

Praktická úloha řešena Genetickým algoritmem

Dokumentace k projektu z předmětu SFC, zadání č.91

Úvod

Genetické algoritmy se vyznačují svými evolučními schopnostmi, jsou založeny na principech genetiky a mechanismech přirozeného výběru. Jsou používány pro řešení složitých a optimalizačních problémů, pro které neexistuje konkrétní řešení. V tomto textu se zabývám využitím genetického algoritmu v praxi pro sestavení školního rozvrhu, neboť tato úloha je určitě vhodným zastupitelem praktického použití v reálném životě.

Reprezentace znalostí

Mezi základní pojmy, se kterými genetické algoritmy operují jsou následující:

- chromozóm,
- gen,
- populace a
- fitness hodnota.

Chromozóm je lineárně zakódována informace, která vyjadřuje vlastnosti daného jedince. Může nabývat mnoho podob, avšak nejčastěji se vyjadřuje v binární soustavě.

Gen je nejmenší dílčí část chromozómu.

Populace je soubor jedinců, popsány svými chromozómy v rámci jedné generace.

Fitness hodnota vyjadřuje kvalitu každého jedince. Může také vyjadřovat kvalitu celé populace, například vypočtením průměrné hodnoty fitness všech jedinců aktuální populace. Obvykle se jedná o reálné číslo na intervalu od 0 do 1. Pro problém řešený daným genetickým algoritmem se musí stanovit fitness funkce, která bude stanovovat hodnotu jedince.

Algoritmus

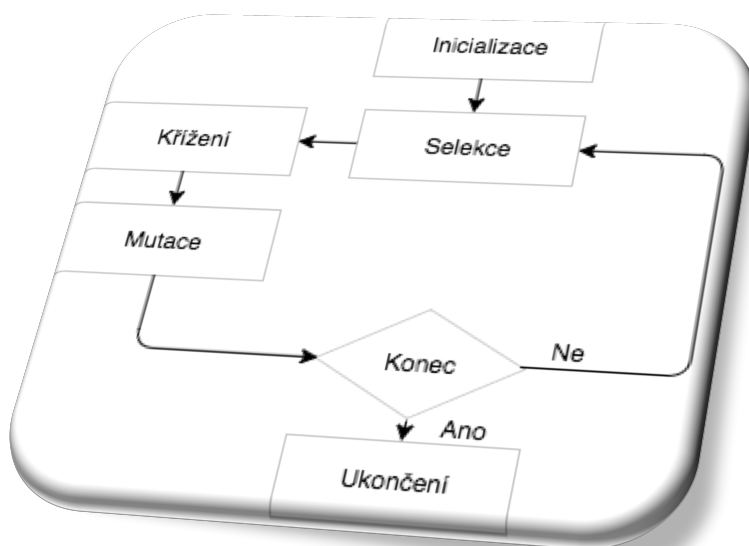
Dění genetického algoritmu představují tři operace (viz. Obrázek 1):

- selekce,
- křížení a
- mutace.

Selekce slouží k výběru jedinců, kteří budou sloužit jako rodiče pro potomky. Potomci od rodičů následně do určité míry zdědí jejich vlastnosti. Tento výběr rodiče je hlavně ovlivněn jeho kvalitou jako jedince, tedy rozhoduje fitness hodnota. K výběru si lze vybrat z mnoha klasických algoritmů k tomu určených, jakož jsou např. ruleta, stochastické univerzální vzorkování, výběr elity či turnaj.

Křížení je proces produkce nových jedinců z informací, jež se vyskytovaly v genech rodičů, dochází tedy k výměně genetického kódu rodičů a projekce do vzniklých potomků. Místa v chromozómu, kde k takovéto výměně dochází, se vybírají náhodně a nazývají se místy křížení.

Mutace je operace, která tak jako předchozí dvě zmíněné může mít také více podob. Princip je ale vesměs vždy podobný. S velice malou předem danou pravděpodobností se z populace vybere jedinec, případně jedinci, u kterých dojde ke změně hodnoty náhodně vybraného genu.



Obrázek 1. Znáznornění průběhu genetického algoritmu.

Praktická úloha

Jak již bylo řečeno, pro demonstraci praktického uplatnění jsem zvolil úlohu pro sestavení učebního rozvrhu. V této kapitole se budu odkazovat na již vysvětlené pojmy z předešlé kapitoly a popíši jejich význam a funkci pro zvolenou úlohu.

Vstupní informace

K tomu, aby bylo vůbec z čeho sestavovat, je potřeba načíst vstupní informace a těmi jsou:

- předměty(Subjects) a
- místnosti(Rooms) pro výuku.

Každá z těchto skupin je načtena z CSV (rooms.csv) souboru, kde každá položka je na novém řádku a své parametry oddělují znakem středník. Ukázka takového zápisu, kde první sloupec je jméno předmětu, druhý je kapacita předmětu a třetí je počet vyučovacích hodin přes týden:

MAT;320;4
SFC;90;2
TIN;340;3
POVa;45;2

Soubor s místnostmi má obdobný formát, krom posledního parametru, tedy obsahují jen název místnosti a kapacitu. Po načtení informací ze vstupních souborů, jsou tyto informace uloženy v podobě tříd *Subject* a *Room*. Za zmínku také stojí, že předměty s větším počtem vyučovacích hodin, než je jedna, jsou rozděleny a uloženy do tříd zvlášť, tolikrát, kolik vyučovacích hodin původně zastupovali, mají společný identifikátor, ale liší se offsetem.

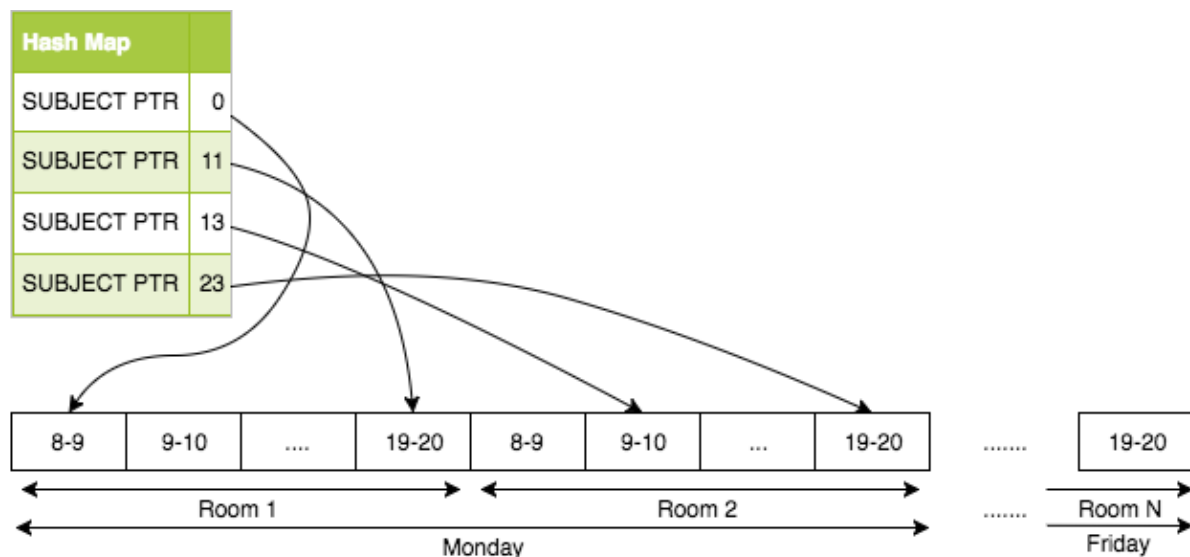
Chromozóm (*chromosome*)

Nyní je velmi důležité zvážit, jak bude vypadat chromozóm, a tudíž i možné řešení problému, nejedná se již totiž o jednoduchou reprezentaci v binární soustavě, je třeba vzít do úvahy současně více parametrů a vymyslet sofistikovanější řešení. Jako podobu chromozómu jsem zvolil pole, jehož délka se dá vyjádřit následovně:

$$roomSchedule = \text{konec dne} - \text{začátek dne} * \text{počet dní} * \text{počet místností}$$

Začátkem a koncem dne se zde myslí začátek a konec vyučování v daný den. Takto dostaneme pole, které svou délkou reprezentuje všechny místnosti, pro každý den i vyučovací hodinu.

Spolu s tímto polem je vytvořena HashMapa, která obsahuje tolik položek, kolik je vyučovaných předmětů, tedy $n * Subject$. Tyto předměty jsou zastoupeny jako klíče a jejich hodnoty jsou náhodně přiřazeny indexy z pole *roomSchedule*. Tímto vznikne vztah mezi volnými vyučujícími místnostmi a předměty. Pro demonstraci poslouží Obrázek 2.



Obrázek 2. Znáznornění chromozómu.

Fitness funkce

Po vygenerování chromozómů pro počáteční populaci je třeba tyto chromozómy také ohodnotit. Fitness funkce hodnotí jen dva požadavky pro každý předmět přiřazený předmět chromozómu:

1. Předmět nesmí být přiřazen do místnosti, jejíž kapacita je nižší než kapacita předmětu.
2. Do jedné stejné místnosti nesmí být přiřazeny dva předměty současně ve stejnou hodinu a den. Také se nesmí ve stejném dnu a hodině vyučovat tentýž předmět s rozlišným offsetem (aby studenti nezmeškali látku ze stejného předmětu).

Za každou splněnou podmínku dostane přidělený předmět jeden bod, to logicky znamená, že každý předmět může získat maximálně dva body, vynásobíme-li počet předmětů maximem možných bodů, dostaneme *maxScheduleScore*. Fitness hodnota chromozómu se pak vypočítá jako získaný počet bodů *scheduleScore* / *maxScheduleScore*, výsledkem je hodnota od 0 do 1.

Získáním všech fitness hodnot chromozomů aktuální populace a vydělením počtem chromozomů lze získat průměrná fitness hodnota pro danou populaci.

Selekce

U operace selekce není žádné speciální řešení, náhodně se vybere 25 % jedinců z dané populace jako rodiče a zároveň 10 % jedinců s největším fitness hodnotou se uchová v paměti. Po křížení rodičů se nově získané chromozómy vezmou a nahradí za náhodně vybrané jedince, avšak tito jedinci nesmí patřit do předešlého výběrů jedinců s nejvyšším fitness ohodnocením.

Křížení

Křížení probíhá klasickým vícebodovým křížením, kde jsem stanovil konstantu *NUMBER_OF_CROSSOVER_POINTS* = 20, tudíž dojde ke křížení v chromozómu na dvaceti náhodných místech.

Mutace

Mutace pro danou populaci vybere tolik jedinců, kolik určuje *MUTATION_SIZE*, a u těchto jedinců náhodně prohodí v chromozómu dva předměty.

SPUŠTENÍ SKRIPTU

Vstupní informace lze libovolně s ohledem na formát změnit úpravou CSV souborů v adresáři: „/GATimeSheduler/src/resources/*“.

Java aplikaci lze spustit přes IDE, nebo ručně v terminálu z kořenového adresáře projektu příkazem: „sh runGATimeSheduler.sh“.