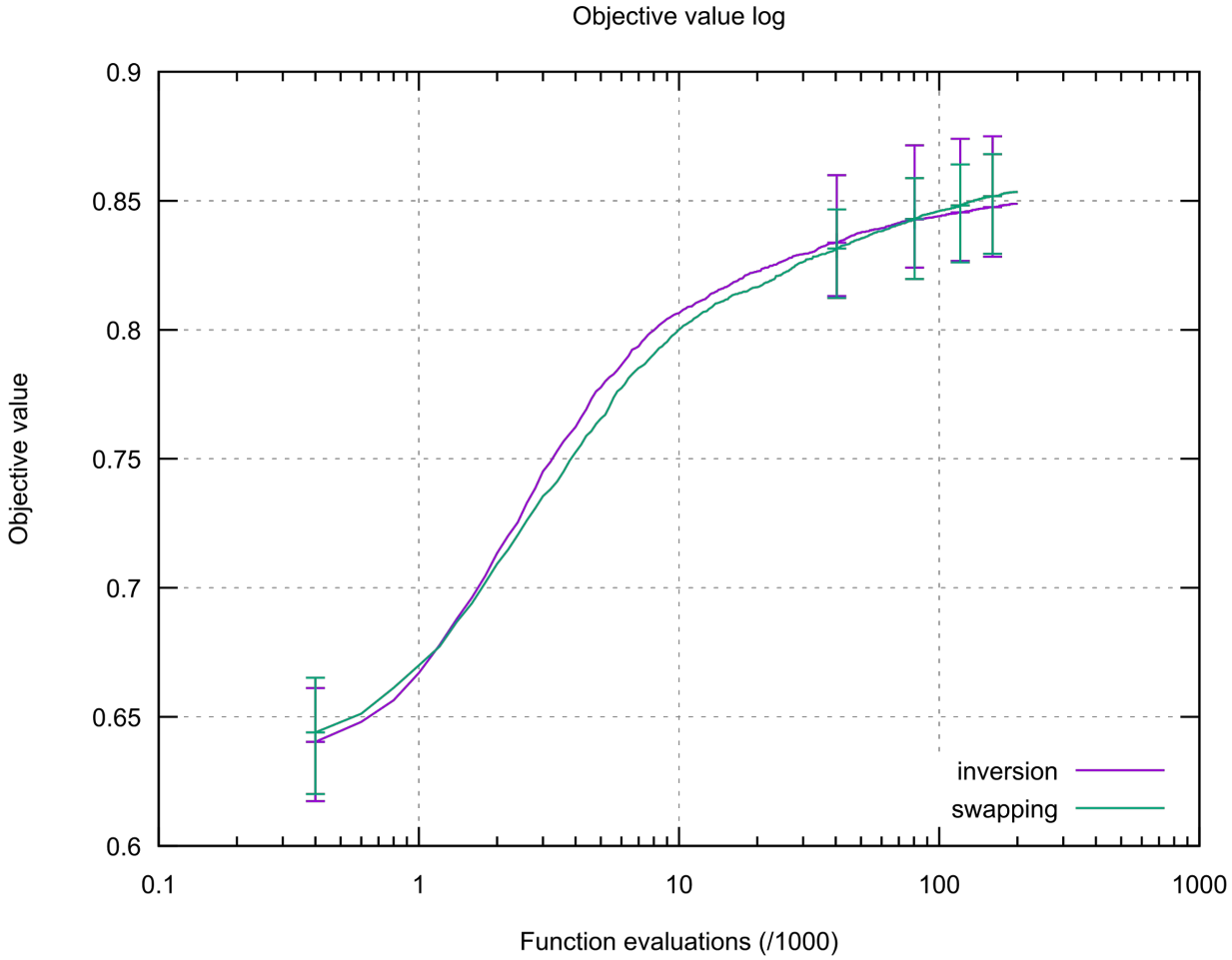
Convergence for increasing number of restriction sites

- 1000 generations
- 200 individuals in population



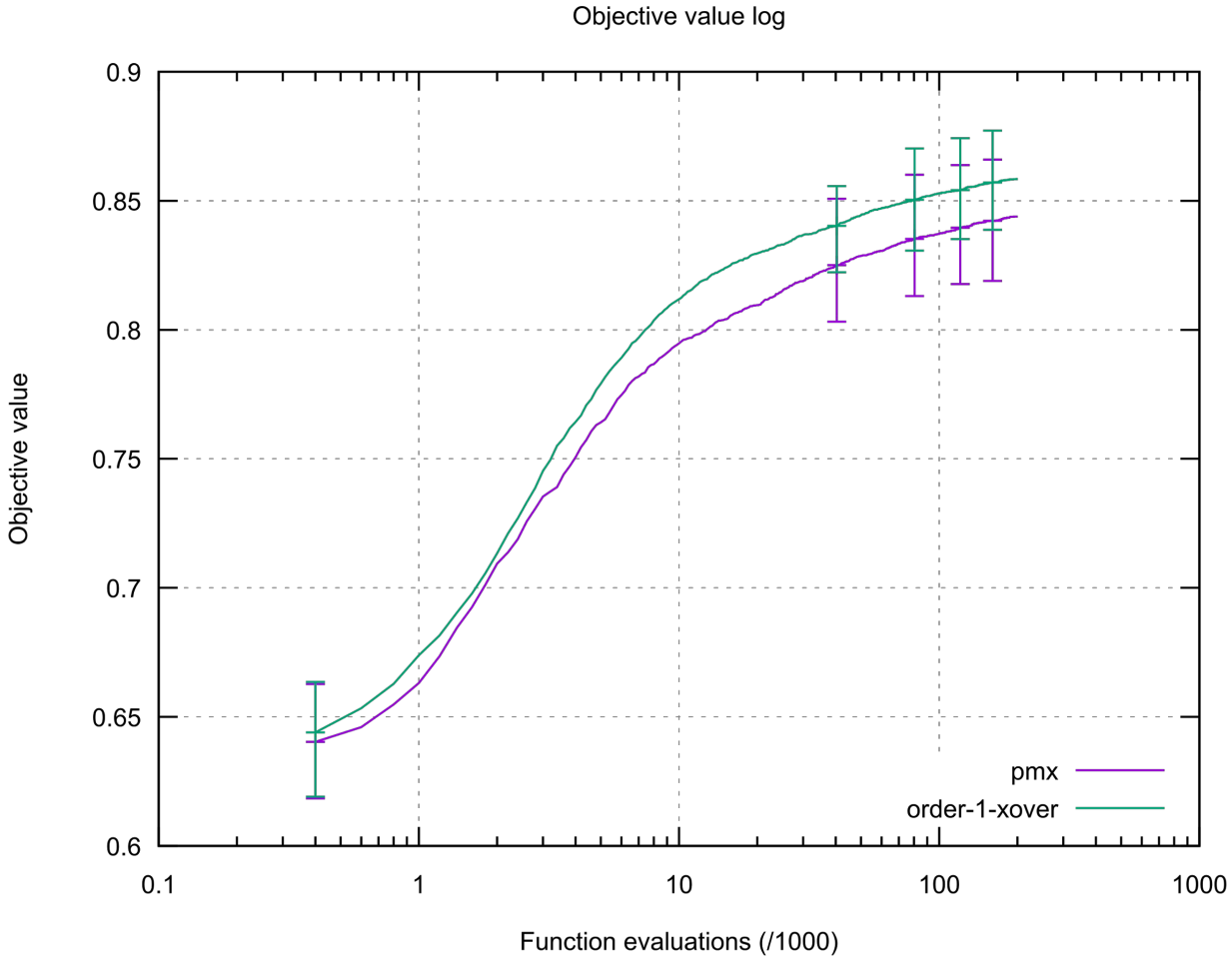
Mutation

- 1000 generations
- 200 individuals in population



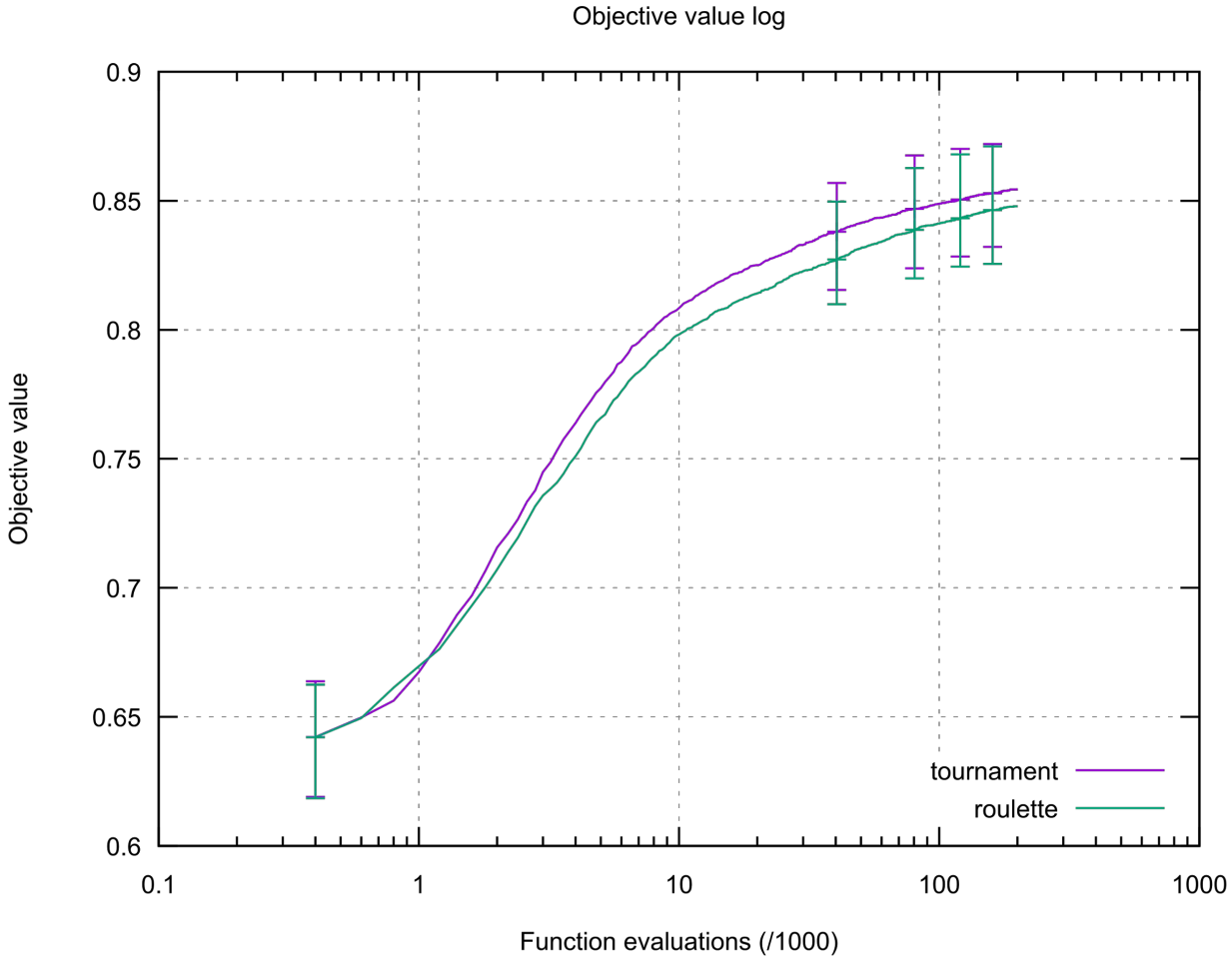
Crossover

- 1000 generations
- 200 individuals in population



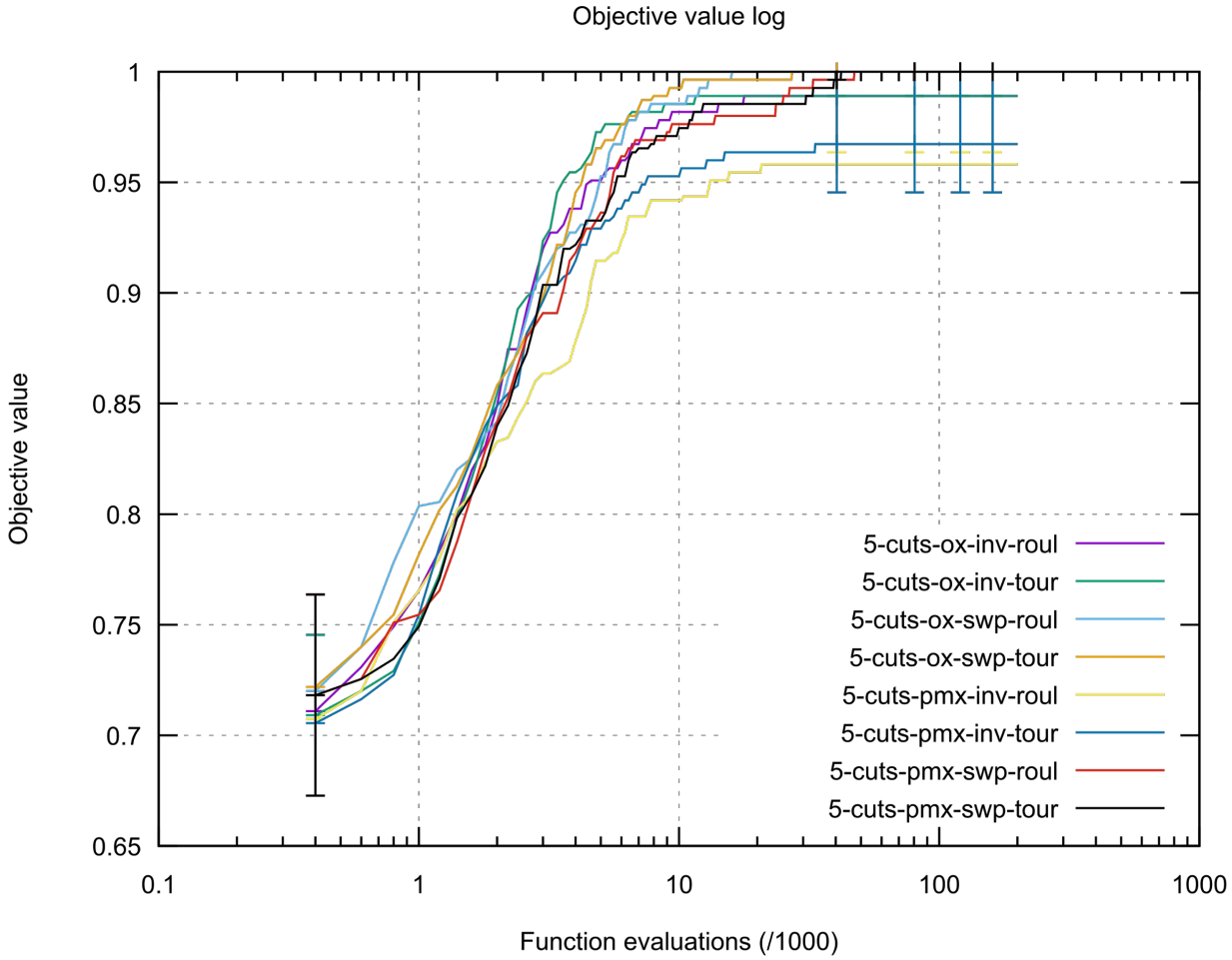
Selection

- 1000 generations
- 200 individuals in population



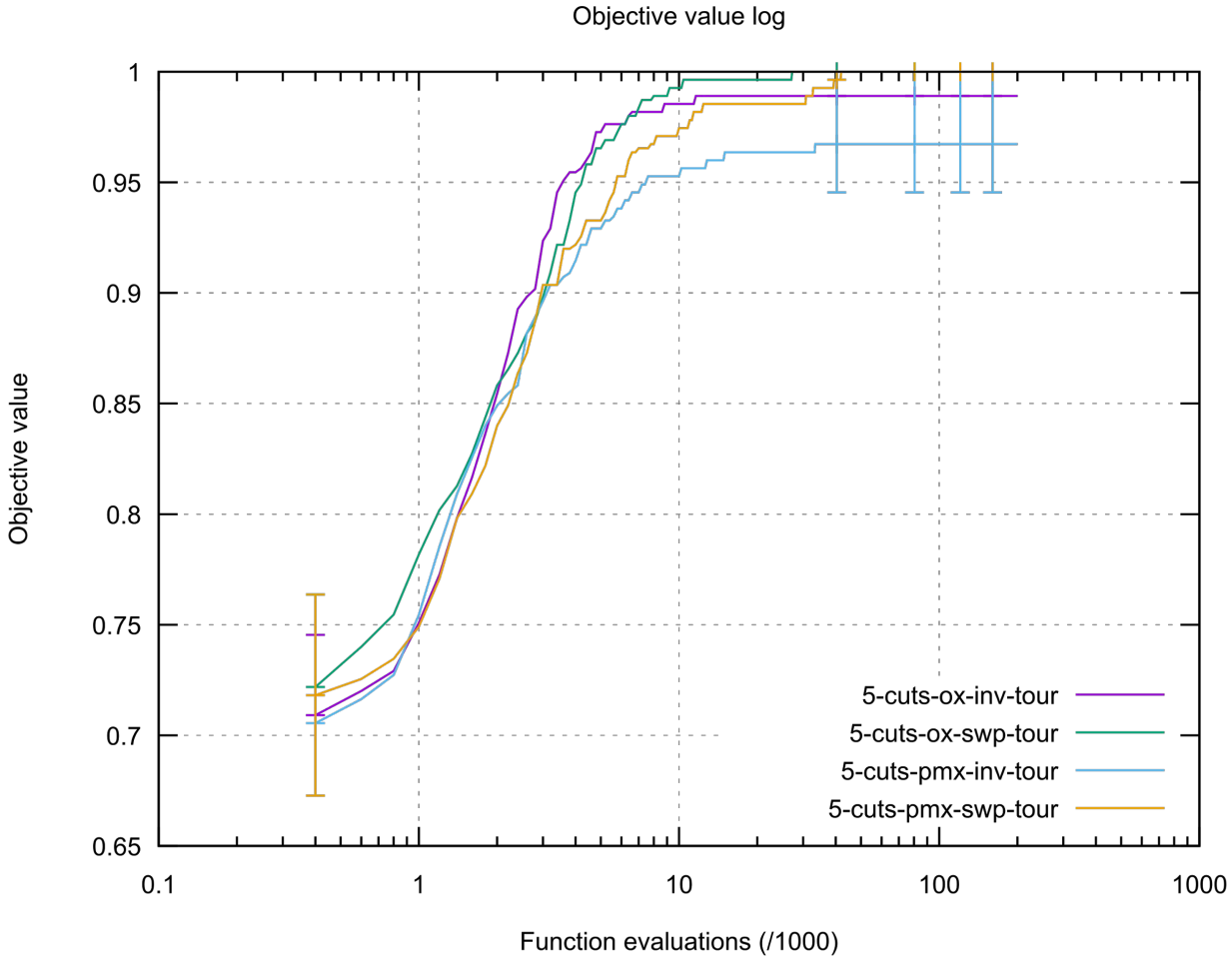
Short data

- 1000 generations
- 200 individuals in population



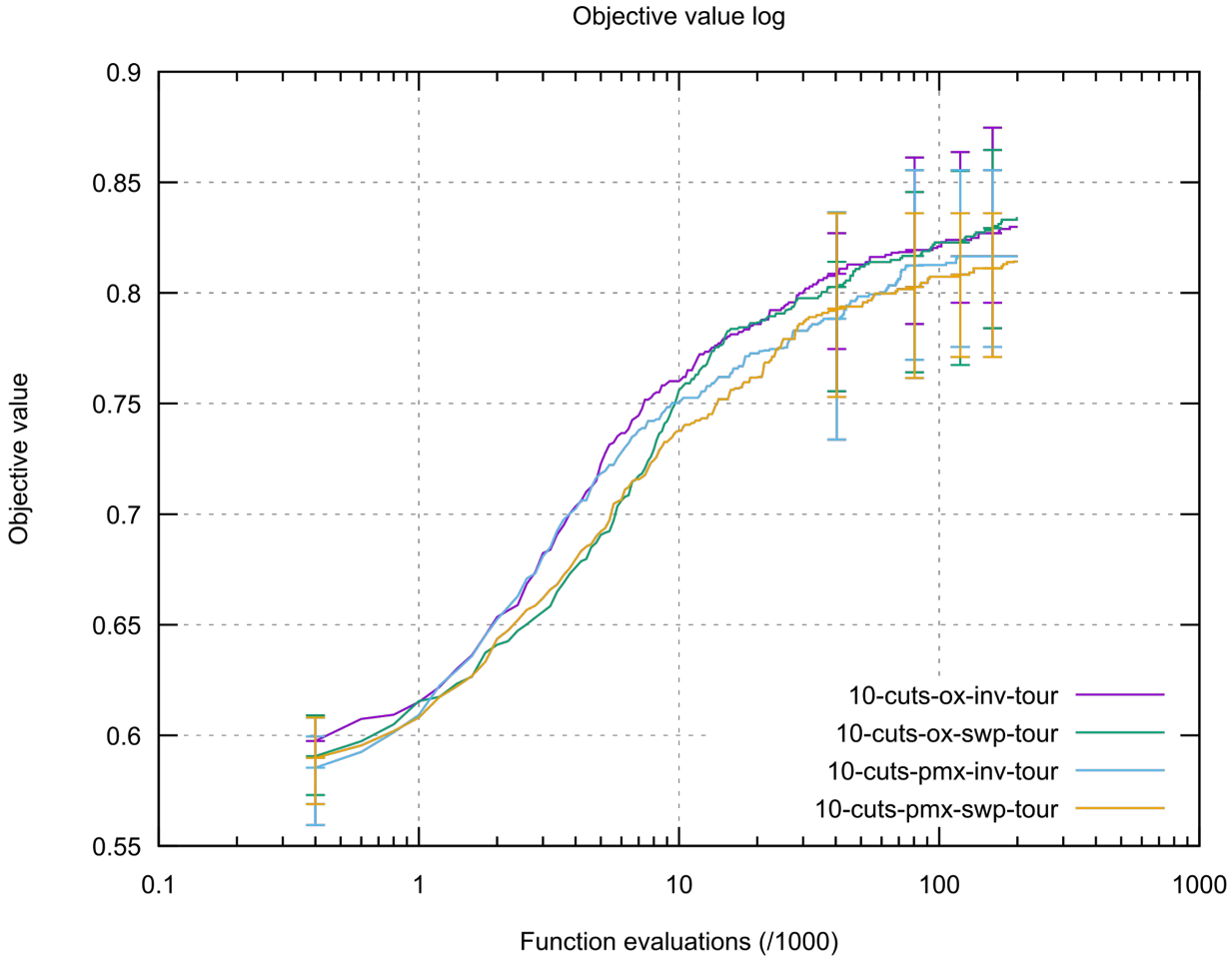
Short data

- 1000 generations
- 200 individuals in population
- tournament selection



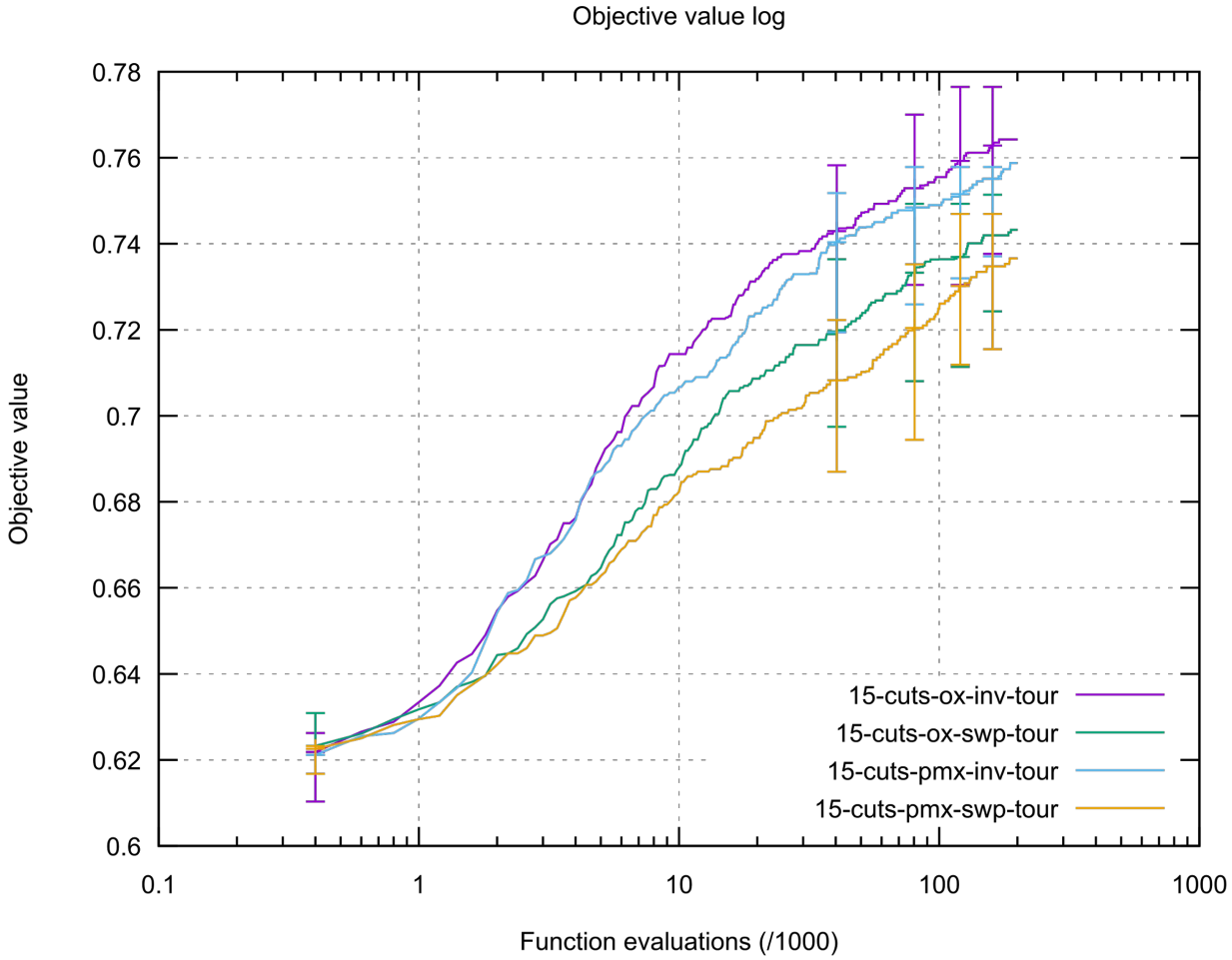
Mid-length data

- 1000 generations
- 200 individuals in population
- tournament selection



Long data

- 1000 generations
- 200 individuals in population
- tournament selection



CSP comparison

- **First attempt**
  - Artificial sequence with 10 restriction sites **easy** to find
  - GA: 1s CSP: 20s
    - ΔA = [1 2 3 4 5 6 7 8 9 10]
    - ΔB = [1 10 1 10 1 10 1 10 1 10]
    - ΔAB = [1 10 1 1 9 1 2 8 1 3 7 1 4 6]
- **Second attempt**
  - Artificial sequence with 10 restriction sites **harder** to find
  - GA: 3s CSP: 45 minutes and still running
    - ΔA = [1 2 3 4 5 6 7 8 9 10 11 12]
    - ΔB = [1 2 3 4 5 6 7 8 9 10 11 12]
    - ΔAB = [1 2 3 4 2 3 6 2 5 5 3 6 3 5 5 2 6 3 2 4 3 2 1]

Related works

Construction of Restriction Maps Using a Genetic Algorithm
(Darren M. Platt, Trevor I. Dix, 1993)

- PMX + inversion mutation
- experiments with more than 2 enzymes
  - improved discrimination of the error measure, but the ability of the algorithm to converge on the correct solution is not enhanced
- notes the problem that small changes by mutation or crossover (e.g. swapping 2 fragments) lead to drastic change of the fitness function
  - “inability to make small iterative adjustments without making radical changes to the restriction map”

Genetic Algorithm for Double Digest Problem
(S. Sur-Kolay et col., 2005)

- attempts to find (almost) all solutions, not just a single one
- *cassette* equivalence classes

Genetic Algorithm Solution for Double Digest Problem
(M. Ganjtabesh et. col, 2012)

- handles erroneous data
  - error ~ # of unmatched fragments
- only single solution (GA halts if optimal solution found)
- faster than Sur-Kolay (probably due to simpler method and no equivalence classes)

Thank you!