Metaheuristiky

- **Metaheuristika Simulated Annealing**
- **Metaheuristika Tabu Search**
- **Genetický algoritmus**
  (by Colin R. Reeves:Genetic Algorithms. In: Handbook of Metaheuristics, ed. Gendreau, M., Potvin, J.Y., Springer, New York, Dordrecht, Heidelberg, London, sec. edition, 2010, 648p.)
- **Metaheuristika Scatter Search**

Genetický algoritmus

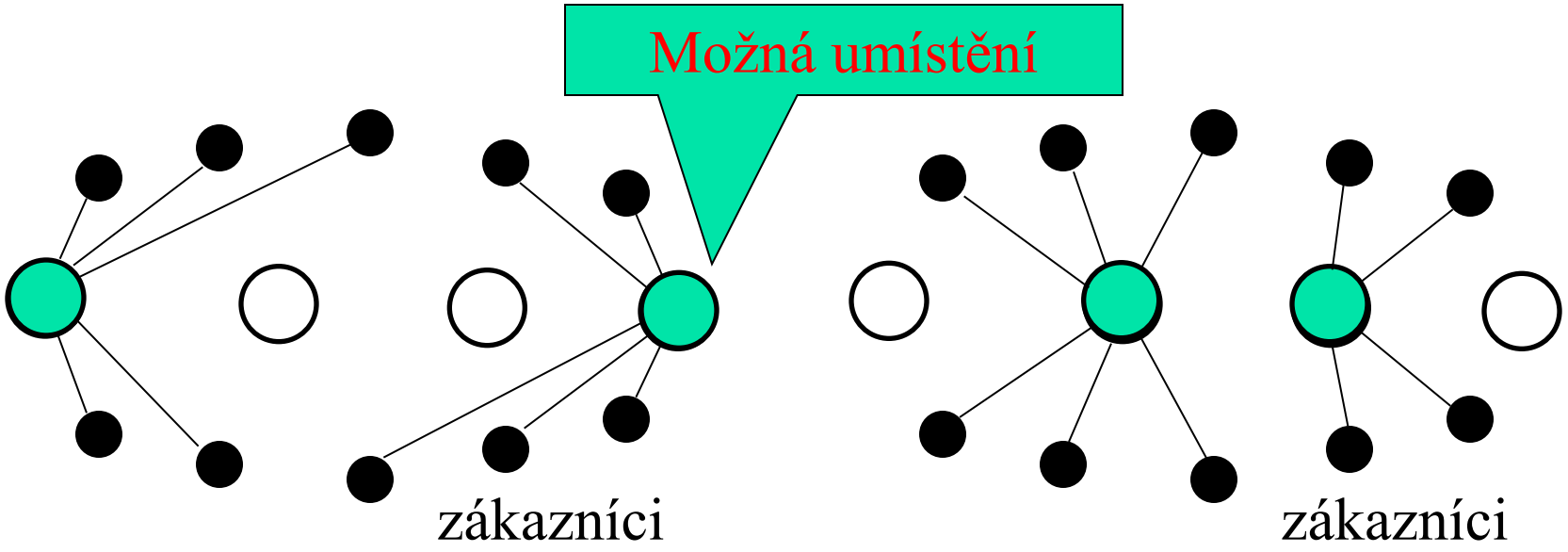
- Je vhodný pro úlohy, jejichž přípustné řešení lze popsat vektorem ***y*** s 0 - 1 složkami.
- Vektor ***y*** je **chromozóm** a jsou na něm definovány operace **mutace** a **křížení**.
- **Mutace** - s danou pravděpodobností. je na genu změněna hodnota některé složky.
- **Křížení** - z dvou chromozomů jsou vytvořeny jiné dva: Je vybráno místo křížení, chromozomy jsou tam rozděleny na počáteční a koncovou část a části jsou kombinovány do nových chromozomů. **Principem je ale jen distribuce rodičovských genů do chromozomů potomků!**

Genetický algoritmus

- GA pracuje současně s množinou řešení - **populací**.
- Práce GA začíná vytvořením počáteční populace a aktualizací dosud nejlepšího řešení.
- Potom jsou chromozomy na základě **výhodnostních koeficientů (fitness)** párovány a kříženy. Noví jedinci jsou mutováni. Nová populaci je redukována **selekcí** na pevně zvolený počet jedinců.

Úloha o p-mediánech (p-Median Problem)

- **Úloha o p-mediánech (p-Median Problem)** vznikne, když je maximální počet *p* středisek zadán a je třeba minimalizovat průměrnou (váženou) vzdálenost mezi zákazníkem a nejbližším umístěným střediskem.



Model úlohy o p-mediánech

$$Minimize \sum_{j \in J} min\{c_{ij} : i \in I, y_i = 1\}$$

$$\sum_{i \in I} y_i \leq p$$

$$y_i \in \{0,1\} \qquad for\ i \in I$$

- Řešení **úlohy o p-mediánech** je dáno posloupností čtyř proměnných [y_1, y_2, y_3, y_4]. Tuto posloupnost můžeme považovat za **chromozom** a jednotlivé prvky za **geny**.

Příklad úlohy o p-mediánech

$p=2$

$$f(I_1) = \sum_{j \in J} min\{c_{ij} : i \in I_1\}$$

c_{ij}	1	2	3	4	5	6	7	8	9	10
1	10	15	20	25	30	11	16	21	26	31
2	22	12	13	24	11	11	15	12	8	10
3	22	16	13	9	11	21	15	12	8	10
4	10	15	10	9	30	21	15	12	8	10
min	10	15	13	9	11	11	15	12	8	10

Model úlohy o p-mediánech s pokutou za nepřípustnost

$$Minimize \sum_{j \in J} min\{c_{ij} : i \in I, y_i = 1\}$$

$$\sum_{i \in I} y_i \leq p$$

$$y_i \in \{0,1\} \qquad for\ i \in I$$

- Podmínku omezení počtu umístění můžeme relaxovat pomocí nelineární pokutové funkce:

$$Q * max\{0, \sum_{j \in J} y_j - p\}$$

Model úlohy o p-mediánech s pokutou za nepřípustnost

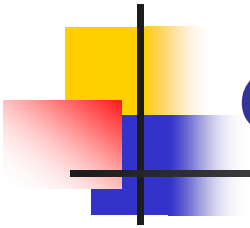
$$Minimize \quad \sum_{j \in J} min\{ c_{ij} : i \in I, y_i = 1 \} + Q * max\{ \sum_{j \in J} y_j - p \}$$

$$y_i \in \{0,1\} \qquad for \ i \in I$$

- Pro vhodně zvolené Q je tato úloha ekvivalentní s předchozí.
- Minimum z prázdné množiny je $+\infty$!

Poznámky k řešení nepřípustnosti potomků

- Aplikace pokutových funkcí znamená obvykle **zvýšení výpočetní náročnosti** účelové funkce a není to obvykle vyvážené získáním kvalitnější populace. Naopak **populace obsahuje horší jedince**, kteří musí být v dalších krocích eliminováni selekcí.
- **Kde je to možné, tak se pokutovým funkcím vyhneme!**
- **Buď nepřípustný chromozom opravíme nějakou pomocnou operací a nebo navrhneme takové operace křížení a mutace, jejichž výsledkem jsou přípustní potomci.**



Genetický algoritmus

- Samotný genetický algoritmus bude postupně nahrazovat starou populaci $\boldsymbol{y}^1, \boldsymbol{y}^2, \ldots, \boldsymbol{y}^m$ novou populací $\boldsymbol{y}^1, \boldsymbol{y}^2, \ldots, \boldsymbol{y}^m$ s průběžnou aktualizací **dosud nejlepšího nalezeného řešení**.
- Pokud proběhne *n* populačních výměn od poslední změny dosud nejlepšího nalezeného řešení, algoritmus skončí práci.
- Celý genetický algoritmus může být popsán následujícími kroky:

Genetický algoritmus- výměna populace

- 0. Inicializujte dosud nejlepší nalezené řešení nejlepším řešením $\boldsymbol{y}^k$ ze vstupní populace $\boldsymbol{Y} = \{ \boldsymbol{y}^1, \boldsymbol{y}^2, ..., \boldsymbol{y}^m \}$. Položte $HH = f(\boldsymbol{y}^k)$ a $t = 0$.

- 1. **Je-li** $t = n$ **končete**, **jinak** uspořádejte jedince $\boldsymbol{y}^k$ ze současné populace $\boldsymbol{Y}$ **vzestupně** dle hodnot $f(\boldsymbol{y}^k)$ do posloupnosti $\boldsymbol{y}^{k(1)}$, $\boldsymbol{y}^{k(2)}$, ..., $\boldsymbol{y}^{k(m)}$ a definujte počet $\underline{m}$ jedinců nové populace jako $\underline{m} = 0$. Aktualizujte dosud nejlepší nalezené řešení.

- 2. {**Selekce rodičů**}V cyklu s pravděpodobností $p(r)$ úměrnou hodnotám $f(\boldsymbol{y}^k)$ vyberte prvního a druhého z hledané dvojice rodičů $\boldsymbol{y}^{k(r)}$ a $\boldsymbol{y}^{k(s)}$.

Genetický algoritmus- výměna populace

- 3. {**Křížení (crossover)**} Vygenerujte číslo $p \in \{1, ..., m\}$ jako hodnotu diskrétní náhodné proměnné s rovnoměrným rozdělením pravděpodobnosti. Proveďte křížení $\boldsymbol{y}^{k(r)}$ a $\boldsymbol{y}^{k(s)}$ na pozici p, čímž vzniknou dva potomci $\underline{\boldsymbol{y}}^r = <y_1^{k(r)}, ..., y_p^{k(r)}, y_{p+1}^{k(s)}, ..., y_m^{k(s)}>$ a $\underline{\boldsymbol{y}}^s = <y_1^{k(s)}, ..., y_p^{k(s)}, y_{p+1}^{k(r)}, ..., y_m^{k(r)}>$.

- 4. {**Mutace**} Postupně pro potomky $\underline{\boldsymbol{y}}^r$ a $\underline{\boldsymbol{y}}^s$ s danou pravděpodobností rozhodněte, zda **provedete** mutaci, pokud ano, tak vygenerujte parametry mutace a proveďte ji. (Například tak, že změníte jeho p-tou složku dle vztahu $\underline{y}_p = 1 - \underline{y}_p$.)

Genetický algoritmus- výměna populace

- 5. {**Selekce do populace**} Z potomků $\underline{\boldsymbol{y}}^r$ a $\underline{\boldsymbol{y}}^s$ vyberte toho, který má menší hodnotu účelové funkce a zařaďte ho jako $\underline{m}$ +1 vého jedince do nové populace $\underline{\boldsymbol{Y}}$ a **je-li** hodnota menší než *HH*, aktualizujte nejlepší nalezené řešení, hodnotu *HH* a položte $t = 0$. Položte $\underline{m} = \underline{m} + 1$.

- 6. **Je-li** $\underline{m} < m$ (nová populace ještě není úplná), **jděte na krok** 2. **Jinak** (byla vytvořena nová populace $\underline{\boldsymbol{Y}}$), definujte současnou populaci $\boldsymbol{Y} = \underline{\boldsymbol{Y}}$, položte $t = t+1$ a **jděte na krok** 1.

Genetický algoritmus- výměna populace

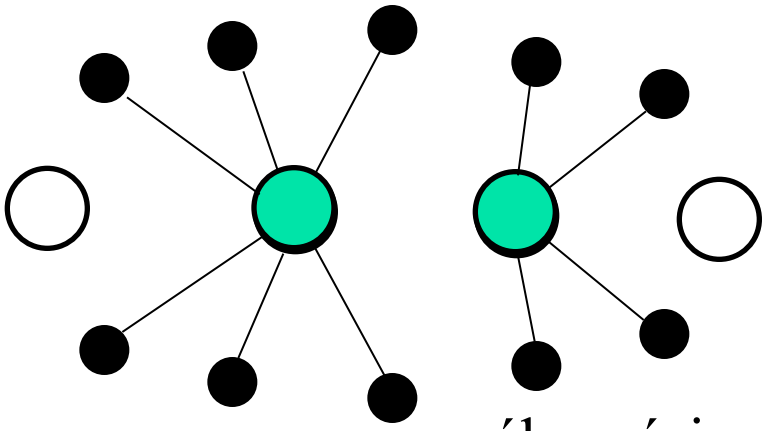
- Poznámka:
- Výběr do následující populace je možné provést také tak, že prováděním křížení a mutací **vytvoříte množinu kandidátů na vstup do další populace** (větší než je populace sama) a z ní **vyberete vhodné jedince** do následující populace.

Všeobecný rámec Genetického Algoritmu

- Vyber počáteční populaci
- (*) Dokud není splněna **podmínka zastavení** opakuj:
- (**) Je-li splněna podmínka **křížení**, vyber dva jedince (rodiče) ze současné populace a proveď **křížení**.
- Je-li splněna podmínka **mutace**, vyber na jedinci body mutace a proveď **mutaci**.
- Vyhodnoť výsledné jedince (**fitness**) a vlož je do seznamu potomků
- Není-li vytvořen dostatečný počet potomků, jdi na (**).
- **Selekcí** vytvoř novou populaci a jdi na (*).

Operace křížení (Crossover)

- Záleží na tom, jak je jedinec-**řešení úlohy** (chromozom, genotyp) reprezentován. Rozlišujeme dva případy:
  a) Jedinec je specifikován **množinou** objektů.
  b) Jedinec je specifikován **pořadím** objektů.
- Příkladem a) je úloha o p-mediánu kde řešení úlohy je dáno **hodnotami** posloupnosti 0-1 proměnných [$y_1, y_2,.., y_n$], např. [0, 1,1, 0].

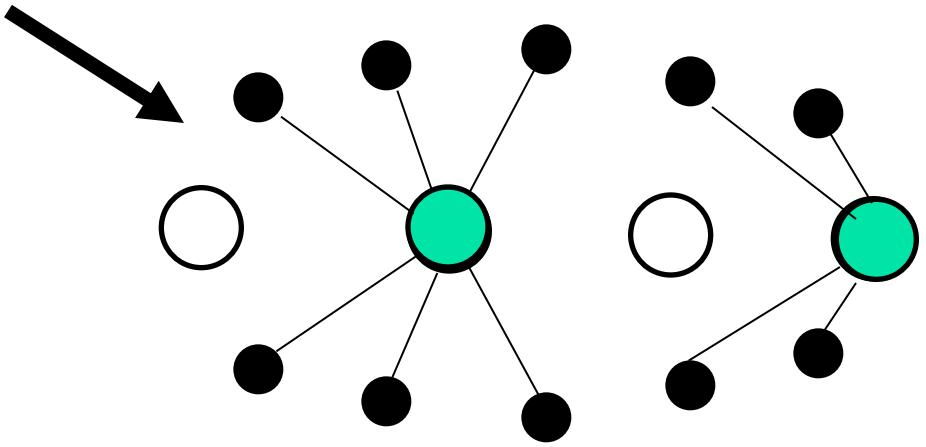


Operace křížení (Crossover)

Jedinec je specifikován **množinou** objektů.

- Křížíme-li dva jedince, např. [1, 0, 0, 1] a [0, 1, 1, 0], potom je vygenerován „bod zlomu“, např. p=2 a křížení proběhne takto:
- [1, 0, 0, 1] [1, 0, 1, 0]
  [0, 1, 1, 0] [0, 1, 0, 1]

p=2



zákazníci

Operace křížení (Crossover)

Jedinec je specifikován **seznamem *p*** objektů.

```
public void Crossover(int m, int p, int [] Zp1, int [] Zp2, int [] Zo1,  int [] Zo2){
    int j;  int pom;  int noZo=0;  int noZc=0;
    for(i=0; i<m; i++){  iY[i]=0;}
    for(i=0; i<p; i++){  iY[Zp1[i]]=1;}
    for(i=0; i<p; i++){
       if(1<++iY[Zp2[i]]){  // i.e. ==2
          Zo1[noZo]=Zp2[i];
          Zo2[noZo]=Zp2[i];
          noZo++;
       } // The common 1-genes have been inherited from parents
    }
    // Saving different genes into sY from 0 to noZc-1
    for(i=0;  i<m;  i++){
       if(iY[i]==1){  sY[noZc++]=i;}
    }
```

Operace křížení (Crossover)

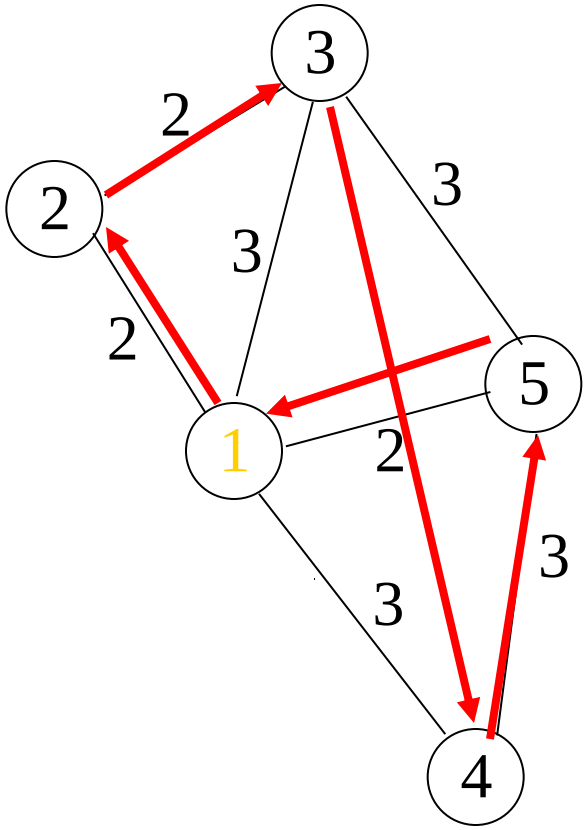
Jedinec je specifikován **seznamem *p*** objektů.

```
// Choose noZc/2 elements randomly
      for(i=0;  i<noZc/2;  i++){
          j=(int)Math.floor((noZc-i)*Math.random());
          pom=sY[j];
          sY[j]=sY[noZc-1-i];
          sY[noZc-1-i]=pom;
       }
      for(i=0; i<noZc/2; i++){
         Zo1[noZo]=sY[i];
         Zo2[noZo]=sY[noZc/2+i];
         noZo++;
       }
  }// end of Crossover
```

Operace křížení (Crossover)

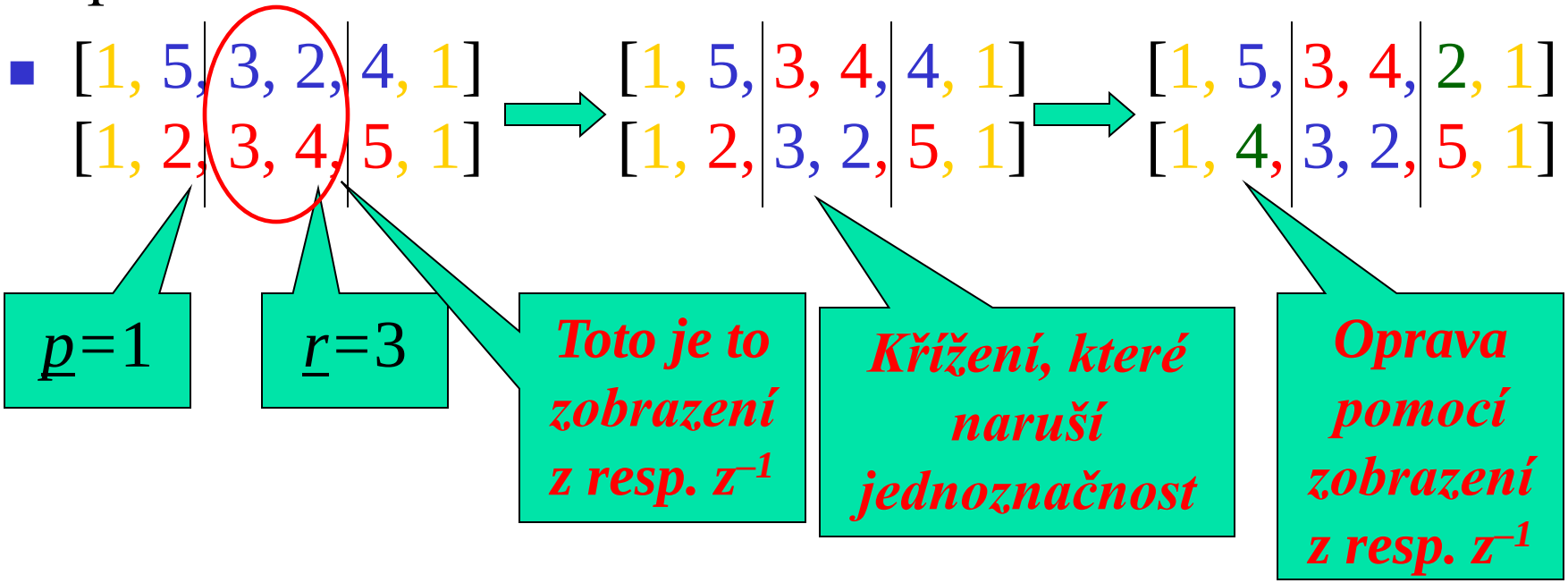
Jedinec je specifikován **pořadím** objektů.

- Příkladem b) je úloha **obchodního cestujícího**, kde řešení je dáno pořadím navštěvovaných uzlů, např.
  [1, 2, 3, 4, 5, 1]
- Zde může být použito některé ze schémat:
- Partially mapped crossover (PMX)
- Uniform crossover (UX)



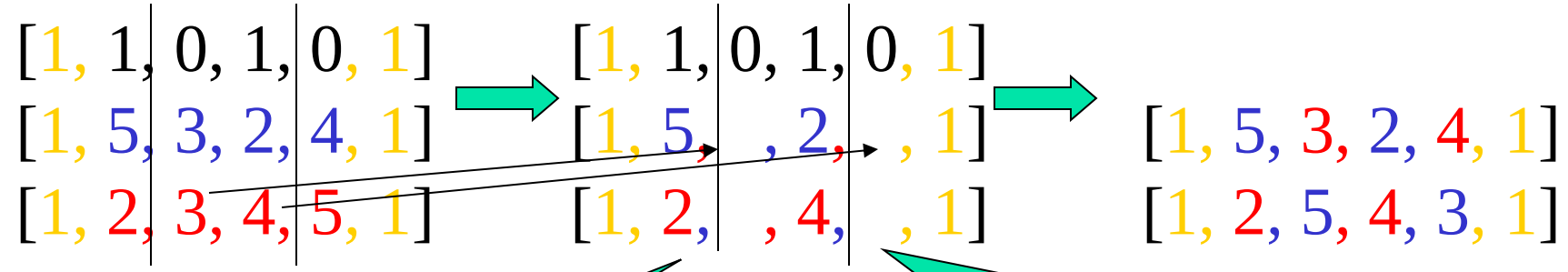
Partially mapped crossover (PMX)

- Křížíme-li dva jedince, např. [1, 5, 3, 2, 4, 1] a [1, 2, 3, 4, 5, 1] , potom jsou vygenerovány **dva** „body zlomu“, např. $\underline{p}=1$ a $\underline{r}=3$ a křížení proběhne takto:



Uniform crossover (UX)

- Křížíme-li dva jedince, např. [1, 5, 3, 2, 4, 1] a [1, 2, 3, 4, 5, 1] , potom je náhodně vygenerována **šablona (mask)** např. [1, 1, 0, 1, 0, 1] a křížení proběhne takto:



Kopírování podle šablony

Prázdná místa se doplní z druhého z rodičů podle pořadí s přeskakováním prvků, které tam už jsou.

Operace mutace (Mutation)

- Nejdříve je třeba rozhodnout, zda se mutace na daného jedince bude uplatňovat.
- Dále opět rozlišujeme dva případy:
  a) Jedinec je specifikován **množinou** objektů.
  b) Jedinec je specifikován **pořadím** objektů.
- V případě a) může být náhodně vygenerována **šablona** např. [1, 0, 1, 0], dle Bernouliho rozdělení (s malým parametrem *p*) a hodnota v chromozomu odpovídající jedničce v šabloně je invertovaná.
- [1, 0, 0, 1] [0, 0, 0, 0]
  [1, 0, 1, 0] ⇨

Operace mutace (Mutation)

- V případě a) může být použito Poissonovo rozdělení s parametrem $\lambda=1$ k vygenerování počtu mutací, který má být proveden na chromozomu a ten počet mutací je proveden na genech, které náhodně (s rovnoměrným rozdělením) *m*- násobným výběrem.

- V případě b) může být mutace provedena například výměnou dvou prvků v daném pořadí, kde pozice vyměňovaných prvků jsou vygenerovány náhodně.

Strategie genetického algoritmu (podmínky křížení a mutace)

- Strategie **Křížení** a **Mutace**
- Strategie **Křížení** nebo **Mutace**

Vhodnost jedince (Fitness)

- Obvykle se jako míra vhodnosti jedince bere **hodnota účelové funkce** daného řešení. Vzniká zde ale problém s proporčností. (Porovnej hodnoty 10 a 20 a hodnoty 110 a 120.)
- Můžeme to řešit buď **změnou měřítka** (**Scaling**) např. vydělením nejmenší hodnotou.
- Nebo **uspořádáním** (**Ranking**)
  Zde uspořádáme podle rostoucí vhodnosti (účelové funkce) jedince a *k –tému* dáme pravděpodobnost
  $P(k) = \alpha + \beta * k$, kde parametry α a β vhodně zvolíme.

Vhodnost jedince (Fitness)
Ranking

- Zde uspořádáme podle rostoucí vhodnosti (účelové funkce) jedince a *k –tému* dáme pravděpodobnost
  $P(k) = \alpha + \beta * k$, kde parametry α a β vhodně zvolíme.
- Definujeme-li „tlak selekce“ (selection pressure) z intervalu <1, 2 > jako $\Phi = (\alpha + \beta * m)/(\alpha + \beta * m/2)$ je možno vyjádřit $\alpha = (2m - \Phi(m+1))/(m(m-1))$ a
  $\beta = 2(\Phi - 1)/(m(m-1))$.
- Pokud nyní vygenerujeme náhodné číslo $r \in <0, 1>$ a chceme zjistit do jakého intervalu daného kumulativními pravděpodobnostmi padne, řešení je snadné.

Vhodnost jedince (Fitness)
Ranking

- $P(k) = \alpha + \beta * k$.
- Pokud nyní vygenerujeme náhodné číslo $r \in \langle 0, 1 \rangle$ a chceme zjistit do jakého intervalu daného kumulativními pravděpodobnostmi padne, řešíme:

$$\alpha k + \beta \frac{k(k+1)}{2} = r$$

$$a\ tedy\ k = \left\lceil \frac{-(2\alpha + \beta) + \sqrt{(2\alpha + \beta)^2 + 8\beta r}}{2\beta} \right\rceil$$

Selekce (Selection)

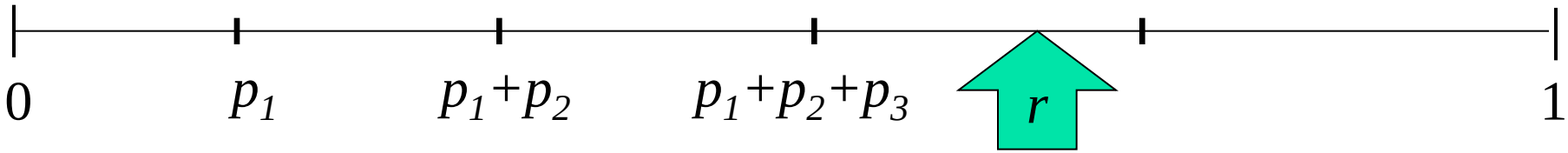
- Selekcí se snažíme jednak vytvořit vhodné páry rodičů a také potom ze seznamu potomků vybrat novou populaci.
- K vytvoření párů rodičů můžeme použít několik schémat:
- Roulete-Wheel-Selection (RWS)
- Baker’s Stochastic Universal Selection (SUS)
- Tournament Selection

Roulete-Wheel-Selection (RWS)

- Hodnoty účelových funkcí (fitness) $f_1, f_2, f_3, \ldots f_m$ se transformují na pravděpodobnosti $p_1, p_2, p_3, \ldots p_m$ například podle vztahu:

$$p_k = \frac{f_k}{\sum_{i=1}^{m} f_i}$$

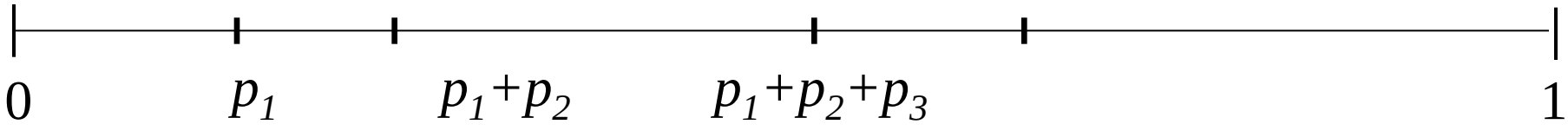
- Potom generujeme s rovnoměrným rozložením $r \epsilon <0, 1>$:



- Na základě pokusu zde vybereme jedince 4.

Baker's Stochastic Universal Selection (SUS)

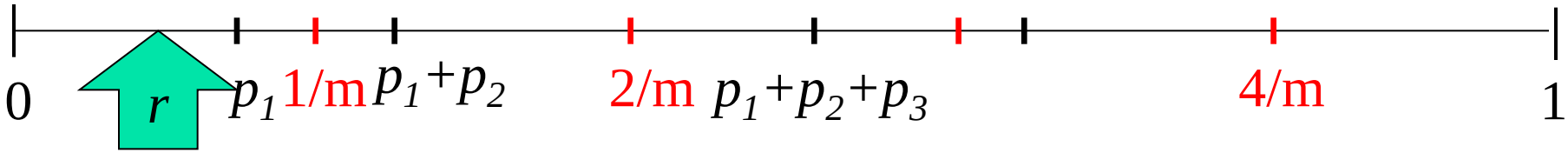
- Hodnoty účelových funkcí (fitness) $f_1, f_2, f_3, \ldots f_m$ se transformují na pravděpodobnosti $p_1, p_2, p_3, \ldots p_m$.



- Potom generujeme s rovn. rozložením $r \in <0, 1/m>$:

Baker's Stochastic Universal Selection (SUS)

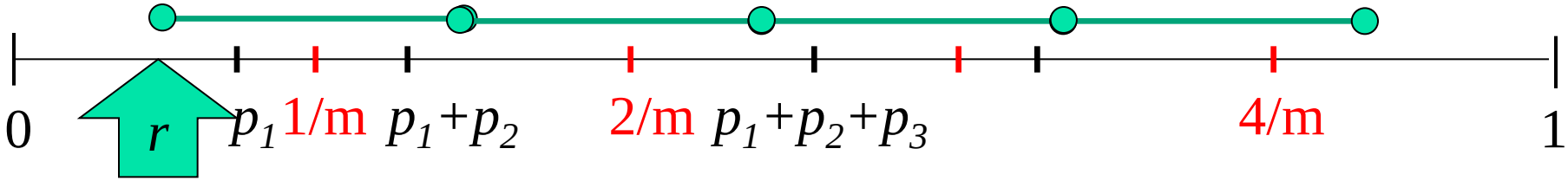
- Hodnoty účelových funkcí (fitness) $f_1, f_2, f_3, \ldots f_m$ se transformují na pravděpodobnosti $p_1, p_2, p_3, \ldots p_m$.



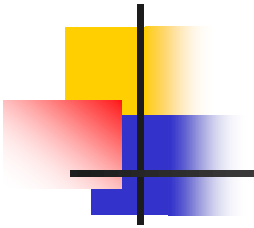
- Potom generujeme s rovn. rozložením $r \in \langle 0, 1/m \rangle$:

Baker's Stochastic Universal Selection (SUS)

- Hodnoty účelových funkcí (fitness) $f_1, f_2, f_3, \ldots f_m$ se transformují na pravděpodobnosti $p_1, p_2, p_3, \ldots p_m$.



- Potom vypočítáme $r_j = r + (j-1)/\mathrm{m}$ pro $j = 1, \ldots, m$.
- Potom vybereme jedince 1, 3, 3, 5, 5 a potom musíme provést **párování**.



Tournament Selection (Výběr turnajem)

- Náhodně vybereme z populace jistou podmnožinu jedinců a z ní podle míry výhodnosti (fitness) určíme vítěze.
- Můžeme „uspořádat“ i vícestupňový turnaj.
- Můžeme výběr do skupiny omezit nějakou hranicí odvozenou od hodnoty dosud nejlepšího výchozího řešení.

Podmínky zastavení a rozličnost (Diversity)

- Jako pravidlo zastavení můžeme použít dosažení jistého počtu vyhodnocení vhodnosti nebo počtu vystřídaných populací.
- Proces také můžeme zastavit, když od poslední aktualizace dosud nejlepšího řešení byl proveden daný počet operací.
- Doporučuje se rovněž zastavit proces, sníží-li se **diversita** pod jistou úroveň. Jistou mírou **diversity** je podíl nejčastějšího výskytu stejné hodnoty na dané pozici chromozomu.

Podmínky zastavení a rozličnost (Diversity)

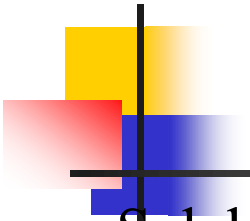
- Doporučuje se rovněž zastavit proces, sníží-li se **diversita** pod jistou úroveň. Jistou mírou **diversity** je podíl nejčastějšího výskytu stejné hodnoty na dané pozici chromozomu.

[1, 5, 3, 4, 2, 1]
[1, 5, 3, 2, 4, 1]
[1, 5, 3, 2, 4, 1]
[1, 2, 5, 4, 3, 1]
[1, 5, 3, 2, 4, 1]

populace

80%

60%



Doporučení

- Selekci radši pomocí „Baker's Stochastic Universal Selection (SUS)“ nebo „Tournament Selection“ než „Roulete-Wheel-Selection (RWS)“.
- Nepoužívat jen jednobodové křížení.
- Používat adaptivní frekvenci mutace.
- Hybridizovat GA (nebát se metodu GA kombinovat s jinými heuristikami využívajícími specifik úlohy).
- Udržovat diversitu.
- Spouštět GA vícekrát.