Stochastic optimization algorithms
Lecture 6, 20200911

Evolutionary algorithms:
Properties

Mattias Wahde, PhD, Professor, Chalmers University of Technology
e-mail: mattias.wahde@chalmers.se, http://www.me.chalmers.se/~mwahde

Today's learning goals

- After this lecture you should be able to
  - Derive and explain the schema theorem, and its implications
  - Derive expressions for the result of selection and mutation in infinite-population models of GAs.
  - Derive expressions for the expected running time for a simple GA.
  - Derive the optimal mutation rate for a simple GA.
  - Explain the concept of premature convergence
  - List methods for avoiding premature convergence

The schema theorem

- Schema = pattern consisting of 1,0,x, where x is a wild card (represents both 0 and 1).
- Example: 100xx1 represents 100001, 100011, 100101, and 100111.
- Different schemata have different level of importance for the problem at hand.
- Consider e.g. maximization of e^{xy}, with 3 bits per variable. Then
  - 11x11x: high level of importance
  - 0000xx: low level of importance

The schema theorem

- Schema = pattern consisting of 1,0,x, where x is a wild card (represents both 0 and 1).
- Example: 100xx1 represents 100001, 100011, 100101, and 100111.
- Different schemata have different level of importance for the problem at hand.
- Consider e.g. maximization of e^{xy}, with 3 bits per variable. Then
  - 11x11x: high level of importance
  - 0000xx: low level of importance

The schema theorem

- Schema = pattern consisting of 1,0,x, where x is a wild card (represents both 0 and 1).
- Example: 100xx1 represents 100001, 100011, 100101, and 100111.
- Different schemata have different level of importance for the problem at hand.
- Consider e.g. maximization of e^{xy}, with 3 bits per variable. Then
  - 11x11x: high level of importance
  - 0000xx: low level of importance

The schema theorem

- Schema = pattern consisting of 1,0,x, where x is a wild card (represents both 0 and 1).
- Example: 100xx1 represents 100001, 100011, 100101, and 100111.
- Different schemata have different level of importance for the problem at hand.
- Consider e.g. maximization of e^{xy}, with 3 bits per variable. Then
  - 11x11x: high level of importance
  - 0000xx: low level of importance

The schema theorem

- GAs treat schemata in such a way as to increase (in the population) the number of schemata associated with high fitness.

The schema theorem

- Let F denote the sum of fitness values, i.e. $F = \sum_{i=1}^{N} F_i$, where F_i is the fitness of individual *i*.
- Let $\bar{F}$ denote the average fitness of the whole population i.e. $\bar{F} = F/N$.
- Considering roulette-wheel selection, the probability of selecting individual *i* (in a single selection step) is then $p_{sel}(i) = F_i/F$.

The schema theorem

- Let F denote the sum of fitness values, i.e. $F = \sum_{i=1}^{N} F_i$, where F_i is the fitness of individual *i*.
- Let $\bar{F}$ denote the average fitness of the whole population i.e. $\bar{F} = F/N$.
- Considering roulette-wheel selection, the probability of selecting individual *i* (in a single selection step) is then $p_{sel}(i) = F_i/F$.

The schema theorem

- Let F denote the sum of fitness values, i.e. $F = \sum_{i=1}^{N} F_i$, where F_i is the fitness of individual *i*.
- Let $\bar{F}$ denote the average fitness of the whole population i.e. $\bar{F} = F/N$.
- Considering roulette-wheel selection, the probability of selecting individual *i* (in a single selection step) is then $p_{sel}(i) = F_i/F$.

The schema theorem

- Let $\overline{F_s}$ = the average fitness of some schema S in the population = the average of the fitness values for those individuals that contain S.
- Let $\Gamma(S, g)$ = the number of copies of S in the population (i.e. the number of individuals that contain S) in generation g of the GA.
- Then the fitness sum of those individuals, denoted F_s, can of course be written as $F_s = \overline{F_s}\Gamma(S, g)$(simply by the definition of the average).

The schema theorem

- Let $\overline{F_s}$ = the average fitness of some schema S in the population = the average of the fitness values for those individuals that contain S.
- Let $\Gamma(S, g)$ = the number of copies of S in the population (i.e. the number of individuals that contain S) in generation g of the GA.
- Then the fitness sum of those individuals, denoted F_s, can of course be written as $F_s = \overline{F_s}\Gamma(S, g)$(simply by the definition of the average).

The schema theorem

- Let $\overline{F}_S$ = the average fitness of some schema S in the population = the average of the fitness values for those individuals that contain S.
- Let $\Gamma(S, g)$ = the number of copies of S in the population (i.e. the number of individuals that contain S) in generation g of the GA.
- Then the fitness sum of those individuals, denoted F_S, can of course be written as $F_S = \overline{F}_S\Gamma(S, g)$(simply by the definition of the average).

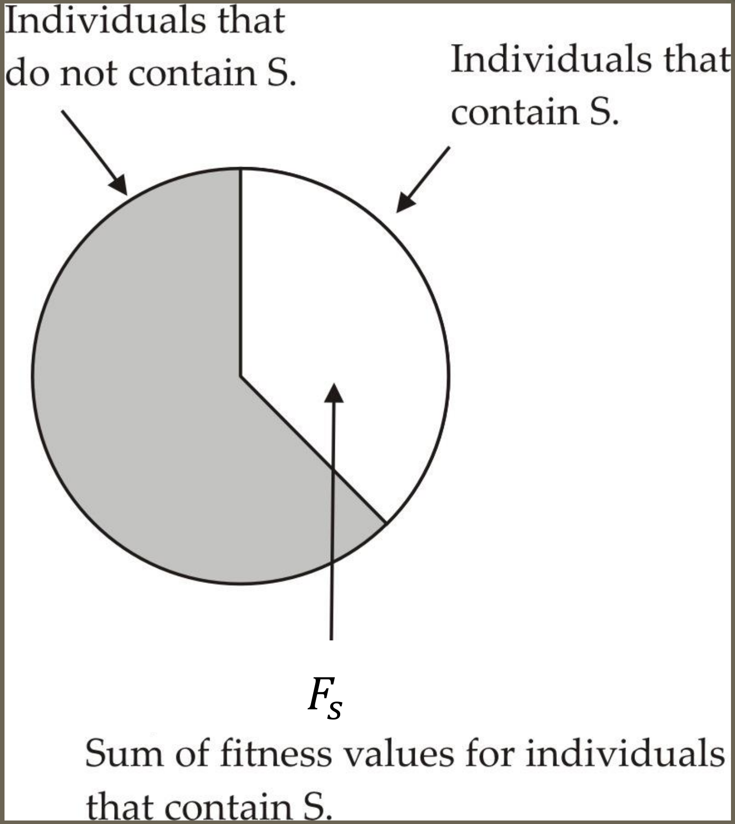
The schema theorem

- In a single selection step, what is the probability of selecting an individual containing S?

The schema theorem

- In a single selection step, what is the probability of selecting an individual containing S?
- This probability will (under roulette-wheel selection) equal the fraction of the wheel F_S/F taken up by individuals containing S.

The schema theorem


Individuals that do not contain S.
Individuals that contain S.
F_s
Sum of fitness values for individuals that contain S.

The schema theorem

- In a single selection step, therefore, the probability of selecting an individual containing schema S is equal to this ratio (F_s/F).
- During selection, there are *N* selection steps. Thus, considering selection only, the expected number of copies of S in generation g+1 will be

$$E(\Gamma(S,g+1)) = N\frac{F_s}{F} = \frac{N\Gamma(S,g)\overline{F_s}}{\mathrm{F}} = \Gamma(S,g)\frac{\overline{F_s}}{\overline{F}}$$

- ...where E() denotes the expected value.

The schema theorem

- In a single selection step, therefore, the probability of selecting an individual containing schema S is equal to this ratio (F_s/F).
- During selection, there are *N* selection steps. Thus, considering selection only, the expected number of copies of S in generation g+1 will be

$$E(\Gamma(S,g+1)) = N\frac{F_s}{F} = \frac{N\Gamma(S,g)\overline{F_s}}{\mathrm{F}} = \Gamma(S,g)\frac{\overline{F_s}}{\overline{F}}$$

- ...where E() denotes the expected value.

The schema theorem

- However, there are also the processes of crossover and mutation that tends to destroy schemata.
- Definitions:
  - The **defining length** of S (denoted D(s)) is the distance between the first and last non-wildcard.
    Example: S = 1x10x00xxx => D(S) = 7-1 = 6
  - The **order** of S (denoted O(s)) is the number of non-wildcards in S.
    Example: S = 00x0110x => O(S) = 6

The schema theorem

- However, there are also the processes of crossover and mutation that tends to destroy schemata.
- Definitions:
  - The **defining length** of S (denoted D(s)) is the distance between the first and last non-wildcard.
    Example: S = 1x10x00xxx => D(S) = 7-1 = 6
  - The **order** of S (denoted O(s)) is the number of non-wildcards in S.
    Example: S = 00x0110x => O(S) = 6

The schema theorem

- However, there are also the processes of crossover and mutation that tends to destroy schemata.
- Definitions:
  - The **defining length** of S (denoted D(s)) is the distance between the first and last non-wildcard.
    Example: S = 1x10x00xxx => D(S) = 7-1 = 6
  - The **order** of S (denoted O(s)) is the number of non-wildcards in S.
    Example: S = 00x0110x => O(S) = 6

The schema theorem

- Consider crossover: As schema is destroyed if the crossover changes the non-wildcard alleles. Since the crossover point is chosen randomly, the probability of destroying as schema S equals $P_d = D(s)/(m-1)$, where m is the chromosome length.
- The probability of survival (under crossover) is then

$$P_s = 1 \; - D(s)/(m-1)$$

The schema theorem

- Consider crossover: As schema is destroyed if the crossover changes the non-wildcard alleles. Since the crossover point is chosen randomly, the probability of destroying as schema S equals $P_d = D(s)/(m-1)$, where m is the chromosome length.
- The probability of survival (under crossover) is then
$$P_s = 1\ - D(s)/(m-1)$$

The schema theorem

- This is, in fact, an underestimate, since a broken schema can (with luck) be reassembled during crossover.
- Example: Let S = xx011x. If the crossover occurs in the middle of S, i.e. xx0|11x, then the schema will be destroyed, but it can reappear if a substring with xx0 (e.g. 000) is joined with a substring that contains 11x, i.e. either 110 or 111.

The schema theorem

- This is, in fact, an underestimate, since a broken schema can (with luck) be reassembled during crossover.
- Example: Let S = xx011x. If the crossover occurs in the middle of S, i.e. xx0|11x, then the schema will be destroyed, but it can reappear if a substring with xx0 (e.g. 000) is joined with a substring that contains 11x, i.e. either 110 or 111.

The schema theorem

- Consider mutations: As schema is destroyed if the mutation changes any non-wildcard allele.
- Let P_{mut} denote the mutation rate.
- The probability of *survival* of S (under mutation) is then equal to $(1 - P_{mut})^{O(s)}$.
- Thus, putting everything together, one finds ...

The schema theorem

- Consider mutations: As schema is destroyed if the mutation changes any non-wildcard allele.
- Let P_{mut} denote the mutation rate.
- The probability of *survival* of S (under mutation) is then equal to $(1 - P_{mut})^{O(s)}$.
- Thus, putting everything together, one finds ...

The schema theorem

- Consider mutations: As schema is destroyed if the mutation changes any non-wildcard allele.
- Let P_{mut} denote the mutation rate.
- The probability of *survival* of S (under mutation) is then equal to $(1 - P_{mut})^{O(s)}$.
- Thus, putting everything together, one finds ...

The schema theorem

$$E\big(\Gamma(S,g+1)\big) \geq \frac{\overline{F_s}}{\overline{F}}\Gamma(S,g)\left(1 - p_c\frac{d(S)}{m-1}\right)(1-p_{\mathrm{mut}})^{o(S)}$$

- ...where the inequality comes from the fact that the probability of survival of S under crossover is an underestimate (see above).
- Derivation pp. 174-176 (Appendix B2.1)

The schema theorem

- Building blocks: Schemata with
  - Low defining length
  - Low order
  - High fitness ($\overline{F_S}$)
- Building block hypothesis: GAs manipulate building blocks in an efficient way.
- No proof, but shown in empirical tests.
- The schema theorem does not, in fact, say very much that is specific (or useful) about GAs.

The schema theorem

- Building blocks: Schemata with
  - Low defining length
  - Low order
  - High fitness ($\overline{F_S}$)
- Building block hypothesis: GAs manipulate building blocks in an efficient way.
- No proof, but shown in empirical tests.
- The schema theorem does not, in fact, say very much that is specific (or useful) about GAs.

The schema theorem

- Building blocks: Schemata with
  - Low defining length
  - Low order
  - High fitness ($\overline{F_S}$)
- Building block hypothesis: GAs manipulate building blocks in an efficient way.
- No proof, but shown in empirical tests.
- The schema theorem does not, in fact, say very much that is specific (or useful) about GAs.

Today's learning goals

- After this lecture you should be able to
  - Derive and explain the schema theorem, and its implications ✓
  - Derive expressions for the result of selection and mutation in infinite-population models of GAs.
  - Derive expressions for the expected running time for a simple GA.
  - Derive the optimal mutation rate for a simple GA.
  - Explain the concept of premature convergence
  - List methods for avoiding premature convergence

Infinite-population models

- The analytical treatment of GAs becomes simpler (at least in some cases) if one lets the population size (N) tend to infinity.
- Note: The chromosome length (m) remains finite!
- Enumeration of the 2^m possible strings: j = 1,2,3,… 2^m.
- When N is infinite one obtains probability distributions (instead of frequencies) for each string (j): $p = p(j)$.

Infinite-population models

- The analytical treatment of GAs becomes simpler (at least in some cases) if one lets the population size (N) tend to infinity.
- Note: The chromosome length (m) remains finite!
- Enumeration of the 2^m possible strings: j = 1,2,3,… 2^m.
- When N is infinite one obtains probability distributions (instead of frequencies) for each string (j): $p = p(j)$.

Infinite-population models

- The analytical treatment of GAs becomes simpler (at least in some cases) if one lets the population size (N) tend to infinity.
- Note: The chromosome length (m) remains finite!
- Enumeration of the 2^m possible strings: j = 1,2,3,… 2^m.
- When N is infinite one obtains probability distributions (instead of frequencies) for each string (j): $p = p(j)$.

Infinite-population models

- The analytical treatment of GAs becomes simpler (at least in some cases) if one lets the population size (N) tend to infinity.
- Note: The chromosome length (m) remains finite!
- Enumeration of the 2^m possible strings: j = 1,2,3,... 2^m.
- When N is infinite one obtains probability distributions (instead of frequencies) for each string (j): $p = p(j)$.

Infinite-population models

- Selection, crossover, and mutation operators combined:

$$\mathcal{G}(p) = \mathcal{G}_m(p) \circ \mathcal{G}_c(p) \circ \mathcal{G}_s(p)$$

- If one considers selection only (in proportion to fitness):

$$\mathcal{G}_s(p) = \frac{F(j)p(j)}{\sum_{j \in \Omega} F(j)p(j)} = \frac{F(j)p(j)}{\bar{F}}$$

Ω = set of all possible chromosomes

Infinite-population models

- Additional simplification: Consider **functions of unitation**, i.e. function in which the fitness f only *depends on the number of ones in the chromosomes*.
- Example: the Onemax function $\mathrm{F}(j) = j$, where (NOTE!) j = the *number of ones* in the (binary) chromosome.
- From now on, $p_q(j)$ denotes the probability distribution for chromosomes with j ones (thus $m - j$ zeros) in generation q, where *j* thus ranges from 0 to *m* (*not* 2^m as before).

Infinite-population models

- Additional simplification: Consider **functions of unitation**, i.e. function in which the fitness f only *depends on the number of ones in the chromosomes*.
- Example: the Onemax function $\mathrm{F}(j) = j$, where (NOTE!) j = the *number of ones* in the (binary) chromosome.
- From now on, $p_q(j)$ denotes the probability distribution for chromosomes with j ones (thus $m - j$ zeros) in generation q, where *j* thus ranges from 0 to *m* (*not* 2^m as before).

Infinite-population models

- Additional simplification: Consider **functions of unitation**, i.e. function in which the fitness f only *depends on the number of ones in the chromosomes*.
- Example: the Onemax function $\mathrm{F}(j) = j$, where (NOTE!) j = the *number of ones* in the (binary) chromosome.
- From now on, $p_q(j)$ denotes the probability distribution for chromosomes with j ones (thus $m - j$ zeros) in generation q, where *j* thus ranges from 0 to *m* (*not* 2^{m} as before).

Infinite-population models

- Consider strings of length m.
- How many strings are there with 0 ones (hereafter: 1s)?
- Answer: 1, namely 000 ... 000.
- How many strings are there with 1 one?
- Answer: m : 100... 0 , 010 ... 0, 001 ... 0, 000 ... 1.
- In general, there are $\binom{m}{j}$ strings containing j ones.

Infinite-population models

- Consider strings of length m.
- How many strings are there with 0 ones (hereafter: 1s)?
- Answer: 1 , namely 000 ... 000.
- How many strings are there with 1 one?
- Answer: m : 100... 0 , 010 ... 0, 001 ... 0, 000 ... 1.
- In general, there are $\binom{m}{j}$ strings containing j ones.

Infinite-population models

- Consider strings of length m.
- How many strings are there with 0 ones (hereafter: 1s)?
- Answer: 1 , namely 000 ... 000.
- How many strings are there with 1 one?
- Answer: m : 100... 0 , 010 ... 0, 001 ... 0, 000 ... 1.
- In general, there are $\binom{m}{j}$ strings containing j ones.

Infinite-population models

- Consider strings of length m.
- How many strings are there with 0 ones (hereafter: 1s)?
- Answer: 1 , namely 000 ... 000.
- How many strings are there with 1 one?
- Answer: m : 100... 0 , 010 ... 0, 001 ... 0, 000 ... 1.
- In general, there are $\binom{m}{j}$ strings containing j ones.

Infinite-population models

- Consider strings of length m.
- How many strings are there with 0 ones (hereafter: 1s)?
- Answer: 1 , namely 000 ... 000.
- How many strings are there with 1 one?
- Answer: m : 100... 0 , 010 ... 0, 001 ... 0, 000 ... 1.
- In general, there are $\binom{m}{j}$ strings containing j ones.

Infinite-population models

- Thus, the initial distribution, assuming random initialization, takes the form:

$$p_1(j) = 2^{-m}\binom{m}{j}$$

number of string containing j ones

divide by the total number of strings of length m = 2^m

Infinite-population models

- One can now compute the average fitness in the first generation as

$$\overline{F}_1 = \sum_{j=0}^{m} j p_1(j) = 2^{-m} \sum_{j=0}^{m} j \binom{m}{j} = \frac{m}{2}$$

Infinite-population models

- One can now compute the average fitness in the first generation as

$$\bar{F}_1 = \sum_{j=0}^{m} j p_1(j) = 2^{-m} \sum_{j=0}^{m} j \binom{m}{j} = \frac{m}{2}$$

$\sum_{j=0}^{m} j \binom{m}{j} = m2^{m-1}$. Eq. (B17) in Appendix B and below.

Infinite-population models

- The probability distribution of the second generation then becomes

Probability of getting a chromosome with j ones, assuming Onemax fitness

$$p_2(j) = \frac{j p_1(j)}{\sum_{j=0}^{m} j p_1(j)} = \frac{j p_1(j)}{\overline{F_1}} = 2^{1-m} \frac{j}{m} \binom{m}{j}$$

- In principle, one can proceed analytically to compute the probability distribution in generations 3, 4, 5, … (but the equations soon become very messy, see below).

Infinite-population models

- In general, sums of the form $\sum_{j=0}^{m} j^q \binom{m}{j}$ (for some positive integer q) can be computed by starting from the binomial theorem:

$$(a+b)^m = \sum_{j=0}^{m} a^j b^{m-j} \binom{m}{j}$$

- Setting a = x and b = 1, one obtains

$$(x+1)^m = \sum_{j=0}^{m} x^j \binom{m}{j}$$

Infinite-population models

- In general, sums of the form $\sum_{j=0}^{m} j^q \binom{m}{j}$ (for some positive integer q) can be computed by starting from the binomial theorem:

$$(a+b)^m = \sum_{j=0}^{m} a^j b^{m-j} \binom{m}{j}$$

- Setting a = x and b = 1, one obtains

$$(x+1)^m = \sum_{j=0}^{m} x^j \binom{m}{j}$$

Infinite-population models

- Taking the derivative with respect to x and then multiplying by x, one gets

$$xm(x+1)^{m-1} = \sum_{j=0}^{m} j x^j \binom{m}{j} \qquad \text{(Equation A)}$$

- Setting x = 1 one then obtains Eq. (B17).

Infinite-population models

- Taking the derivative with respect to x and then multiplying by x, one gets

$$xm(x+1)^{m-1} = \sum_{j=0}^{m} j x^{j} \binom{m}{j} \qquad \text{(Equation A)}$$

- Setting x = 1 one then obtains Eq. (B17).

Infinite-population models

- If instead of setting x = 1, one again takes the derivative of Equation A (previous slide) with respect to x, one obtains

$$m(x+1)^{m-1}+xm(m-1)(x+1)^{m-2}=\sum_{j=0}^{m} j^2 x^{j-1}\binom{m}{j}$$

- Multiplying by

$$xm(x+1)^{m-1}+x^2m(m-1)(x+1)^{m-2}=\sum_{j=0}^{m} j^2 x^{j}\binom{m}{j}$$

- Finally, with x = 1, one obtains Eq. (B18):

$$m2^{m-1}+m(m-1)2^{m-2}=m(m+1)2^{m-2}=\sum_{j=0}^{m} j^2\binom{m}{j}$$

Infinite-population models

- If instead of setting x = 1, one again takes the derivative of Equation A (previous slide) with respect to x, one obtains

$$m(x+1)^{m-1}+xm(m-1)(x+1)^{m-2}=\sum_{j=0}^{m} j^2 x^{j-1}\binom{m}{j}$$

- Multiplying by

$$xm(x+1)^{m-1}+x^2m(m-1)(x+1)^{m-2}=\sum_{j=0}^{m} j^2 x^{j}\binom{m}{j}$$

- Finally, with x = 1, one obtains Eq. (B18):

$$m2^{m-1}+m(m-1)2^{m-2}=m(m+1)2^{m-2}=\sum_{j=0}^{m} j^2\binom{m}{j}$$

Infinite-population models

- If instead of setting x = 1, one again takes the derivative of Equation A (previous slide) with respect to x, one obtains

$$m(x+1)^{m-1}+xm(m-1)(x+1)^{m-2}=\sum_{j=0}^{m} j^2 x^{j-1}\binom{m}{j}$$

- Multiplying by

$$xm(x+1)^{m-1}+x^2m(m-1)(x+1)^{m-2}=\sum_{j=0}^{m} j^2 x^{j}\binom{m}{j}$$

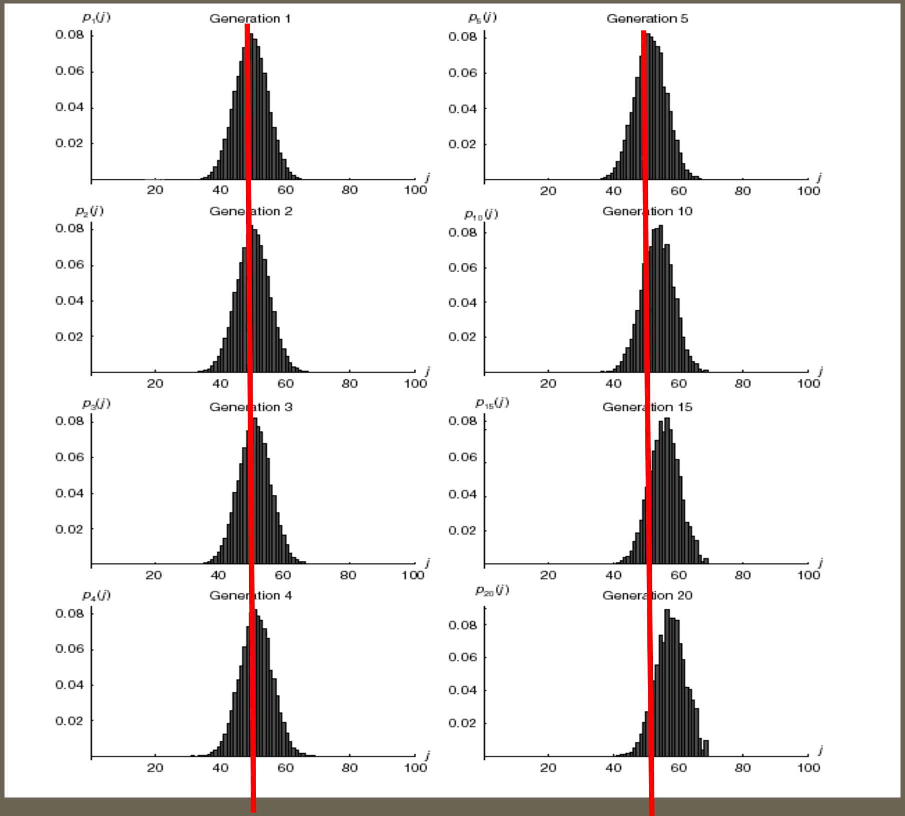
- Finally, with x = 1, one obtains Eq. (B18):

$$m2^{m-1}+m(m-1)2^{m-2}=m(m+1)2^{m-2}=\sum_{j=0}^{m} j^2\binom{m}{j}$$

Infinite-population models

- Unfortunately there is no simple, closed-form expression for $\sum_{j=0}^{m} j^q \binom{m}{j}$.
- Thus, for q > 2 (which is needed in order to compute, say, the average fitness in the qth generation), one has to proceed iteratively: Without setting x =1, again taking the derivative (of the second equation on the previous slide) with respect to x, then multiplying by x, then setting x = 1 etc.).

Infinite population models



Infinite population models

- Thus, one can get an exact description of the evolution of the probability distribution $p_q(j)$.
- So far only selection. Crossover is difficult to treat analytically. One can treat mutation, though, in a simplified way:
- Consider a GA, applied to the Onemax function where, with probability p_μ, *exactly* one gene mutates.

Infinite population models

- In that case, one finds (Appendix B2.3.5):

$$p_2(j) = 2^{1-m}\left(\frac{j}{m} + p_\mu \frac{m-2j}{m^2}\right)\binom{m}{j}$$

- Here, selection has a positive effect for j > m/2, whereas mutation has a negative (immediate) effect.
- At some point the effects balance each other out .

Today’s learning goals

- After this lecture you should be able to
  - Derive and explain the schema theorem, and its implications ✓
  - Derive expressions for the result of selection and mutation in infinite-population models of GAs. ✓
  - Derive expressions for the expected running time for a simple GA.
  - Derive the optimal mutation rate for a simple GA.
  - Explain the concept of premature convergence
  - List methods for avoiding premature convergence

Expected running time

- Consider a very simple GA with a single individual, which is modified by mutation only, such that the new individual is kept if it is better than the old one.
- Let $p_{\mathrm{mut}} = k/m$, where $\mathrm{k} \ll m$.
- Apply this GA to the Onemax problem.
- The expected running time (number of evaluations L) can then be estimated (see the following slides)

Expected running time

- Consider a very simple GA with a single individual, which is modified by mutation only, such that the new individual is kept if it is better than the old one.
- Let $p_{\text{mut}} = k/m$, where $\mathrm{k} \ll m$.
- Apply this GA to the Onemax problem.
- The expected running time (number of evaluations L) can then be estimated (see the following slides)

Expected running time

- Consider a very simple GA with a single individual, which is modified by mutation only, such that the new individual is kept if it is better than the old one.
- Let $p_{\mathrm{mut}} = k/m$, where $\mathrm{k} \ll m$.
- Apply this GA to the Onemax problem.
- The expected running time (number of evaluations L) can then be estimated (see the following slides)

Expected running time

- Let l the number of 0s in a chromosome (i.e. $l = m - j$, where j is the number of 1s.
- Let $P(l, p_{\mathrm{mut}})$ denote the probability of improving a chromosome (= getting more 1s).

Expected running time

- Let l the number of 0s in a chromosome (i.e. $l = m - j$, where j is the number of 1s.
- Let $P(l, p_{\mathrm{mut}})$ denote the probability of improving a chromosome (= getting more 1s).
- $P(l, p_{\mathrm{mut}})$ can be approximated as

$$P(l, p_{\mathrm{mut}}) = (1 - p_{\mathrm{mut}})^{m-l}(1 - (1 - p_{\mathrm{mut}})^{l}),$$

Expected running time

- Let l the number of 0s in a chromosome (i.e. $l = m - j$, where j is the number of 1s.
- Let $P(l, p_{\text{mut}})$ denote the probability of improving a chromosome (= getting more 1s).
- $P(l, p_{\text{mut}})$ can be approximated as

$$P(l, p_{\text{mut}}) = \underbrace{(1 - p_{\text{mut}})^{m-l}}_{\text{Probability of \textit{not} mutating any 1s}}(1 - (1 - p_{\text{mut}})^{l}),$$

Expected running time

- Let l the number of 0s in a chromosome (i.e. $l = m - j$, where j is the number of 1s.
- Let $P(l, p_{\text{mut}})$ denote the probability of improving a chromosome (= getting more 1s).
- $P(l, p_{\text{mut}})$ can be approximated as

$$P(l, p_{\text{mut}}) = (1 - p_{\text{mut}})^{m-l}(1 - \underbrace{(1 - p_{\text{mut}})^{l}}_{\text{Probability of \textit{not} mutating any 0s.}}),$$

Expected running time

- Let l the number of 0s in a chromosome (i.e. $l = m - j$, where j is the number of 1s.
- Let $P(l, p_{\text{mut}})$ denote the probability of improving a chromosome (= getting more 1s).
- $P(l, p_{\text{mut}})$ can be approximated as

$$P(l, p_{\text{mut}}) = (1 - p_{\text{mut}})^{m-l}\underbrace{(1 - (1 - p_{\text{mut}})^{l})}_{\text{Probability of mutating at least one 0.}},$$

Expected running time

- With $p_{\mathrm{mut}} = k/m$ (see above), the expected number of evaluations $E(\Delta L(l, p_{\mathrm{mut}})) \equiv E(\Delta L(l, k/m))$ becomes

$$E(\Delta L(l, k/m)) = \frac{1}{P\left(l, \frac{k}{m}\right)}$$

- Assuming random initialization, the first individual will have around m/2 0s. The expected number of evaluations E(L) to obtains a chromosome with only 1s then becomes
- $E(L) = E\left(\Delta L\left(\frac{m}{2}, \frac{k}{m}\right)\right) + E\left(\Delta L\left(\frac{m}{2} - 1, \frac{k}{m}\right)\right) + \ldots + E\left(\Delta L\left(1, \frac{k}{m}\right)\right)$

Expected running time

- With $p_{\text{mut}} = k/m$ (see above), the expected number of evaluations $E(\Delta L(l, p_{\text{mut}})) \equiv E(\Delta L(l, k/m))$ becomes

$$E(\Delta L(l, k/m)) = \frac{1}{P\left(l, \frac{k}{m}\right)}$$

- Assuming random initialization, the first individual will have around m/2 0s. The expected number of evaluations E(L) to obtains a chromosome with only 1s then becomes
- $E(L) = E\left(\Delta L\left(\frac{m}{2}, \frac{k}{m}\right)\right) + E\left(\Delta L\left(\frac{m}{2} - 1, \frac{k}{m}\right)\right) + \ldots + E\left(\Delta L\left(1, \frac{k}{m}\right)\right)$

Expected running time

- Finally using the assumption $k \ll m$, we get

$$P(l,k/m) \equiv \left(1-\frac{k}{m}\right)^{m-l}\left(1-\left(1-\frac{k}{m}\right)^{l}\right)$$

$$\approx \left(1-\frac{k}{m}\right)^{m-l}\frac{lk}{m} \rightarrow e^{-k}lk/m$$

using $(1-x)^{a} \approx 1-ax$

Expected running time

- Finally using the assumption $k \ll m$, we get

$$P(l, k/m\,) \equiv \left(1 - \frac{k}{m}\right)^{m-l} \left(1 - \left(1 - \frac{k}{m}\right)^{l}\right)$$

$$\approx \left(1 - \frac{k}{m}\right)^{m-l} \frac{lk}{m} \rightarrow e^{-k} lk/m$$

using $(1 - k/m)^{m} \rightarrow\ e^{-k}$ for large m

Expected running time

- Thus, reversing the order of summation, we get

$$E(L) = e^k \frac{m}{k} \sum_{l=1}^{m/2} 1/l \approx e^k \frac{m}{k} \ln \frac{m}{2}$$

- ...which is the expected running time.
- The approximation is very good from small l, which is where it matters (=where improvements take longest time), so the values of $E(L)$ are very close to values found in numerical simulations (at least if $k \ll m$).
- See also pp. 181-182 in the course book.

Expected running time

- Thus, reversing the order of summation, we get

$$E(L) = e^k \frac{m}{k} \sum_{l=1}^{m/2} 1/l \ \approx \ e^k \frac{m}{k} \ln \frac{m}{2}$$

- ...which is the expected running time.
- The approximation is very good from small l, which is where it matters (=where improvements take longest time), so the values of $E(L)$ are very close to values found in numerical simulations (at least if $k \ll m$).
- See also pp. 181-182 in the course book.

Today's learning goals

- After this lecture you should be able to
  - Derive and explain the schema theorem, and its implications ✓
  - Derive expressions for the result of selection and mutation in infinite-population models of GAs. ✓
  - Derive expressions for the expected running time for a simple GA. ✓
  - Derive the optimal mutation rate for a simple GA.
  - Explain the concept of premature convergence
  - List methods for avoiding premature convergence

Optimal mutation rate

- Consider the same simple GA as in the runtime computation, and the same (Onemax) problem.
- Using the equation for the probability of an improvement:

$$P(l, p_{\text{mut}}) = (1 - p_{\text{mut}})^{m-l}(1 - (1 - p_{\text{mut}})^{l}),$$

… one obtains (see pp. 182-183) $p^{*}{}_{\text{mut}} = \frac{1}{m}$.

- This mutation rate (one or a few times 1/*m*) typically works well for most fitness functions (with binary chromosomes).

Optimal mutation rate

- Consider the same simple GA as in the runtime computation, and the same (Onemax) problem.
- Using the equation for the probability of an improvement:

$$P(l, p_{\mathrm{mut}}) = (1 - p_{\mathrm{mut}})^{m-l}(1 - (1 - p_{\mathrm{mut}})^{l}),$$

… one obtains (see pp. 182-183) $p^{*}{}_{\mathrm{mut}} = \frac{1}{m}$.

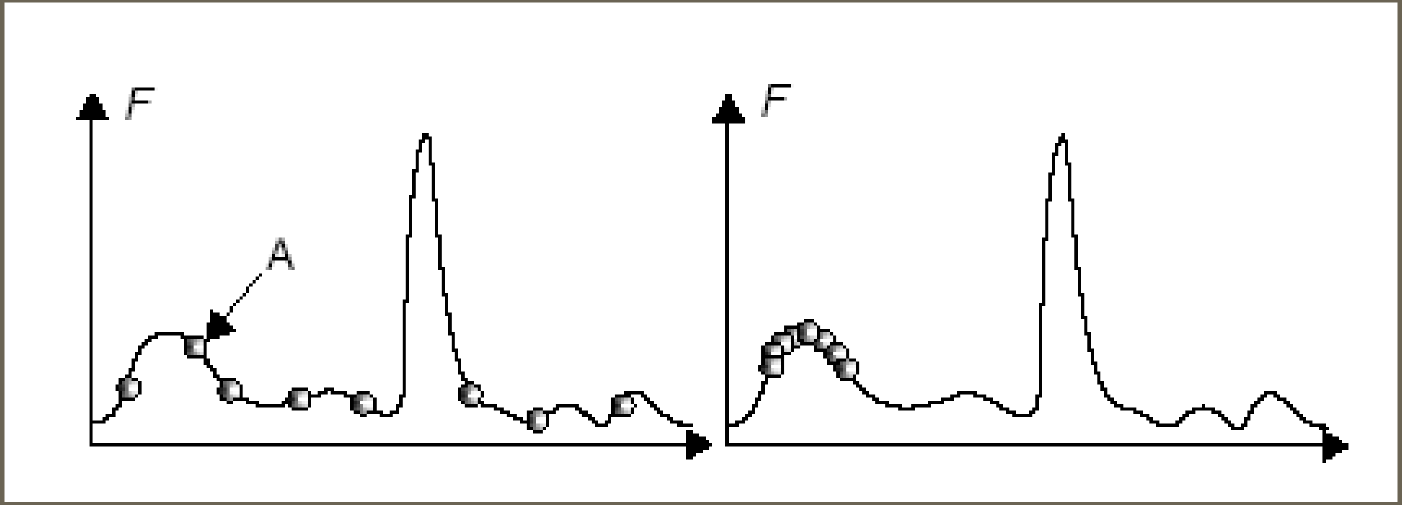
- This mutation rate (one or a few times 1/*m*) typically works well for most fitness functions (with binary chromosomes).

Today’s learning goals

- After this lecture you should be able to
  - Derive and explain the schema theorem, and its implications ✓
  - Derive expressions for the result of selection and mutation in infinite-population models of GAs. ✓
  - Derive expressions for the expected running time for a simple GA. ✓
  - Derive the optimal mutation rate for a simple GA. ✓
  - Explain the concept of premature convergence
  - List methods for avoiding premature convergence

Premature convergence

- Since GAs are very efficient in their search for an optimum, they may get stuck at a local optimum, a phenomenon known as **premature convergence**:



Today's learning goals

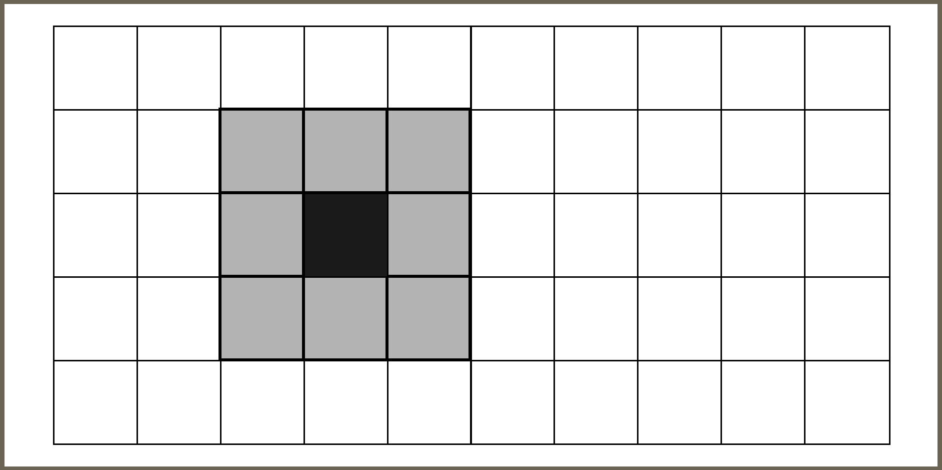
- After this lecture you should be able to
  - Derive and explain the schema theorem, and its implications ✓
  - Derive expressions for the result of selection and mutation in infinite-population models of GAs. ✓
  - Derive expressions for the expected running time for a simple GA. ✓
  - Derive the optimal mutation rate for a simple GA. ✓
  - Explain the concept of premature convergence ✓
  - List methods for avoiding premature convergence

Premature convergence

- Premature convergence can be avoided in many different ways:
  - With fitness ranking (if roulette-wheel selection is used, not needed if tournament selection is used!)
  - Reducing the crossover probability,
  - Using *varying* mutation rates (see pp. 69-71; Fig. 3.15),
  - Introducing mating restrictions (e.g. diffusion models).

Diffusion models

- Place the individuals on an imaginary grid.
- For any selected individual, allow mating only with one of its neighbors:



Premature convergence

- However, an alternative approach is simply to restart the GA, with a different random number sequence.
- It is often a good idea to make a few short trial runs to find good parameter settings; see e.g. Tables 3.1-3.3.

Today’s learning goals

- After this lecture you should be able to
  - Derive and explain the schema theorem, and its implications ✓
  - Derive expressions for the result of selection and mutation in infinite-population models of GAs. ✓
  - Derive expressions for the expected running time for a simple GA. ✓
  - Derive the optimal mutation rate for a simple GA. ✓
  - Explain the concept of premature convergence ✓
  - List methods for avoiding premature convergence ✓