Facetwise Modeling of Genetic Algorithms

Dirk Thierens

Utrecht University
The Netherlands

Run Time Complexity

- In typical application the total run time of a genetic algorithm is determined by the number of fitness function evaluations.
- Run time of selection algorithm and variation operators can be ignored.
- Number of fitness function evaluations is equal to the number of generations times the population size:

$$\boxed{\#FitnessFct.Evals = \#Generations \times PopulationSize}$$

Convergence speed

- Rate at which a population converges is determined by the selection pressure:
  - high selection pressure: fast convergence
  - low selection pressure: slow convergence
- Size of population determines quality of solution found:
  - large population size: more reliable convergence
  - small population size: less reliable convergence
- Trade-off between selection pressure and population size

Key questions

1. How long does a GA - with a certain selection pressure - runs before it converges ?
2. What is the minimal population size to ensure reliable convergence ?

$\rightarrow$ problem dependent, but:

- We can build analytical models for simple problems,
- Use this as an approximation for some real, complex problems,
- Gives insight in and guidance for designing performant GAs.

Models

1. First, we will build analytical models for the convergence behavior, assuming large enough populations,
2. Second, we will build analytical models for the minimal required population size,
3. Third, we will test the models on a real, complex problem (map labeling).

Selection Intensity

- To quantify the speed of convergence we need a quantitative measure of selection pressure.
- The selection differential $S(t)$ is the difference between the mean fitness of the parent population at generation t and the population mean fitness at generation t.
- The selection intensity $I(t)$ is the scaled selection differential, obtained by dividing by the standard deviation of the fitness values.
- $I(t)$ is dimensionless since the standard deviation has the units in which the selection differential is expressed:

$$I(t) = \frac{S(t)}{\sigma(t)} = \frac{\bar{f}(t^s) - \bar{f}(t)}{\sigma(t)}.$$

Counting Ones fitness function

- Counting Ones, 'fruit fly' of GA theory

$$CO(X) = \sum_{i=1}^{\ell} x_i \qquad x_i \in \{0, 1\}$$

- Probability having 1 at a certain locus: $p(t)$
- Fitness binomial distributed
- Mean fitness at generation $t : \bar{f}(t) = l.p(t)$
- Variance at gen. $t : \sigma_p^2(t) = l.p(t)(1 - p(t))$
- Recombination makes no change to population mean fitness

$\Rightarrow$ simple, yet accurate convergence models

Proportionate selection

- Probability selecting i (fitness f_i, proportion $P_i(t)$): $P_i(t^s) = P_i(t)\frac{f_i}{\overline{f(t)}}$
- Selection Differential $S(t)$:

$$
\begin{aligned}
\overline{f(t^s)} - \overline{f(t)} &= \sum_{i=1}^{N} P_i(t^s) f_i - \overline{f(t)} \\
&= \sum_{i=1}^{N} P_i(t) \frac{f_i^2}{\overline{f(t)}} - \overline{f(t)} \\
&= \frac{1}{\overline{f(t)}} (\overline{f^2(t)} - (\overline{f(t)})^2) \\
&= \frac{\sigma^2(t)}{\overline{f(t)}}
\end{aligned}
$$

- Selection intensity $I(t) = \frac{\sigma(t)}{\overline{f(t)}}$

Proportionate Selection: Counting Ones

- mean fitness increase: $\overline{f(t+1)} - \overline{f(t)} = \frac{\sigma^2(t)}{\overline{f(t)}}$
- proportion of optimal alleles p(t)

$$p(t+1) - p(t) = \frac{1}{l}(1 - p(t))$$

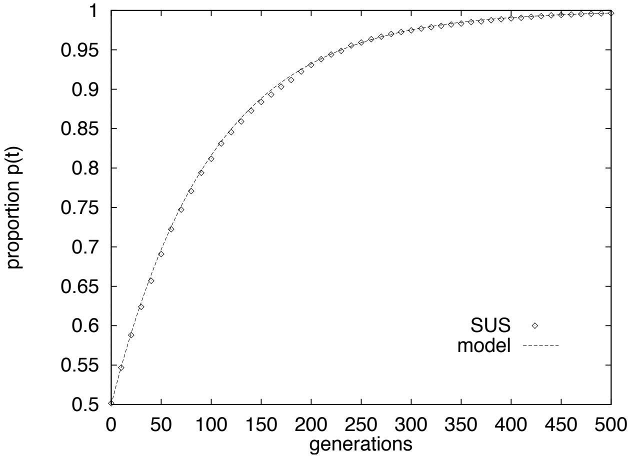
$$\frac{dp(t)}{dt} \approx \frac{1}{l}(1 - p(t))$$

- convergence model ($p(0) = 0.5$)

$$\boxed{p(t) = 1 - 0.5e^{-t/l}}$$

- convergence speed: $p(t_{conv}) = 1 - 1/(2\ell)$

$$\boxed{t_{conv} = \ell \ln(\ell)}$$


1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
proportion p(t)
0
50
100
150
200
250
300
350
400
450
500
generations
SUS
model

Truncation Selection

- Truncating a normal distribution at the top $\tau\%$ gives fitness increase proportional to the standard deviation:

$$f(t^s) - f(t) = c(\tau).\sigma(t)$$

- Selection intensity: $I(\tau) = c(\tau)$
- Values of selection intensity I for truncation selection are constant:

τ	1%	10%	20%	40%	50%	80%
I	2.66	1.76	1.2	0.97	0.8	0.34

Truncation Selection

- mean fitness increase

$$\overline{f(t+1)} - \overline{f(t)} = I\,\sigma(t)$$

- proportion of optimal alleles p(t)

$$p(t+1) - p(t) = \frac{I}{\sqrt{l}}\sqrt{p(t)(1-p(t))}$$

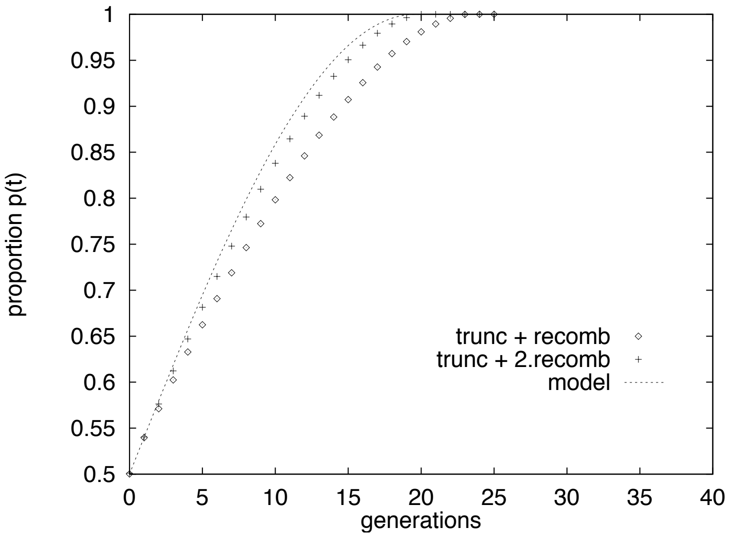
$$\frac{dp(t)}{dt} \approx \frac{I}{\sqrt{l}}\sqrt{p(t)(1-p(t))}$$

- convergence model ($p(0) = 0.5$)

$$\boxed{p(t) = 0.5(1 + \sin(\tfrac{I}{\sqrt{l}}t))}$$

- convergence speed ($p(t_{conv}) = 1$)

$$\boxed{t_{conv} = \frac{\pi}{2}\,\frac{\sqrt{l}}{I}}$$


proportion p(t)
generations
trunc + recomb
trunc + 2.recomb
model
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0
5
10
15
20
25
30
35
40

Tournament Selection

- Tournament size s: the selection intensity i is equal to the expected value of the best ranked individual of a sample from s individuals taken from the standard normal distribution:
- Can be computed using order statistics: $I = u_{s:s}$

s	2	3	4	5	6	7
$I = u_{s:s}$	$\frac{1}{\sqrt{\pi}} = 0.56$	0.85	1.03	1.16	1.27	1.35

Tournament Selection

- Same model as truncation selection, for instance for tournament size $s = 2$:
- mean fitness increase

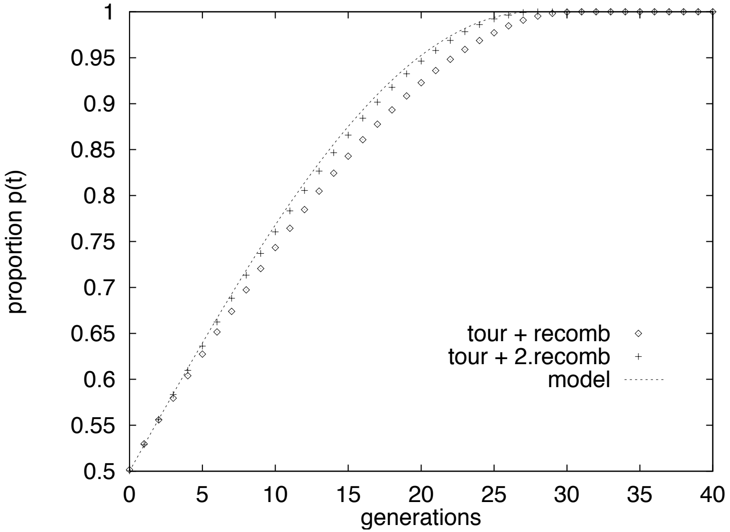
$$\overline{f(t+1)} - \overline{f(t)} = I\,\sigma(t) = \frac{1}{\sqrt{\pi}}\,\sigma(t)$$

- convergence model ($p(0) = 0.5$)

$$\boxed{p(t) = 0.5(1 + \sin(\frac{t}{\sqrt{\pi l}}))}$$

- convergence speed ($p(t_{conv}) = 1$)

$$\boxed{t_{conv} = \frac{\pi}{2}\sqrt{\pi l}}$$


proportion p(t)
generations
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0
5
10
15
20
25
30
35
40
tour + recomb
tour + 2.recomb
model

Population sizing

- Correct size of the population important:
  - too small: premature convergence to sub-optimal solutions
  - too large: computational inefficient
- We focus on the Counting-Ones problem, but the model can be extended to (slightly) more complex functions
- Key question: how does the optimal population size scales with the complexity of the problem, ie. the length of the string ?

Selection Error

- Tournament selection:
  s_1 : 1100011100, fitness = 5
  s_2 : 0100111101, fitness = 6
  $\Rightarrow$ string s_2 is selected !
- Competition at the schema level:
  (order-1 sufficient since we focus on Counting-Ones)
  - partition $f*********$:
    schema $0*********$ wins from schema $1*********$
    $\Rightarrow$ selection decision error.
  - partitions $****f*****$ and $*********f$:
    schema $****1*****$ wins from schema $****0*****$, and
    schema $*********1$ wins from schema $*********\ 0$
    $\Rightarrow$ correct selection decisions.
  - other partitions: nothing changes.

Selection Error

- What is the probability of making a selection error ?
- How many selection errors can we afford to make before the optimal bit-value at a cdertain position is completely lost in the population = premature convergence ?
- Population sizing is basically a statistical decision making problem.

Probability selection decision error

- Schemata fitness $f(H_1 : ***1*********)$ and $f(H_2 : ***0*********)$ binomial distributed
  $\rightarrow$ approximating with normal distribution $\mathbb{N}(\mu, \sigma^2)$:

$$\mu_{H_1} = 1 + (\ell - 1)p, \qquad \sigma^2_{H_1} = (\ell - 1)p(1 - p)$$

$$\mu_{H_2} = (\ell - 1)p, \qquad \sigma^2_{H_2} = (\ell - 1)p(1 - p)$$

  ($p =$ probability of having a bit value 1 at any position).
- Distribution of the fitness difference of the best schema and the worst schema $f(H_1) - f(H_2)$ is also normal distributed:

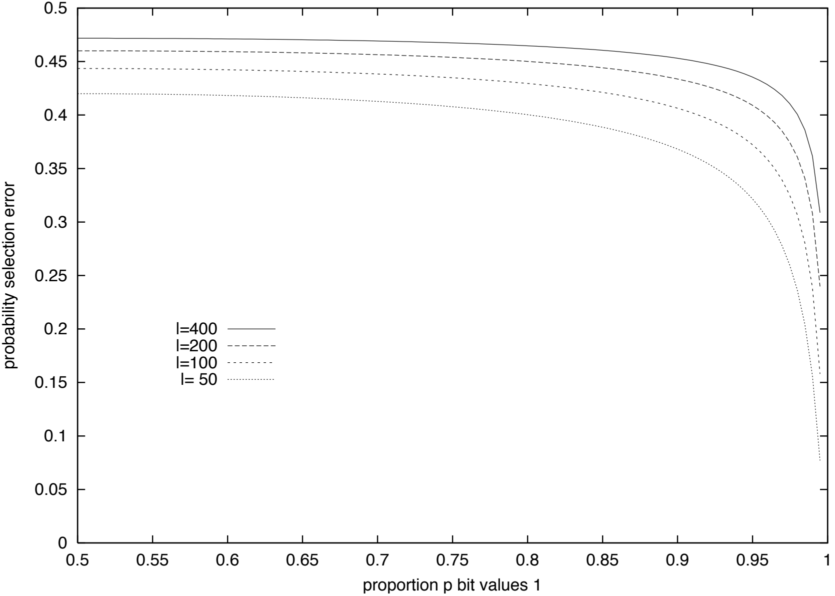
$$\mu_{H_1 - H_2} = 1, \qquad \sigma^2_{H_1 - H_2} = 2(\ell - 1)p(1 - p)$$

Probability selection decision error

- Probability selection error is equal to the probability that the best schema is sampled by a string with fitness less than the sample of the worst schema, which is equal to the probability that the fitness difference of the strings is negative:

$$
\begin{aligned}
P[SelErr] &= P(F_{H_1-H_2} < 0) \\
&= \Phi\left(\frac{-1}{\sqrt{2(\ell-1)p(1-p)}}\right)
\end{aligned}
$$

- $\Phi(x)$: Cumulative distribution function of the standard normal distribution.
- $P(X < b) = \Phi(\frac{b-\mu}{\sigma})$


probability selection error
proportion p bit values 1
l=400
l=200
l=100
l= 50
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0.5 0.55 0.6 0.65 0.7 0.75 0.8 0.85 0.9 0.95 1

Probability selection decision error

Approximation

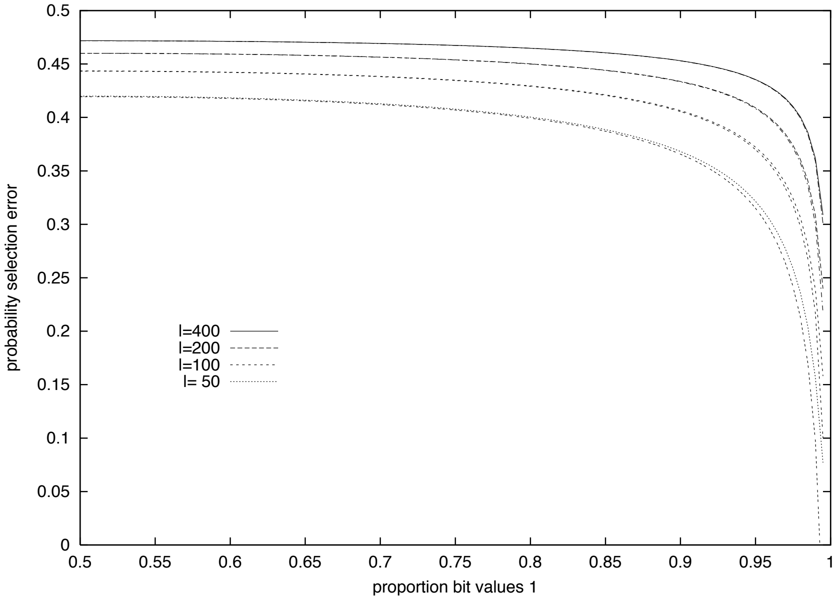
- Approximation by first two terms of power series expansion for the normal distribution:

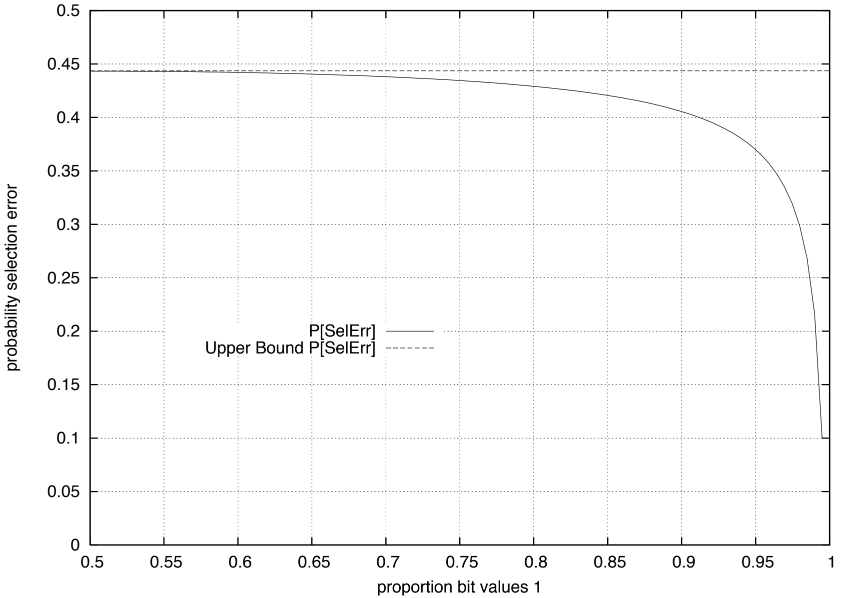
$$P[SelErr] \approx \frac{1}{2} - \frac{1}{2\sqrt{\pi(\ell-1)p(1-p)}}$$

- Selection error is upper bounded by:

$$\boxed{P[SelErr] \leq \frac{1}{2} - \frac{1}{\sqrt{\pi\ell}}}$$

this is a conservative estimate of the selection error that ignores the reduction in error probability when the proportion of optimal bit values $p(t)$ increases.


0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0.5
0.55
0.6
0.65
0.7
0.75
0.8
0.85
0.9
0.95
1
probability selection error
proportion bit values 1
l=400
l=200
l=100
l= 50


0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0.5
0.55
0.6
0.65
0.7
0.75
0.8
0.85
0.9
0.95
1
probability selection error
proportion bit values 1
P[SelErr]
Upper Bound P[SelErr]

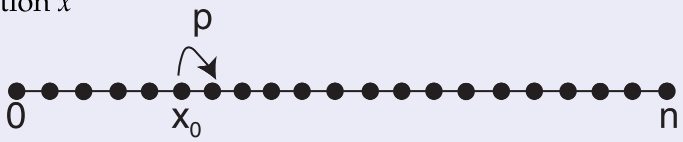
GA population sizing

How many selection errors ?

- Selection viewed as decision making process within partitions: schemata competition.
- When best schema looses competition we have a selection decision error.
- How many decision errors can we afford to make given a certain population size ?
- Answer given by Gambler's ruin model: within each partition a random walk is played.

Gambler's ruin random walk model

- one-dimensional, discrete space of size $N + 1$.
- one particle at position $x \in \{0, \ldots, N\}$.
- the particle can move one step to the right with probability p, and one step to the left with probability $1 - p$.
- when the particle reaches the boundaries ($x = 0$, or $x = N$) the random walk ends.
- call $P_N(x)$ (resp. $P_0(x)$) the probability that the particle is absorbed by the boundary $x = N$ (resp, $x = 0$) when it is currently at position x



Gambler's ruin random walk model

- Difference equation:

$$P_N(x) = pP_N(x+1) + (1-p)P_N(x-1)$$

  with boundary conditions: $P_N(N) = 1$, and $P_N(0) = 0$
- Probability the particle - starting from position x_0 - is absorbed by the $x = N$ boundary:

$$P_N(x_0) = \frac{1 - (\frac{1-p}{p})^{x_0}}{1 - (\frac{1-p}{p})^N}$$

- $P_0(x_0) = 1 - P_N(x_0)$
- when $p = 1 - p = 0.5$ we get $P_N(x_0) = \frac{x_0}{N}$

Gambler's ruin model (GR) $\rightarrow$ GA

- Position x in GR:
  $\rightarrow$ the number of optimal bit values '1' in the population at a certain partition (position in the string).
- Boundaries $x = N$ (resp. $x = 0$) in GR:
  $\rightarrow$ all bit values in the population at the partition are equal to the bit value '1' (resp, '0').
- Absorbing boundary states in GR:
  $\rightarrow$ population converged to all ones or all zeroes at that partition.
- Probability p particle moves one step to the right in GR:
  $\rightarrow$ probability that the number of optimal bit values 1 in the population at the partition is increased by one = probability correct selection decision.
- Convergence to $x = N$ (resp. $x = 0$) boundary:
  $\rightarrow$ Population converges to optimal bit value 1
  (resp. converges to wrong bit value = premature convergence).

- Recall probability selection decision error: $P[SelErr] \leq \frac{1}{2} - \frac{1}{\sqrt{\pi\ell}}$
- Probability convergence to the optimal bit value:

$$
\begin{aligned}
P[OptConv] &= \frac{1 - \left(\frac{P[SelErr]}{1-P[SelErr]}\right)^{N/2}}{1 - \left(\frac{P[SelErr]}{1-P[SelErr]}\right)^{N}} \\
&\approx 1 - \left(\frac{P[SelErr]}{1 - P[SelErr]}\right)^{N/2} \\
&\approx 1 - \left(\frac{\frac{1}{2} - \frac{1}{\sqrt{\pi\ell}}}{\frac{1}{2} + \frac{1}{\sqrt{\pi\ell}}}\right)^{N/2} \\
&\approx 1 - \left(\frac{\sqrt{\pi\ell} - 2}{\sqrt{\pi\ell} + 2}\right)^{N/2}
\end{aligned}
$$

Approximation: denominator approaches 1 much more rapidly as the numerator since $P[SelErr] < 1 - P[SelErr]$

- Taking the logs:

$$\frac{N}{2} \ln \frac{\sqrt{\pi \ell} - 2}{\sqrt{\pi \ell} + 2} \approx \ln(1 - P[OptConv])$$

- Using the Taylor series approximation:

$$\begin{aligned}
\ln \frac{x-2}{x+2} &= \ln(x-2) - \ln(x+2) \\
&\approx (\ln x - \frac{2}{x} - \frac{2}{x^2}) - (\ln x + \frac{2}{x} - \frac{2}{x^2}) \\
&\approx -\frac{4}{x}
\end{aligned}$$

we get:

$$\frac{N}{2} \frac{-4}{\sqrt{\pi \ell}} \approx \ln(1 - P[OptConv])$$

- Critical population size:

$$\boxed{N \approx \ln(1 - P[OptConv]) \; \frac{\sqrt{\pi \ell}}{-2}}$$

The minimal required population size scales as the square root of the problem complexity !

- Probability optimal bit value will be found at certain position:

$$\boxed{P[OptConv] \approx 1 - e^{\frac{-2N}{\sqrt{\pi \ell}}}}$$

Convergence string length ℓ

- The number of optimal bits F in the entire string of length ℓ is binomially distributed:

$$P(F = x) = \binom{\ell}{x} P[OptConv]^x (1 - P[OptConv])^{\ell - x}$$

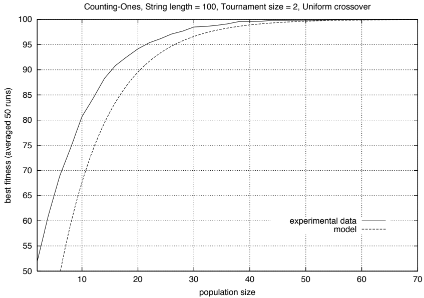
  with mean: $\mu = \ell \, P[OptConv]$
  and variance: $\sigma^2 = \ell P[OptConv](1 - P[OptConv])$,
- The probability the optimal string will be reached is:

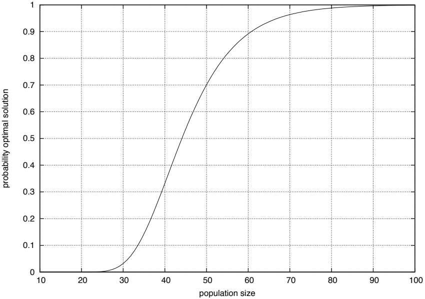
$$P[OptimalString] = P[OptConv]^{\ell}.$$

Experimental validation

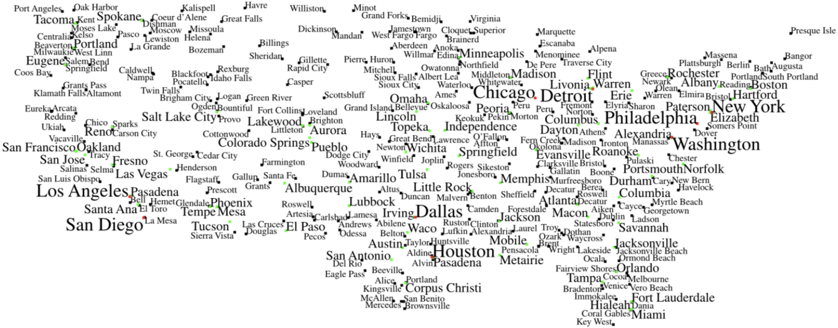
$$E[Fitness] = 100\ (1 - e^{\frac{-2N}{\sqrt{100\pi}}})$$

$$P[OptimalString] = (1 - e^{\frac{-2N}{\sqrt{100\pi}}})^{100}$$





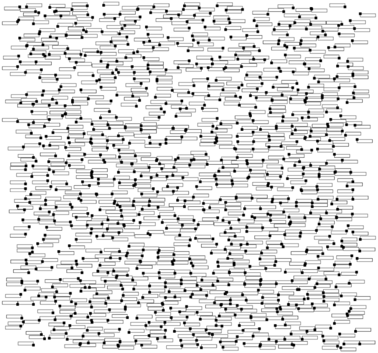
Map Labeling problem



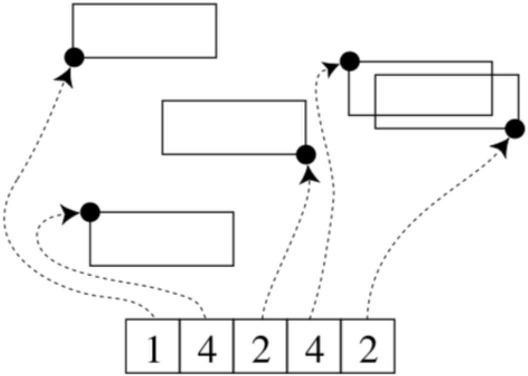
- Place labels next to map features.
- Even basic instances are NP-hard.
- Numerous cartographic rules need to be considered.

Basic map-labeling problem

- Set of points in the plane.
- Each point has rectangular fixed sized label at 4 possible positions.
- Find a labeling with maximum number of non-overlapping labels.

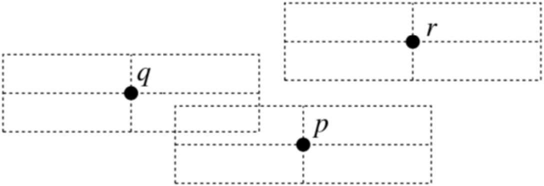


Encoding



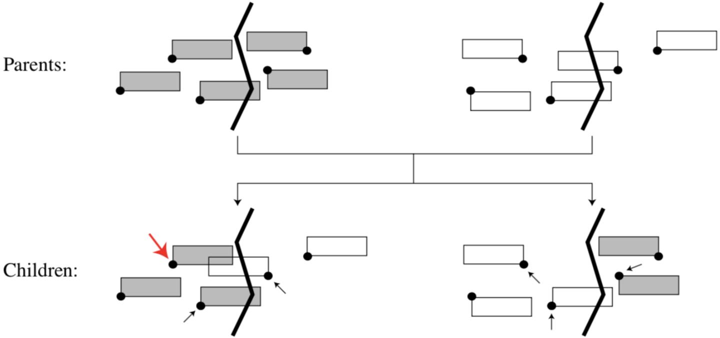
Rival Groups

- Two points are rivals if their labels can overlap.
- A point together with its rivals is called a rival group.



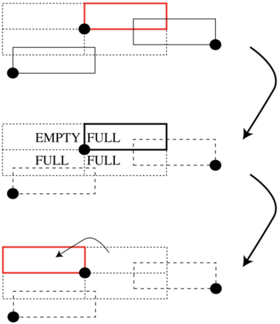
Crossover on Rival Groups

- Crossover is done by repeatedly choosing rival groups.
- Crossover is complementary: half of a parent is copied to a child and the other half is copied from the other parent.

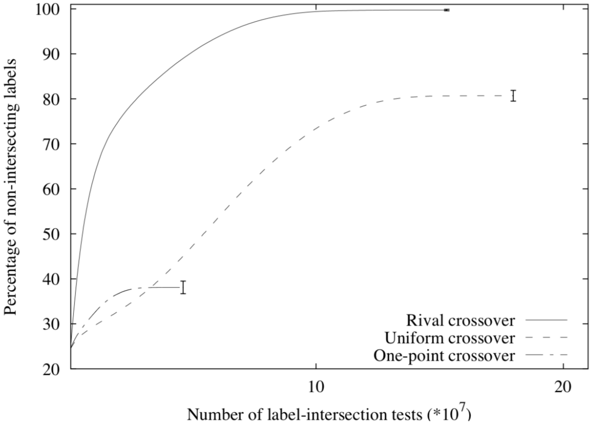


Geometric Local Search: slot filling

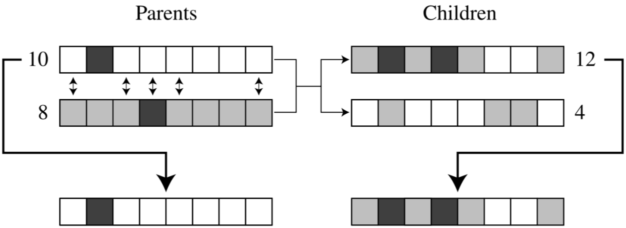
- After crossover a geometrically local optimizer is applied to points which may have a conflict.



Rival Crossover



Elitist Recombination



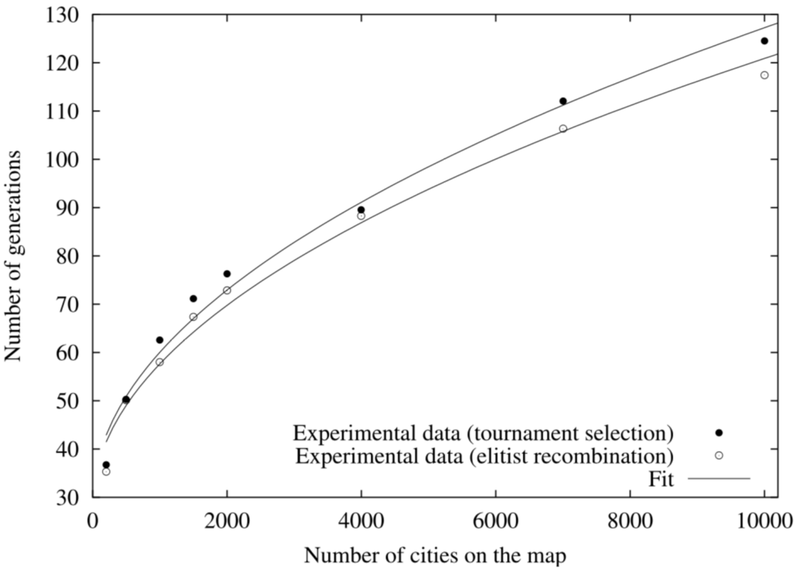
Best two of family replace parents.

Scalability

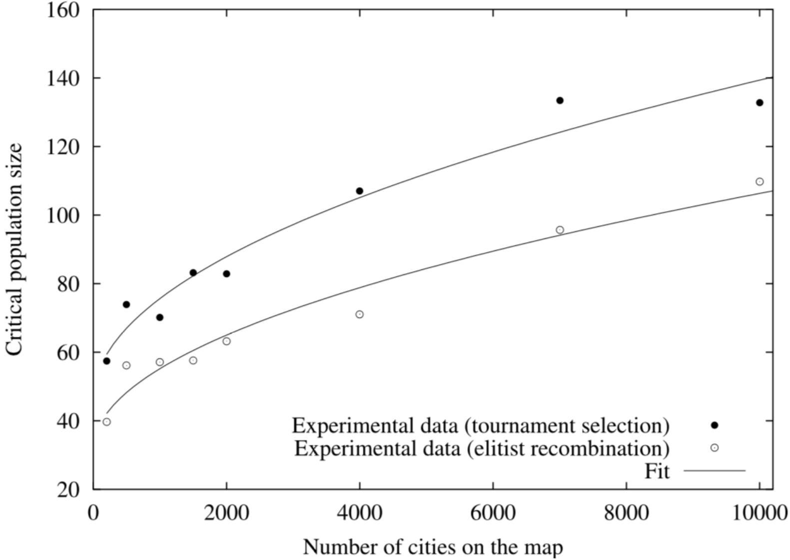
- $Cost(Eval) = O(\ell)$: each city can be checked in constant time.
- $PopSize = O(\sqrt{\ell})$: If gambler's ruin model is applicable.
- $Generations = O(\sqrt{\ell})$: If convergence model is applicable.
- $RunTime = O(\ell^2)$

$$RunTime = Cost(Eval) \times PopSize \times Generations$$

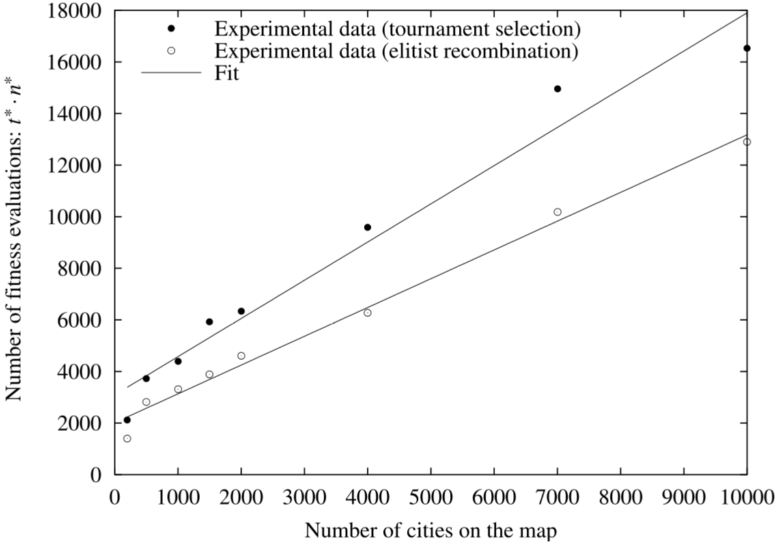
Scalability Number of Generations



Scalabiilty Minimal Population Size



Scalability Number of Fitness Evaluations



Modeling applicable ?

Assumptions of models are satisfied:

- Fitness function can be kept simple
  (uniformly scaled, semi-separable, and additively decomposable).
- Crossover is linkage-respecting and mixes well.
- Disruption is minimized by the geometrically local optimizer.

Bottom line

Theoretical insights can be used to design efficient genetic algorithms for real-world problems.