OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

INF FAKULTÄT FÜR INFORMATIK

Evolutionary Algorithms

Encoding, Fitness, Selection

Prof. Dr. Rudolf Kruse **Pascal Held**

{kruse,pheld}@iws.cs.uni-magdeburg.de

Otto-von-Guericke-Universität Magdeburg

Fakultät für Informatik

Institut für Wissens- und Sprachverarbeitung

Outline

Desirable properties of an encoding

now: deeper investigation of the different elements of an EA

At first: **encoding of a solution candidate**

- encoding has to be chosen problem-specific
- no general „recipe“ to find a good encoding
- but: some principes which should be taken into consideration

Desirable properties of an encoding:

- Representation of similar phenotypes by similar genotype
- Similar fitness on similar candidates
- Closure on Ω under the used evolutionary operators

Encoding: first desirable trait

Similar phenotypes should be represented by similar genotypes

- mutations of certain genes result in similar genotypes (particular changes of allels $\rightarrow$ small change of the chromosome)
- if trait is not satisfied, obvious changes cannot be generated in some cases
- consequence: huge change of the genotype to end up in a similar (and perhaps better) phenotype

Demonstration example:

- Optimization of a real function $y = f(x_1, \ldots, x_n)$
- Representation of the (real) arguments by binary codes
- Problem: binary representation leads to „Hamming-Cliffs“

Binary coding of real numbers

given: real interval $[a, b]$ and coding precision ε

desired: coding rule for $x \in [a, b]$ as binary number z
so that devation of z and x is lower than ε

Idea: divide $[a, b]$ in equidistant sections of length $\leq \varepsilon$

$\Rightarrow$ 2^k sections with $k = \left\lceil \log_2 \frac{b-a}{\varepsilon} \right\rceil$
coded by $0, \dots, 2^k - 1$

Binary coding of real numbers

Sections: $k = \left\lceil \log_2 \frac{b-a}{\varepsilon} \right\rceil$ or $k = \left\lceil \log_2 \frac{b-a}{2\varepsilon} \right\rceil$

Coding: $z = \left\lfloor \frac{x-a}{b-a}(2^k - 1) \right\rfloor$ or $z = \left\lfloor \frac{x-a}{b-a}(2^k - 1) + \frac{1}{2} \right\rfloor$

Decoding: $x = a + z \cdot \frac{b-a}{2^k-1}$

Example: intervall $[-1, 2]$, precision $\varepsilon = 10^{-6}$, $x = 0.637197$

$$k = \left\lceil \log_2 \frac{2-(-1)}{10^{-6}} \right\rceil = \left\lceil \log_2 3 \cdot 10^6 \right\rceil = 22$$

$$z = \left\lfloor \frac{0.637197-(-1)}{2-(-1)}(2^{22} - 1) \right\rfloor = 2288966_{10}$$

$$= 1000101110110101000110_2$$

Hamming-Cliffs

Problem:

- adjacent numbers can be coded very differently
- encodings have big Hamming-distance (# different Bits)
- Mutations/Crossover overcome „Hamming-Cliffs“ very hardly

Example:

- Representation of the numbers from 0 til 1 by 4-Bit-Numbers
- also mapping $\frac{k}{15} \to k$

$\Rightarrow$ $\frac{7}{15}$ (0111) and $\frac{8}{15}$ (1000) have the same Hamming-distance 4

Gray-Codes: Avoidance of Hamming-Cliffs

Solution: Gray-Codes
Hamming-distance of adjacent numbers = 1 Bit

binär	Gray
0000	0000
0001	0001
0010	0011
0011	0010

binär	Gray
0100	0110
0101	0111
0110	0101
0111	0100

binär	Gray
1000	1100
1001	1101
1010	1111
1011	1110

binär	Gray
1100	1010
1101	1011
1110	1001
1111	1000

Gray-Codes: Computation

- **Gray-Codes are not unique**
- each code where encodings of adjacent numbers differ in only 1 Bit is called Gray-Code
- Computation of Gray-Codes is usually started from binray number enconding

Most frequent form:

$$\textit{Encoding:} \quad g = z \oplus \left\lfloor \frac{z}{2} \right\rfloor$$

$$\textit{Decoding:} \quad z = \bigoplus_{i=0}^{k-1} \left\lfloor \frac{g}{2^i} \right\rfloor$$

$\oplus$: Exclusive-Or of the binary representation

Gray-Codes: Computation

Example: interval $[-1, 2]$, precision $\varepsilon = 10^{-6}$, $x = 0.637197$

$$\begin{aligned} z &= \left\lfloor \frac{0.637197 - (-1)}{2 - (-1)} (2^{22} - 1) \right\rfloor = 2288966_{10} \\ &= 1000101110110101000110_2 \end{aligned}$$

$$\begin{aligned} g &= 1000101110110101000110_2 \\ &\oplus \;\; 100010111011010100011_2 \\ &= 1100111001101111100101_2 \end{aligned}$$

Gray-Codes: Implementation

```
unsigned int num2gray (unsigned int x)
{                                /* --- convert number to Gray code */
  return x ^ (x >> 1);           /* exclusive or with shifted input */
}  /* num2gray() */
```

```
unsigned int gray2num (unsigned int x)
{                                /* --- convert Gray code to number */
  unsigned int y = x;            /* copy the input number and */
  while (x >>= 1) y ^= x;        /* do an exclusive or with all */
  return y;                      /* shift downs of the input */
}  /* gray2num() */              /* and return the result */
```

```
unsigned int gray2num (unsigned int x)
{                                /* --- convert Gray code to number */
  x ^= x >> 16; x ^= x >> 8;     /* do an exclusive or with all */
  x ^= x >>  4; x ^= x >> 2;     /* shift downs of the input */
  return x ^ (x >> 1);           /* and return the result */
}  /* gray2num() */              /* (32 bit integers only) */
```

Encoding: second desirable trait

Similarly encoded candidate solutions should have a similar fitness

Problem of the epistasis:

- *in biology:* one allele of a (so-called epistatic) gene suppresses the effect of all possible alleles of another gene
- *in evolutionary algorithms:*
  interaction between genes of a chromosome,
  Changes of the fitness by modifying one gene strongly depends on the value(s) of (an)other gene(s)

Epistasis in biology

Deviations from Mendelian laws are grounded on epistasis.

- Crossing of homozygous black and white seed beans:
  \# black : white- : brown seed beans = 12:1:3 (2nd child gen.)

$\Rightarrow$ Contradiction to the Mendelian laws

Example: The Traveling Salesman Problem

Find a round trip of n cities with respect to minimal costs

two different encodings of the round trip:

1. Permutation of the cities
   - visit city at the k-th position in the k-th step
   - *low epistasis:* z.B. swapping two cities alters fitness(costs) by comparable amounts (only local changes)

2. Specification of a list of numbers that state the position of the next city to be visited in a (sorted) list from which all already visited cities have been deleted
   - *high epistasis:* Modifying a single gene (esp. the closer to the top) may alter the complete round trip (global tour-change)

$\Rightarrow$ leads mostly to large changes of the fitness

Second encoding: Effect of a Mutation

Mutation	Chromosome	Remaining cities	Round trip
before	5 3 3 2 2 1	1, 2, 3, 4, **5**, 6 1, 2, **3**, 4, 6 1, 2, **4**, 6 1, **2**, 6 1, **6** **1**	5 3 4 2 6 1
after	**1** 3 3 2 2 1	**1**, 2, 3, 4, 5, 6 2, 3, **4**, 5, 6 2, 3, **5**, 6 2, **3**, 6 2, **6** **2**	1 4 5 3 6 2

Epistasis: summary

- high epistatic encoding: no regularities

$\Rightarrow$ Mutation/Crossover leads to almost random fitness changes

$\Rightarrow$ optimization problem is very hard to solve by EAs

- very low epistatic encoding: other methods often more successful

- [Davidor, 1990] tried to classify optimization problems as *easy or hard to solve by an EA* based on the notion of epistasis $\Rightarrow$ failure
- since: epistasis = property of the encoding, **not** of the problem itself
- $\exists$ encodings of a problem with higher and lower epistasis
- $\exists$ problems with low epistatic encoding: too hard to solve by an EA

Encoding: 3^{rd} desirable trait

If possible, the search space Ω should be closed under the used genetic operators.

In general: **Space is left**, if

- new chromosome cannot be meaningfully interpreted or decoded
- a candidate solution does not fulfill certain basic requirements
- a candidate solution is evaluated incorrectly by the fitness function

Problem of **coordination** of encoding and EA-operators:

- choose or design encoding-specific genetic operators
- use mechanisms to „repair" chromosomes
- introduce a penalty term that reduces the fitness of such individuals $\notin \Omega$

Leaving the space: example

n-Queens-Problem

Two different encodings: chromosome of length n

1. File positions of queens per rank (alleles $0, \dots, n-1$)
   Operators: One-point Crossover, standard mutation
   generates always valid vectors of file position
   $\Rightarrow$ search space is not left

2. Numbers of the field (alleles $0, \dots, n^2-1$) of the queens
   Operators: One-point crossover, standard mutation
   generates chromosomes with more than one queen per field
   $\Rightarrow$ search space is left

Leaving the space: solving approachs

n-Queens-Problem

- **Use other encoding:** first encoding avoids problem and Ω is considerably smaller (if feasible, best method!)
- **Encoding-specific evolutionary operators:**
  - *Mutation:* Excluding of already existing alleles on random
  - *Crossover:* look at first for field numbers of each chromosome which are not contained in other chromosomes and apply one-point crossover on shortened chromosomes
- **Repair mechanisms:** find und replace multiple occuring field numbers until all field numbers are distinct
- **Penalty term:** reduce fitness by amount of multiple allocations of fields multiplied with weight if necessary

Leaving space using the example of TSP

- Representation of the round trip by permutation of the cities (city at k-th position is visited in the k-th step.)
- one-point crossover can exceed the space of permutations

3	5	2	8	1	7	6	4
1	2	3	4	5	6	7	8

3	5	2	4	5	6	7	8
1	2	3	8	1	7	6	4

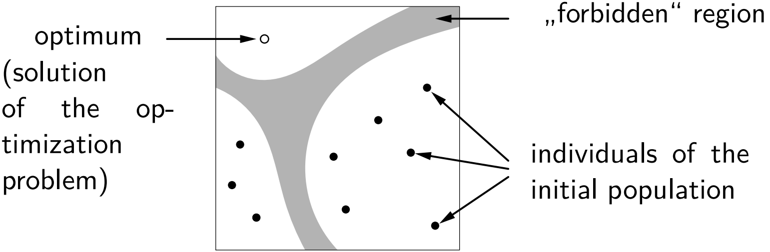
- **Encoding-specific evolutionary operators:**
  - *Mutation:* e.g. pair swaps, shift/cyclic permutation, inversion
  - *Crossover:* edge recombination (will be discussed later)
- **Repair mechanisms:** remove twice occuring cities and append the missing cities at the end:

3	5	2	4	~~5~~	6	7	8	1

- **Penalty term:** reduce fitness by value c for each missing city

Leaving the space

- if Ω is *not connected*, *repair mech.* can complicate the search
- immediately restoring of „forbidden“ $x \in \Omega$ in permitted regions



- in such cases: introduce **penalty term**
- $x \in \Omega$ in „forbidden“ region is penalized but not removed
- penalty term should be increased on time: suppresses $x \in \Omega$ in „forbidden“ regions in following generations

Outline

Principe of selection

- better individuals (better fitness) should have better chances to create offspring(differential reproduction)
- **Selective pressure:** Strength of preferencing good individuals
- Choice of selective pressure: Contrast of
  Exploration of the space:
  - deviation of the individuals over Ω as wide as possible
  - preferably big chances to find global optimum

  $\Rightarrow$ smaller selective pressure is desired

  Exploitation (of good individuals):
  - Strive for (perhaps local) optimum in the vicinity of good individuals
  - Convergence to optimum

  $\Rightarrow$ higher selective pressure is preferred

Comparison of selection methods

Comparison of methods for created selective pressure by metrics

- *time to takeover:* # generations until population converges (population is called *converged*, if all individuals are identical)
- **selection intensity:** decides by selection differential between average quality before and after the selection

Selection intensity

according to [Weicker, 2007]

Definition (Selection intensity)

Let $(\Omega, f, \succ)$ be a considered optimization problem and a selection operator $\text{Sel}^{\xi} : (\mathcal{G} \times \mathcal{Z} \times \mathbb{R})^r \to (\mathcal{G} \times \mathcal{Z} \times \mathbb{R})^s$ is applied on a population P with an average quality μ_f and standard deviation σ_f. Then, let μ_f^{sel} be the average quality of the population P_{sel} and the selection operator has the *selection intensity*

$$I_{\text{sel}} = \begin{cases} \frac{\mu_f^{\text{sel}} - \mu_f}{\sigma_f} & \text{falls } \succ = >, \\ \frac{\mu_f - \mu_f^{\text{sel}}}{\sigma_f} & \text{sonst.} \end{cases}$$

Selection intensity

the higher I_{sel}, the higher the selective pressure

Example:

- 10 Indiv. with fitness: 2.0, 2.1, 3.0, 4.0, 4.3, 4.4, 4.5, 4.9, 5.5, 6.0
- selection leads to individuals with quality: 2.0, 3.0, 4.0, 4.4, 5.5

$$\mu_f = \frac{1}{|P|}\sum_{i=1}^{|P|} A^{(i)}.F \qquad \text{(Average of the fitness)}$$

$$\sigma_f = \sqrt{\frac{1}{|P|-1}\sum_{i=1}^{|P|}(A^{(i)}.F - \mu_f)^2} \qquad \text{(standard deviation)}$$

$$\Rightarrow \quad \mu_f = 4.07, \quad \sigma_f = 1.27, \quad \mu_f^{\text{sel}} = 3.78, \quad I_{\text{sel}} = \frac{4.07-3.78}{1.27} = 0.228$$

Criticism on selection intensity:

- metric requires a standard normal distribution of values
- rarely applicable on general optimization problems

Choice of the selective pressure

- **best strategy:** time-dependent selective pressure
  low selective pressure in prior generations
  higher selective pressure in later generations

$\Rightarrow$ at first good exploration of the space, then exploitation of the promising region

- regulation of the selective pressure by adapting the fitness function or by the parameter of selection method

- important **selection methods:**
  Roulette-wheel Selection, Rank-based Selection, Tournament Selection
- important **adaption methods:**
  Adaption of the variation of the fitness, linear dynamical scaling, σ-scaling

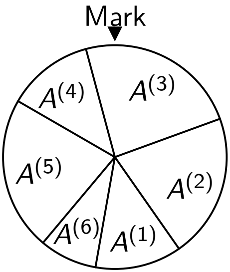
Roulette wheel selection (dt. Glücksradauswahl)

- best known selection method
- compute the relative fitness of the individuals $A^{(i)}$, $1 \leq i \leq |P|$

$$f_{\text{rel}}\left(A^{(i)}\right) = \frac{A^{(i)}.F}{\sum_{j=1}^{|P|} A^{(j)}.F}$$

  and interprete $f_{\text{rel}}\left(A^{(i)}\right)$ as a probability to be selected
  (so called **fitness-proportionate Selection**)
- **Please note:** absolute fitness $A.F$ may not be negative
- **Attention:** fitness has to be maximized
  (otherwise: selection of bad individuals with high probability)
- **Demonstration:** Roulette-wheel with 1 sector per individuual $A^{(i)}$,
  sector size = relative fitness values $f_{\text{rel}}\left(A^{(i)}\right)$

Roulette-wheel selection: Demonstration



Selection of an individual:

1. set the roulette-wheel into motion
2. choose the ind. of the corresponding sector

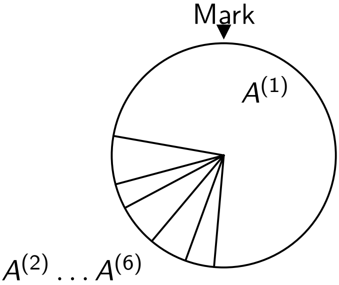
Selection of the next population:

- repeat selection # individuals-times

Disadvantage: Calculation of the relative fitness by summing up all fitness values (normalization factor)

- constant initial population during the selection
- aggravated parallelization of the implementation

Roulette-wheel selection: Dominance Problem



- individual with a very high fitness may **dominate** the selection
- Due to many copies/very similar individuals: dominance may become even stronger in subsequent generations

$\Rightarrow$ **Crowding:** population of very similar/identical individuals

- results in a very fast find of the (local) optimum
- **Disadvantage:** diversity of the population vanishes
  - Exploitation of worse individuals
  - No exploration of the space but local optimization (preferred in later generations, undesirable at the beginning)

Roulette-wheel selection: selection intensity

Satz

When using a simple fitness-proportionate selection in a population with average quality μ_f and variation of the quality σ_f^2, the selection intensity is

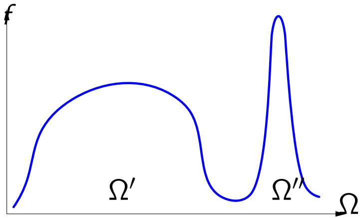
$$I_{sel} = \frac{\sigma_f}{\mu_f}.$$

- Proof in exercise sheet

Fitness function: Premature convergence

Dominance problem illustrates the strong influence of the fitness function on the effect of the fitness-proportionate selection

- Problem of **premature convergence:** If (value) range of the maximizing function is very huge
- Example: no chromosome at the beginning in the section Ω'' $\rightarrow$ population remains by selection in the vicinity of the local maximum in the section Ω'



Individuals which converge to the section between Ω' and Ω'' have worse chances to create offspring

Vanishing selective pressure

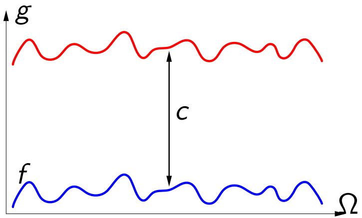
Problem of the **absolute height** of the fitness values i.c. to the **variation**

or: Problem of the **vanishing selective pressure:**

- Maximizing of $f : \Omega \to \mathbb{R}$ is equivalent to the maximization of $g : \Omega \to \mathbb{R}$ with $g(x) \equiv f(x) + c, \; c \in \mathbb{R}$

$$c \gg \sup_{x \in \Omega} f(x) \Longrightarrow \forall x \in \Omega : g_{\text{rel}}(x) \approx \frac{1}{|P|} \quad (\text{Pop.-größe}|P|)$$

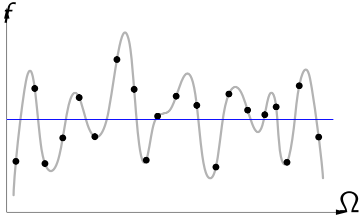
$\Rightarrow$ (too) small selective pressure



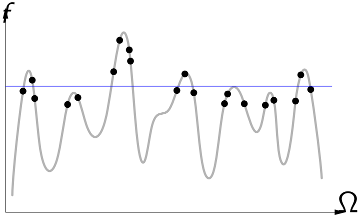
- although maxima correspond: with EA differently to find
- with g: only (too) small differences of the f_{rel}

Vanishing selective pressure

- Problem is perhaps grounded on the EA itself
- it increases tendentially the (average) fitness of the individuals
- higher selective pressure at the beginning due to random fitness values
- later: smaller selective pressure (inverse way is preferred)
- Example: points illustrate individuals of the generation



early generation



later generation

Adapting of the fitness function

Approach: **Scaling of the fitness**

linear dynamical scaling:

$$f_{\text{lds}}(A) = \alpha \cdot A.F - \min\left\{A^{(i)}.F \mid P(t) = \left\{A^{(1)}, \ldots, A^{(r)}\right\}\right\}, \quad \alpha > 0$$

- instead minimum of $P(t)$, minimum of the last k generations can be used
- usually $\alpha > 1$

σ-Scaling:

$$f_\sigma(A) = A.F - (\mu_f(t) - \beta \cdot \sigma_f(t)), \quad \beta > 0$$

- Problem: Choice of the parameter α and β

Adapting of the fitness function

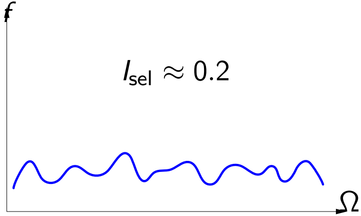
- consider **variation coefficient** of the fitness function

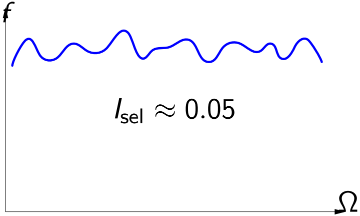
$$v = \frac{\sigma_f}{\mu_f} = \frac{\sqrt{\frac{1}{|\Omega|-1} \sum_{x' \in \Omega} \left(f(x') - \frac{1}{|\Omega|} \sum_{x \in \Omega} f(x) \right)^2}}{\frac{1}{|\Omega|} \sum_{x \in \Omega} f(x)}, \quad v(t) = \frac{\sigma_f(t)}{\mu_f(t)}$$

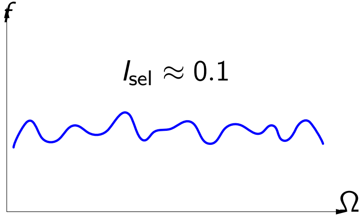
- empirical discovery: $v \approx 0.1$ yields good ratio of exploration and exploitation
- if $v \neq 0.1$, then adapting of f (e.g. by scaling)
- v is not calculat-, but estimable
- practical calculations of v: Replace of Ω by $P(t)$
- hence: approximation of v by **selection intensity** $I_{sel}(t)$

$\Rightarrow$ in each generation: calculate $I_{sel}(t)$ and adapt f accordingly (σ-scaling with $\beta = \frac{1}{I^*_{sel}}$, $I^*_{sel} = 0.1$)

Illustration of the selection intensity







- too high I_{sel}: premature convergence
- too small I_{sel}: vanishing selective pressure
- appropriate: $I_{sel} \approx 0.1$

Adaption of the fitness function: dependence on time

- determine f_{rel} not directly from $f(x)$ but $g(x) \equiv (f(x))^{k(t)}$
- time-dependent exponent $k(t)$ regulates selective pressure
- **Method to determine** $k(t)$ [Michalewicz, 1996]
  (should limit selection intensity I_{sel} in the vicinity of $I^*_{\text{sel}} \approx 0.1$)

$$k(t) = \left(\frac{I^*_{\text{sel}}}{I_{\text{sel}}}\right)^{\beta_1} \left(\tan\left(\frac{t}{T+1} \cdot \frac{\pi}{2}\right)\right)^{\beta_2 \left(\frac{I_{\text{sel}}}{I^*_{\text{sel}}}\right)^{\alpha}}$$

I^*_{sel}, β_1, β_2, α: parameter of the method
I_{sel}: coefficient of variation (e.g. estimated from $P(t=0)$)
T: max. number of remaining generations to be computed
t: current time step (number of generation)

- Recommended: $I^*_{\text{sel}} = 0.1,\ \beta_1 = 0.05,\ \beta_2 = 0.1,\ \alpha = 0.1$

Adaption of the fitness function: Boltzmann-Selection

- determine relative fitness not directly from $f(x)$ but from $g(x) \equiv \exp\left(\frac{f(x)}{kT}\right)$
- time-dependent **temperature** T controls selective pressure
- k is normalizing constant
- Temperature decreases e.g. linearly to the predefined maximum number of generations

Outline

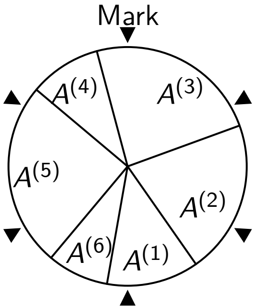
Roulette-wheel Selection: vaiance problem

- Selection of individuals is indeed proportional to the fitness, but random
- no guarantee that „fitter“ individuals are taken to the next generation, not even for the best individual
- gen.: high deviation (**high variance**) of the offspring of an individual
- Computation of the mean value: see the exercise sheet
- very simple but not implicitly a recommendable solution: **Discretization of the fitness range**
    - compute $\mu_f(t)$ and $\sigma_f(t)$ of P
    - if $\mu_f(t) - \sigma_f(t) > f(x)$: 0 offspring
    - if $\mu_f(t) - \sigma_f(t) \leq f(x) \leq \mu_f(t) + \sigma_f(t)$: 1 offspring
    - if $f(x) > \mu_f(t) + \sigma_f(t)$: 2 offsprings

Expected value model: Sol. of the variance problem

- generate $\lfloor f_{\mathrm{rel}}(s) \cdot |P| \rfloor$ individuals for each solution candidate
- fill the (intermediary) population by Roulette-wheel selection

Alternative: **Stochastic Universal Sampling**



Selection of the next population:

- Rotate Roulette-wheel once
- Choose one chromosome per mark
  Here:
  $1 \times A^{(1)},\ 1 \times A^{(2)},\ 2 \times A^{(3)},\ 2 \times A^{(5)}.$
- Better-than-average individuals are taken into the next population definitely

Expected value model: Variants

Create of remaining individuals of the (intermediary) population by

- Method of the **voting evaluation** (mandate/seat apportionment, e.g. largest remainder, Hare-Niemeyer, d'Hondt)

- **Roulette-wheel selection** but:
  - for each individual A with 1 offspring: $A.F' \leftarrow A.F - \Delta f$
  - if $A.F' < 0$, no further offspring of A
  - Principe of choice of Δf: best individual gets at most predefined number k of offspring:

$$\Delta f = \frac{1}{k} \max\{A.F \mid A \in P(t)\}$$

Rank-based Selection

1. Sort individuals decendingly according to their fitness:
   Rank is assigned to each individual in population
   (from statistics: distribution-free techniques, z.B. rank correlation)
2. Define prob. distribution over Rank scale:
   the lower the rank the lower the probability
3. Roulette-wheel selection based on the distribution

Adavantage:

- Avoidance of dominance problem: decoupling of fitness value and selection probability
- regulation of the selective pressure by prob. distribution on rank scale

Disadvantage: Sort of individuals (complexity: $|P| \cdot \log |P|$)

Tournament selection

1. Draw k individuals $(2 \leq k < |P|)$ randomnly from $P(t)$ (draw can be reclined or not, selection *without* regarding the fitness, let k be the **tournament size**).
2. Individuals carry out the tournament and best indivual wins: Tournament winner receives a descendant in the next population
3. *All* participants (even the winner) of the tournament are returned to $P(t)$

Advantage:

- Avoidance of the dominance problem: decoupling of fitness value and selection probability
- regulation of the selective pressure by tournament size with limitations

Modification: f_{rel} of the participants determine winning probability (Roulette-wheel selection of an individual in tournament)

Elitism

- only the expected value model (and some of its variants) ensures that the best individual enters the next generation
- if best individual in next population: no protection from modifications by genetic operators
(even in the expected value model)

$\Rightarrow$ fitness of the best individual can decrease from one generation to the next (= undesired)

Solution: Elitism

- *unchanged* transfer of the best individual (or the k, $1 \leq k < |P|$ best individuals) into the next generation
- *elite* of a population never gets lost, hence *elitism*
- **Attention:** elite is *not* exclued from normal selection: genetic operator can improve them

Elitism

- many times: offspring (Mutation-/Crossover products) replace their parents
- **„local“ elitism** (Elitism between parents and offspring)
  - *Mutation:* mutated individual replaces its parents $\leftrightarrow$ it has at least the same fitness
  - *Crossover:* sort the four involved individuals (2 parents, 2 descendants) according to the fitness, both best individuals $\rightarrow$ next generation
- **Adavantage:** better convergence as the local optimum is intended more consequently
- **Disadvantage:** pretty high risk of getting stuck in local optima as no local degradation is possible

Niche Techniques: Prevention of Crowding

Deterministic Crowding :

- *Idea:* generated offspring should always replace those individuals in the population that are most similar

$\Rightarrow$ local density of individuals in Ω cannot grow so easily

- *requires:* Similarity or distance metric for individuals (on binary coded chromosomes e.g. Hamming-distance)

Variant of deterministic Crowding:

- Crossover: group individuals into two pairs (1 parents, 1 offspring), child is assigned to the parent to which it is more similar
- take the best individual of each pair
- *Advantage:* much fewer similarity computations between individuals (only a part of the Pop. is considered)

Niche Techniques: Sharing

- *Idea:* reduce the fitness of an individual if there are other individuals in its neighborhood
  Intuitively: **individuals share the resources of a niche**
- *requires:* similarity or distance metric for individuals
- *Example:*

$$f_{\text{share}}(A) = \frac{A.F}{\sum_{B \in P(t)} g(d(A, B))}$$

d: Distance metric of the individuals

g: Weighting function, defines both shape and size of the niche, e.g. so called **power law sharing**:

$$g(x) = \begin{cases} 1 - \left(\frac{x}{\varrho}\right)^{\alpha} & \text{if } x < \varrho, \\ 0, & \text{otherwise} \end{cases}$$

ϱ: Niche radius, α: controls the strength of the influence

Characterization of selection methods

static	probability of selection remains constant
dynamic	probability of selection changes
extinguishing	Probabilities of selection may be 0
preservative	All probabilities of selection must be greater than 0
pure-bred	Individuals can only have offspring in one generation
under-bred	Individuals are allowed to have offspring in more than one
rechts	All individuals of a population may reproduce
links	The best individuals of a population may *not* reproduce
generational	The set of parents is fixed until all offspring are created
on the fly	Created offspring directly replace their parents

Referenced literature

Davidor, Y. (1990).
Lamarckian Sub-Goal reward in genetic algorithm.
In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, page 189–194.

Michalewicz, Z. (1996).
Genetic Algorithms + Data Structures = Evolution Programs.
Springer-Verlag, New York, NY, USA, 3rd (extended) edition.

Weicker, K. (2007).
Evolutionäre Algorithmen.
Teubner Verlag, Stuttgart, Germany, 2nd edition.