15.5.2019

**Report k semestrální práci z předmětu BI-ZUM**

Ondřej Schejbal

[schejond@fit.cvut.cz](mailto:schejond@fit.cvut.cz)

**Abstrakt**

Mým úkolem bylo vyšlechtit pozici pěti bodů tak, aby bylo možné podle souřadnic těchto bodů rozeznat 26 znaků anglické abecedy. Rozhodl jsem se tedy úlohu vyřešit za použití genetického algoritmu. Během psaní a ladění algoritmu jsem narazil na spoustu situací, kdy mnou šlechtěné generace vždy po určité době začaly konvergovat k jedné hodnětě fitness funkce a bylo obtížné a zdlouhavé se dále přibližovat ke správnému výsledku. Postupně jsem implementoval 3 evoluční techniky: *deterministic crowding*, *katastrofu* a *hill climbing*. Po testování, ladění programu a hlavně spojení katastrofy s hill climbingem určitých jedinců jsem dospěl k výsledku, kde je můj algoritmus schopný vyšlechtit jedince obsahujícího řešení vždy efektivně v krátkém času a během maximálně pár set generací.

**Parametry algoritmu**

Můj program na vstupu přijme absolutní cestu ke složce obsahující sadu obrázků s písmeny anglické abecedy. Ze složky načtu jednotlivé podoby obrázků a každý uložím do objektu Letter reprezentujícího jedno písmeno. Jedinec pro můj genetický algoritmus je tvořen pěti genomy, kde každý genom je pozice v 2D poli o rozměrech 16x16, což jsou i rozměry odpovídající obrázkům se znaky abecedy v obdrženém datasetu. Kdykoliv během algoritmu vytvářím nového, křížím mezi sebou, nebo jen mutuji jedince, vždy po dokončení této operace zavolám opravný operátor, který zajistí aby v rámci jedince nebyly některé pozice stejné. Jako fitness funkci jsem si zvolil počet písmen, které jsem schopen pomocí jedince rozlišit. Hledaná hodnota fitness funkce je tedy 26. Pravděpodobnosti křížení a mutace jsem během implementace a testování různě měnil a nakonec jsem našel optimální hodnoty a to 33%, resp. 4%. Za chodu si také počítám opakovaný výskyt nejvyšší fitness hodnoty mezi generacemi a pokud se mi 15 generací po sobě nejlepší hodnota fitness funkce nezměnila, tak způsobím katastrofu, která zabije ¾ populace, nechá na živu pouze náhodnou ¼ populace a vítěze z turnaje, který se konal v původním plném počtu populace. Každého vítěze před vložením do této nové populace nejdřív projedu hill climbing algoritmem.

**Operátor mutace**

Operátor mutace náhodně rozhodne pro každou pozici v jedinci, zda ji nahradí za zcela nově vygenerovanou pozici. Na konci se opět zavolá opravný operátor.

**Operátor křížení**

Křížení jsem zvolil jednobodové, ale sprvkem náhody. Vygeneruji si náhodné číslo od 0 do 4 a na základě něj určím bod křížení rodičů. Po zkřížení rodičů a vytvoření potomků nechámpotomky bojovat vždy se sobě podobnějším rodičem o to, kdo půjde do další generace mé populace (takzvaný *deterministic crowding)*

**Selekce**

Použil jsem turnajovou selekci, kde vyberu 1/20 jedinců z populace ze soubojů skupinek o velikosti 1/6 populace. Vybral jsem tyto hodnoty na základě testování a snahy aby vybraných jedinců nebylo příliš mnoho a zároveň jsem pomocí vybraných jedinců byl schopný zlepšovat fitness hodnotu. Vyselectěné jedince poté projedu a s větší pravděpodobností (44%) skřížím mezi sebou.

Po vytvoření celé nové populace jsem přidal s 0.5% katastrofu, ve které smažu ½ náhodných jedinců, vložím vyselectěné jedince z turnaje a doplním počet jedinců v generaci náhodnými, nově vytvořenými jedinci.

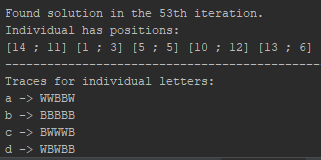
Každou generaci také 25 jedinců, kteří nebyli zvoleni ani pro mutaci ani pro křížení, namísto pouhého zkopírování do další generace, nejdříve projedu hill climbing algoritmem a snažím se je zlepšit na základě jejich okolí.

**Hill climbing**

Hill climbing jsem implementoval tak, že se postupně pro každou souřadnici z jedince rozhlížím o krok doprava, doleva, nahoru a pak dolu, zda by se mi nezlepšila fitness hodnota. Pokud ne, zkusím udělat tento náhled z další souřadnice jedince. Po projetí všech souřadnic opakuji tentokrát se náhledem na dva kroky najednou. Pokud se nedoberu zlepšění ani do vzdálenosti až 8 kroků, nechám jedince být, tak jak byl. Pokud se doberu zlepšění, zarekurzím se a zavolám znovu hill climbing tentokrát na jedince s touto výhodnější pozicí.

**Vývoj hodnoty fitness**

todo add graph

**Shrnutí a výsledky**

Mnou napsaný algoritmus hledá a vyšlechtí ideálního jedince během velmi malého počtu generací. Vypíše, ve které generaci se jedinec nalezl, jedincovi souřadnice a jakým způsobem od sebe odliší různá písmena.

Ukázka nalezeného řešení

Při implementaci jsem měl problémy zvláště zezačátku implementace, ale vhodné zakomponování a aplikace hill climbingu můj algoritmus nakonec zavedla ke zdárnému konci. Práce na semestrální práci mě bavila, i když jsem si chvílema přišel bezradný a nedařilo se mi vyšlechtit řešení v rozumném čase. Nakonec jsem ale k řešení dospěl a mám pocit, že evolučnímu algoritmu a evolučním technikám nyní rozumím mnohem lépe, než jsem se dozvěděl pouze z přednášek předmětu BI-ZUM, ze kterých jsem také vycházel.