MATEMATICKO-FYZIKÁLNÍ FAKULTA, UK

Student Genetic

zápočtový program

Martin Mareš

Obsah

Uživatelská dokumentace

O aplikaci

Aplikace se snaží pomocí genetických algoritmů nalézt ideálního studenta. Kvalita studenta je určována podle jeho znalostí a spokojenosti na konci „školního života“. Konkrétní průběh semestru modeluje simulace, která je ovlivněna vlastnostmi studenta.

Po skončení simulování tedy aplikace ukazuje, které vlastnosti dělají studenta spokojeného a dobře naučeného. Na druhou stranu byla ale aplikace vytvořena bez větší znalostí psychologie, takže větší část postupů v simulaci vychází z mých domněnek a pozorování.

V současné chvíli generuje současná verze aplikace očekávatelný výsledek tj. studenta, který chodí na přednášky a učí se průběžně.

Princip fungovaní

Aplikace nechává žít několik studentů ve společném prostředí, ve kterém spolu navzájem sdílí náhodně generovaný rozvrh. Toto prostředí předává studentům navíc společné události (testy, domácí úkoly a volno časové aktivity).

Každý student na tyto události reaguje jinak v závislosti na jeho vlastnostech, povaze a stavu těla. Hlavními faktory jsou schopnost učit se a dovednost zapamatovat si věci dlouhodobě. Učením, odpočíváním, spaním, apod. se tyto vlastnost vylepšují nebo zhoršují.

Ve zbývajícím čase si každý student volí činnosti podle toho, jakou má povahu. Všechny činnosti a události mají určité hodnocení (pozitivní nebo negativní) a v závislosti na povaze studenta a jeho tělesném stavu se započítávají do jeho celkové spokojenosti.

Uživatelské prostředí

Po spuštění uvítá uživatele úvodní obrazovka, která je tvořena spodními tlačítky pro posun v čase simulace a vrchní částí, která popisuje aktuální stav simulace. Simulaci lze krokovat ručně nebo lze spustit automatické hledání až do nalezení studentů, kteří na konci „školního života“ budou splňovat zadané parametry z pole „Cíl hledání“.

Jednotlivé časové údaje jsou udávány v „akademických“ hodinách (1,5 „standardní“ hodiny). Toto označení bylo zvoleno kvůli úspoře výpočetního výkonu a kvůli snazšímu vyplňování volného času.

Nejlepší student je označení pro studenta, který dosahuje nejlepších výsledků ve škole a také největší celkové spokojenosti.

Nastavení parametrů simulace

Jednotlivé „psychologické“ parametry a koeficienty lze upravovat v „parametrech simulace“. Je možné změnit parametry generovaných rozvrhů – počet hodin (aplikace náhodně volí v uzavřeném intervalu `<# hodin – 1 , # hodin + 1>`) a maximální počet hodin v jednom dni.

Poté je možné ovlivnit pravděpodobnost, že student pojede domů, pokud si ohřívá jídlo od maminky a dobu cesty domů.

Koeficienty simulace

Schopnost učit se je desetinné číslo v intervalu `(koeficient schopnosti učení, při kterém se učení vzdává, 2 - koeficient schopnosti učení, při kterém se učení vzdává)` *např. (0.3, 1.7)*. Označuje relativní velikost znalostí, které se student naučí za hodinu, pokud není unavený a má standardní náladu.

Koeficient zapomínání po dni učení je relativní velikost zapomenutých znalostí z předchozího dne, které si student dnes po učení zapamatuje dlouhodobě.

Dále je možné ovlivnit množství látky, které si může student odnést z hodiny, kde dává pozor a z hodiny, kde nedává pozor (z hodiny, na kterou nejde si neodnese nic).

Programátorská dokumentace

Zdrojové kódy jsou rozděleny do několika souborů, uživatelské prostředí je tvořeno soubory `MainForm.cs` a `SettingsForm.cs`. V těchto souborech jsou obslouženy události uživatelského prostředí. Při aktivaci automatické simulace dochází ke spuštění dvou vláken na pozadí – jedno postupně krokuje den po dni samotnou simulaci a druhé každých 500 ms aktualizuje údaje o studentovi v UI.

Společné prostředí simulace

Simulace je samostatně funkční a mohla by být používána pouze jako knihovna. Samotné logika její logika je pak ve třídě `Simulation`, tato třída inicializuje `SimulationCalendar` a `LifeEnviroment`. Tyto třídy je dále možné z třídy `Simulation` ovládat (posouvat se na další den a zaregistrovat se na příjem událostí.

`SimulationCalendar` slouží pouze pro manipulaci s datem a pro posun po dnech. V případě přetečení data (na další týden, semestr, generaci) třída zasílá událost registrovaným posluchačům.

`LifeEnviroment` složí jako společné prostředí pro studenty. V této třídě je například generován rozvrh. Na začátku každého semestru je náhodně vytvořen nový rozvrh podle parametrů z nastavení, kde počet hodin v týdnu je náhodně zvolen v intervalu ±1 za dodržení maximálního počtu hodin v jednom dni.

Každý den také vytváří prostředí náhodně několik událostí (pravděpodobnost vytvoření je 30%). Každá událost má náhodně zvolený typ, datum provedení. Tyto události jsou definovány jako potomky `EnviroEvent`. Události jsou typu test, domácí úkol a volno časová aktivita. Pro datum je použit součet dvou náhodných čísel menších než 7. Díky tomu jsou nejpravděpodobněji události uskutečněny kolem 7 dnů od vytvoření. Tyto události jsou na začátku dne předány studentům, kteří je zpracují a rozplánují si je.

Po předání události si každý student prožije simulaci jednoho dne. Tato simulace je prováděna pro studenty paralelně ve více vláknech pomocí `Parallel.ForEach()`, díky tomu dochází ke zvýšení výkonnosti. Na konci každého dne je zvýšeno množství znalostí `LastBestKnowladge`, které se zatím mohl student dozvědět. Znalosti jsou reprezentovány ve formě čísla, kde po jedné hodině ve škole se student může maximálně dozvědět `Core.KNOWLADGE_PER_ACADEMIC_HOUR` (v současné chvíli 100).

Na konci každého semestru se v rozvrhu zruší výuka a student dostane šanci naučit se na „test“ na konci týdne – tím je simulováno zkouškové období.

Po průchodu 5 semestrů (zvoleno z důvodu zvýšení rychlosti) dochází k hledání dalších studentů pomocí genetických algoritmů (viz dále).

Simulace prožití dne

Ve třídě `Student` dochází k simulování jednotlivých událostí v průběhu dne. Na začátku dne student nejprve obdrží události od společného prostředí.

Domácí úkoly hned naplánuje na datum odevzdání. U testů si rovnoměrné rozloží učení na `Gens.LengthOfPlanning`[1] dnů před plánovaným datem psaní. Oproti tomu volno časovou aktivitu si student vloží do kalendáře pouze pokud na tento den má zatím jen málo událostí. Málo událostí je počet událostí, které je schopen student zvládnout spolu s volno časovou aktivitou v závislosti na jeho povaze (vlastnost `Gens.LerningPreferences`). Odmítnutí účasti na události zapříčiní snížení spokojenosti studenta přidáním záporných bodů.

Po zpracování událostí dochází k prožívání dne. Při tom zpracovává události poděděné z `LifeEvent`. Na začátku každého dne zapomene student nějaké znalosti podle koeficientu zapomínání. Poté se ráno nasnídá, což mu trvá určitou dobu podle toho, jestli si vaří, nebo jestli si ohřívá jídlo, nebo jestli chodí do restaurace. Při vaření se mu zvyšuje schopnost učit se, protože se odreagovává. Zatímco při cestě do restaurace se mu snižuje spokojenost, protože utrácí.

Podle povahy jde poté do školy, nebo zůstává doma. Pokud jde do školy, tak se může aktivně účasnit výuky (psát si poznámky) nebo může jenom pasivně poslouchat. Pokud je aktivní tak, si ze školy odnese více znalostí, ale přichází domů s nižší schopností učit se. Pokud je pasivní, tak si toho méně odnáší, ale doma se mu lépe učí a je spokojenější, protože si toho nemusí moc psát.

Pokud tento den nemá školu, tak může jet domů. Když si na koleji ohřívá jídlo, tak je větší pravděpodobnost (lze zvolit v nastavení), že pojede domů. Když ale jel domů v předchozích pár dnech, tak se pravděpodobnost cesty domů snížila. Pravděpodobnost se snižuje nebo zvyšuje vytvářením aritmetického průměru z aktuální pravděpodobnosti a konstanty z nastavení (`konstanta` pro den, kdy nejel domů a `100 – konstanta` pro den, kdy jel domů).

Poté jde na oběd a podle povahy začne buď prací a nebo zábavou. Při práci zpracuje rozplánované události. Pokud má ale už nízkou schopnost učení, tak to vzdá, protože by se už nic nenaučil. Stejně tak skončí dříve pokud už všechno umí. Pokud ale nemá žádnou událost a

1 Jedna z vlastností studenta; na začátku se volí náhodně.

má na to povahu, tak se může průběžně učit (v závislosti na jeho znalostech se učí buď jednu nebo dvě hodiny)

Při zábavě může podle podle povahy dělat věci, které ho baví. Při sportu si odpočne a zvýší se mu schopnost učení. Při čtení se mu méně zvýší schopnost učení a výletem s přáteli stráví více času. Zároveň může vyrazit na volno časovou aktivitu, pokud jí na dnešek má naplánovanou. Pokud ve volném čase chodí nejraději s přáteli, tak stráví na aktivitě více času než v ostatních případech.

Když po všech těchto událostech má ještě nějaký čas, tak může ještě podniknout nějakou věc, která by ho mohla bavit (voleno náhodně).

Poté si dá večeři a jde spát. Pokud spí krátkou dobu, tak to sníží jeho spokojenost, pokud spí delší dobu, tak to zvýší jeho spokojenost. Pokud se v tomto dni něčemu naučil, tak se mu sníží relativní množství znalostí, co na začátku každého dne zapomene.

Může se stát, že danou událost také nestíhá to se pak projeví zvýšením negativního hodnocení. Jednotlivé vlastnosti události jsou zadefinovány uvnitř rozhraní `LifeEvent`.

Schopnosti a vlastnosti studenta

Každá schopnost je reprezentována číslem, které se po každém učení/odpočinku násobí určeným koeficientem, aby se snížila nebo zvýšila. Díky tomu je zohledněn fakt, že po pár hodinách učení mu už nejdou znalosti tak snadno do hlavy. Většina schopností je omezena určitými konstantami, aby například po spánku nebyla schopnost učení příliš vysoká.

Navíc student sbírá kladné a záporné body, které udávají jeho spokojenost. Spokojenost studenta pak označuje `(kladné / (kladné + záporné)) * 100`. Tato hodnota je udávána v procentech.

Vlastnosti jsou reprezentovány výčtovými proměnnými (`enum`) a jsou spojeny do logických celků. Navíc je přítomná ještě proměnná, která říká jak dopředu se student učí `Gens.LengthOfPlanning`.

Všechny vlastnosti se převádějí do pole `int` a toto pole se poté nazývá Geny studenta. Pomocí Geny lze studenta vytvořit a využívá se v genetickém algoritmu.

Genetický algoritmus

Po dokončení 5 semestrů dochází k vyváření nové generace studentů pomocí genetických algoritmů[2]. Studenti jsou seřazeni podle bodů, které označují číslo: `znalosti * 2 + spokojenost.` Nejlepší dvě třetiny studentů se navzájem promíchají a zbylá třetina se vygeneruje opět náhodně.

2 Využil jsem tohoto textu pro konstrukci http://cgg.mff.cuni.cz/~pepca/prg022/luner.html

Studenti, co se promíchají mají 95% šanci, že se jejich DNA bude prohazovat (křížení). Pokud se má prohodit, tak mají jednotlivé chromozomy 50% šanci, že budou prohozeny. Navíc každý chromozom má 10% šanci, že bude náhodně vygenerován (mutace). Takto vzniklí studenti jsou základem pro další simulaci v novém životním cyklu. Tyto vlastnosti jsou zadefinována ve třídě `Gens`.

Závěr

Vnější prostředí je simulováno po dni a vnitřní život studenta je simulován po akademických hodinách. Díky tomuto přístupu není ale simulace tak rychlá, jak bych chtěl. Řešení pomocí prostředků matematické analýzy, které by odstranilo velkou část iterací, je složité a selhává díky velkému množství proměnných, které ovlivňují jednotlivé děje.

Podařilo se mi ale vytvořit simulaci, která generuje očekávatelné výsledky. Na druhou stranu je založena pouze na mých pocitech a domněnkách, takže se nedá s jistotou tvrdit, že výsledek je obecně správný.