

Evoluční výpočetní techniky (EVT)

- Nacházejí svoji inspiraci v přírodních vývojových procesech
- Stejně jako přírodní jevy mají silnou náhodnou složku, která nezanedbatelným způsobem ovlivňuje jejich vlastnosti a chování
- Klasickým představitelem jsou **genetické algoritmy**, nejmladším představitelem je **genetické programování**.

Něco k historii

- 1. polovina 60. let – 3 studenti univerzity v Berlíně, kteří se zabývali konstrukcí převodovek, přišli s myšlenkou náhodně kombinovat dvojice existujících konstrukcí ve snaze nalézt konstrukci s lepšími užitnými vlastnostmi (připomíná to přírodní křížení v rámci téhož biologického druhu)
- 2. polovina 60. let – *evoluční programování* – evoluce chování konečných automatů a vyvinutí jejich schopnosti predikovat změny prostředí, v němž se automat nachází
- 1975 – položen základ *genetickým algoritmům*
- 1989 – systematické studium genetických algoritmů z pohledu technického, tedy z pohledu, který se snaží genetické algoritmy chápat jako techniku obecně aplikovatelnou na širokou třídu úloh
- 1992 – *genetické programování* – jeho cílem je automatizovaně generovat počítačové programy, které řeší určitou konkrétní úlohu nebo skupinu úloh. Od genetických algoritmů se genetické programování liší v tom, jak jsou reprezentováni jedinci podílející se na evolučním procesu a jak jsou tito jedinci interpretováni a ohodnocováni.

Základní charakteristiky EVT

- V přírodní evoluci je základní úlohou biologického druhu vyhledávání výhodných adaptací vůči složitému a dynamicky se měnícímu prostředí. „Znalost“, která charakterizuje každý biologický druh, byla získána vývojem a je shrnuta v chromozomech každého jedince.
- Evoluci v přírodě lze tedy chápat jako optimalizaci, kdy vyhledávané adaptace (změny) musí být „výhodné“ ve smyslu nějakého hodnotícího kritéria. V přírodě je zpravidla tímto hodnotícím kritériem schopnost přežití a reprodukce, tzv. **fitness**.
- Z myšlenek o přírodě vychází také algoritmické využití EVT.
- Nejčastěji se používají pro hledání ve stavovém prostoru, jejich cílem je nalezení optimálního řešení dané úlohy.
- Jsou to stochastické algoritmy, jejichž prohledávací schopnost je posílena modelováním Darwinovského zápasu o přežití a modelováním genetické dědičnosti.
- Tradiční optimalizační postupy iterativně vylepšují jeden počáteční odhad řešení.
- EVT nepracují s jedním kandidátem řešení, ale s jejich množinami, tj. s **populacemi** jedinců – kandidátů řešení $x_{t,i}$ ve tvaru

$$G(t) = \{x_{t,1}, x_{t,2}, \dots, x_{t,N}\}$$

kde t je vývojový čas a N označuje rozsah populace. Vývojový čas zpravidla běží v diskrétních krocích a o posloupnosti populací se často hovoří jako o **generacích**. Všichni jedinci v populaci jsou implementováni pomocí týchž datových struktur S . Každý jedinec $x_{t,i}$ je ohodnocen svojí mírou kvality (fitness), která je vyjádřena obvykle reálným číslem.

- Obecný tvar algoritmu EVT sestává z následujících kroků:

{ $t = 0$

inicializace $G(t)$

vyhodnocení $G(t)$

dokud není splněna zastavovací podmínka, pak

{ $t = t+1$

selekce vhodných jedinců z $G(t-1)$ pro $G(t)$

změna některých jedinců pro $G(t)$

vyhodnocení $G(t)$

náhrada generace $G(t-1)$ generací $G(t)$

}

}

- Při *inicializaci* je počáteční populace obvykle vytvořena náhodně, může však být vytvořena i na základě apriorní znalosti úlohy. Při *vyhodnocení* se počítá kvalita (fitness) všech jedinců v populaci. *Selekce* simuluje proces přirozeného výběru a jsou při ní vybíráni jedinci z předchozí populace, kteří se budou podílet na vytvoření populace nové. Selektce je obvykle realizována nějakým pravděpodobnostním mechanismem, přičemž lepší jedinci (tj. jedinci s větší kvalitou) mají větší šanci být vybráni. Průměrné ohodnocení následující populace by tedy mělo být lepší. *Změna* reprezentuje inovaci v celém vývojovém procesu a je obvykle realizována pomocí tzv. **rekombinačních operátorů** (někdy se nazývají též **genetické operátory**). Ty jsou obvykle 2 typů:
 - o **Mutace** – nový jedinec se vytváří „malou“ změnou původního jedince
 - o **Křížení** – jeden nebo více jedinců se generuje z několika (dvou či více) jedinců – rodičů

Náhrada představuje postup (**vývojovou strategii**), jak ze smíšené množiny „rodičů“ a „potomků“ vybrat množinu, kterou budeme v dalším vývoji považovat za novou populaci. Obvykle se používají 2 typy vývojových strategií:

- **Generační strategie** – jedna populace se úplně nahradí populací následující (analogie s jednoletými rostlinami)
- **Postupná strategie** – změna se týká jen malé části populace a rodiče koexistují se svými potomky (analogie s déle žijícími živočichy a s víceletými rostlinami)

Zastavovací podmínka ukončuje činnost algoritmu. Obvykle se nastavuje tak, že algoritmus končí ve chvíli, kdy nedochází ke změnám v generacích.

Genetické algoritmy (GA)

- Charakteristické tím, že jedinci jsou reprezentováni pomocí struktur S ve tvaru řetězců (s_1, s_2, \dots, s_L) délky L nazývaných **chromozomy**. Každá pozice v řetězci (chromozomu) se nazývá **alela**, konkrétní symbol s_i v chromozomu se nazývá **gen**.

Standardní genetické algoritmy

- Jedinci jsou reprezentovány pomocí řetězců (chromozomů) fixní délky tvořených dvouhodnotovými symboly (zpravidla 0 a 1).
- Při **křížení** se generují právě 2 potomci ze dvou rodičů.
- Používá se tzv. **jednobodové křížení**:
 - Zvolí se tzv. **křížící bod** jako náhodné celé číslo v rozsahu 1 až $(L-1)$, kde L je délka řetězce
 - V křížícím bodě se vede řez oběma rodičovskými řetězci

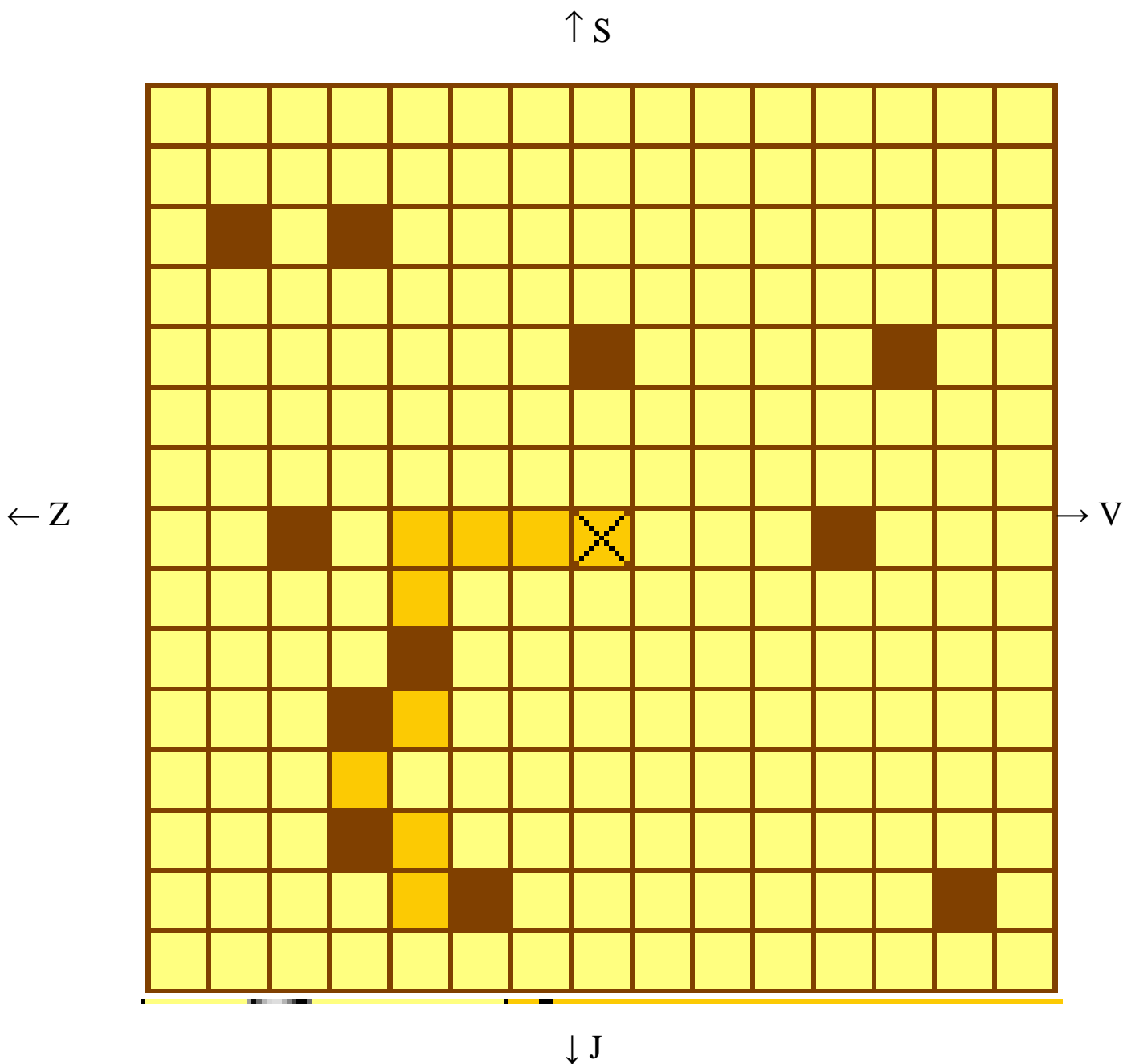
- Potomci vzniknou tak, že jeden je vytvořen levou částí řetězce prvního rodiče a pravou částí řetězce druhého rodiče, u druhého potomka je tomu naopak



- Při **mutaci** se náhodně vybere několik jedinců a u nich se invertuje jeden náhodně vybraný znak (gen)



Příklad – hledání potravy: Najděte optimální cestu, při které mravenec (znázorněný křížkem) najde během svého života co nejvíce krmítek s potravou (znázorněna hnědými políčky). Předpokládejte, že délka života mravence je 12 kroků.



délka života 12 kroků => chromozomy jsou tvořeny 12 znaky (geny)

geny – 4 směry: V Z S J

inicializace – 4 náhodní jedinci

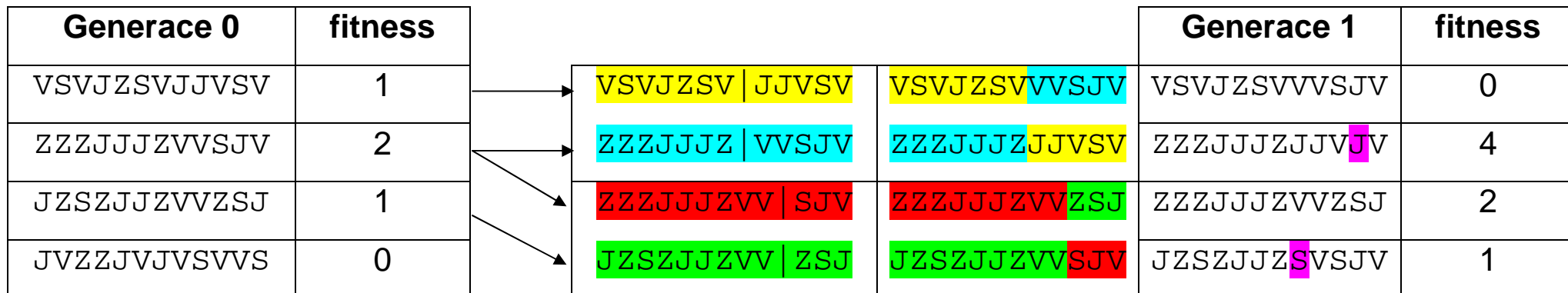
kvalita (fitness) = počet nalezených kousků potravy (hnědá políčka)

selekce – úměrně kvalitě

– selekce →

– křížení →

– mutace →



Oblasti aplikace genetických algoritmů

- základní otázkou efektivní aplikace je vždy nalezení vhodné reprezentace jedinců
- aplikace v architektuře a stavebním inženýrství – jde o úlohy návrhu konstrukčních a tvarových struktur. Základním východiskem je existence „slovníku“ elementárních struktur, např. geometrických tvarů, v různém natočení a v různé velikosti. Tyto struktury jsou reprezentovány pomocí bitových řetězců. Pospojováním takovýchto elementárních struktur vznikne chromozom popisující nejrůznější tvarové struktury. Účelovou funkcí, jejíž extrém se hledá, je ohodnocení konstrukční realizovatelnosti a estetický vzhled nalezené kompozice.
- aplikace v počítačovém a informačním inženýrství – např. plánování úloh na víceprocesorových systémech
- aplikace ve strojním a průmyslovém inženýrství – rozvrhování výroby, optimalizace „nářezového plánu“ (jak rozřezat dlouhou tyč na menší různých délek nebo jak z plechu vyřezat zadané tvary tak, aby bylo co nejméně odpadu)
- identifikace zločinců – vytváření podoby obličeje pachatele (svědkovi se prezentuje několik různých obličejů, on je seřadí podle podobnosti s pachatelem. Nejpodobnější jsou využity ke generování „potomků“, které by měly být pachateli podobnější než „rodiče“)

Genetické programování (GP)

- vychází z principu genetických algoritmů, ale liší se od nich ve 2 směrech:
 - o aplikační oblastí GA je hledání hodnot nezávisle proměnných, pro které určitá hodnoticí funkce nabývá optima; GP je určeno pro automatickou syntézu programů, algoritmů či rozhodovacích postupů
 - o u GA jsou jedinci reprezentováni chromozomy ve tvaru lineárních řetězců, u GP se využívají nejčastěji stromové struktury.
- hledaným řešením v GP jsou výrazy, programy či algoritmy. Ty lze obvykle reprezentovat ve tvaru stromových struktur. Strom je obvykle konstruován ze 2 množin symbolů – z **terminálních symbolů**, které jsou reprezentovány listovými uzly, a z **neterminálních symbolů**, které jsou reprezentovány vnitřními (nelistovými) uzly. Neterminální symboly reprezentují funkce či operace, terminální symboly reprezentují vstupní proměnné, konstanty nebo výstupy senzorů.
- ▶
- ▶
- Při **křížení** se generují právě 2 potomci ze dvou rodičů – v každém z rodičovských chromozomů se náhodně vybere po jednom uzlu a podstromy vycházející z těchto uzlů se vzájemně zamění
- ▶
- **Mutace** probíhá tak, že ve zvoleném jedinci se náhodně vybere jeden uzel a podstrom, který z něho vychází, se zcela nahradí nově vygenerovaným stromem.
- ▶

Oblasti aplikace genetického programování

- hledání analytického předpisu pro generování posloupností
- generování programu popisujícího cílené individuální chování
- generování programu popisujícího chování kolektivu na základě chování jednotlivce

Závěrečné poznámky

- evoluční výpočetní techniky jsou základem tzv. **umělého života (Alife)**, jehož cílem je z jednoduchých mikroskopických spolupracujících prvků generovat takové chování na úrovni makroskopické, které je možné interpretovat jako projev života
- umělý život sleduje dva cíle:
 - o prohloubit naše znalosti o přírodě (a zdokonalit naše představy o umělých modelech)
 - o umožnit tak zlepšení výkonnosti těchto modelů
- existují **3 definice umělého života**:
 - o **slabá definice** – jedná se pouze o simulace,
 - o **silná definice** – umělé entity jsou skutečně živé,
 - o **supersilná definice** – život na bázi křemíku bude dalším evolučním krokem; tato forma života překoná naše schopnosti a pravděpodobně nás i nahradí,