**Příprava na test, viz soubor „Zápočtový test“ na sdíleném Google disku.**

**Genetické algoritmy**

Řeší optimalizační problémy.

Optimalizační problém je hledání nejlepší sady parametrů, tak aby funkce s těmito parametry byla minimalizovaná nebo maximalizovaná.

*ax* + *by* + *c* = 0 je rovnice přímky, která má mít nejmenší sumu čtverců vzdáleností od bodů v rovině, které jsou výsledkem měření reálného jevu, například závislost počtu automobilů na ploše pro dopravní komunikace.

Při optimalizaci hledáme parametry *a*, *b*, *c*, pro které je suma čtverců nejmenší.

Pro lineární regresi existuje algoritmus, a proto tam genetický algoritmus není nutný.

Pro složitější problémy se spoustou lokálních extrémů, viz například <https://www.sfu.ca/~ssurjano/optimization.html>, jsou genetické algoritmy vhodné.

Pokud funkce nemá lokání minima nebo maxima ale jen jedno globální, viz například <https://www.sfu.ca/~ssurjano/boha.html>, tak je pro hledání parametrů vhodný nějaký gradientní algoritmus, například backpropagation nebo regrese.

Genetické algoritmy byly vynalezeny proto, abychom při hledání globálního minima neustrnuli v nějakém lokálním minimu.

Genetické algoritmy to dělají tak, že paralelně testují více možností, neboli více sad hledaných parametrů.

**Jedinec** – testované řešení jako konkrétní sada parametrů, například *a*, *b*, *c*. Celá sada parametrů je genetickou výbavou jedince.

**Populace** – všichni jedinci, kteří jsou současně testováni, nebo existují v paměti počítače.

**Gen** – konkrétní hodnota parametru pro konkrétní řešení, například *a* nebo *b* nebo *c*.

**Fitness function** – hodnota funkce s konkrétní sadou parametrů. Jedinec je víc fit, má větší fitness, když se jeho hodnota víc blíží optimu (maximu nebo minimu, které hledáme).

Genetický algoritmus se snaží vygenerovat jedince s optimálním řešením, tak že si udržuje populaci jedinců a s jedinci dělá operace, které je mění, což jsou nejčastěji **mutace** a **křížení**.

Realizace **mutace**: Jedinec se zkopíruje a jeho kopie si změní geny o nějakou malou hodnotu. Statistické rozdělení mutací může být velmi důležité. Takže například uniformní rozdělení může být horší než normální rozdělení.

Realizace **křížení**: Dva jedinci poskytnou část svých genů pro sestavení potomka, který je zpravidla dál od rodičů než výsledek mutace od původního jedince.

Takto vzniknou noví jedinci a v nich provedeme **selekci** tak, aby populace zůstala stejně velká a byla v ní větší pravděpodobnost, že v ní jsou jedinci, kteří jsou více fit.

Kdyby selekce vybrala jen ty nejlepší jedince, tak se jako u gradientního algoritmu může stát, že se všichni jedinci sejdou v lokálním minimu nebo maximu, tedy populace se stane uniformní.

Proto součástí selekce je i kritérium **diversity**, tedy do další generace vybíráme i jedince, kteří jsou nejdál od nejlepšího jedince.

Kritériem pro selekci může být **rank** jako součet pořadí podle fitness a pořadí podle diversity.

Zdroje:

<https://en.wikipedia.org/wiki/Genetic_algorithm>

<https://link.springer.com/article/10.1007/s11042-020-10139-6>