

Други домаћи задатак из Генетских алгоритама

1. Задатак: PacLike

Осмислити и имплементирати решење из области еволуционих алгоритама за проблем оптималног решавања игре PacLike. Решење треба да садржи предлог алгорита, образложење, имплементацију и добијене резултате на датим тест примерима у форми PowerPoint презентације.

1.1. Увод у проблем

Проблем је модификована верзија игре Pac-Man. Игра се састоји од лавиринта са једним агентом којим управљате и четири њему супротстављена агента који покушавају да га стигну и спрече да покупи све тачкице из лавиринта. Свака тачкица коју ваш агент покупи вреди 1 поен, а циљ је пронаћи оптималну стратегију, односно покупити све тачкице пре него што налетите на супротстављеног агента. Понашање противничких агената је модификовано у односу на оригиналну верзију игре – постоји само један мод, а то је мод прогањања вашег агента, док ваш агент нема на располагању никакве специјалне способности и једино што може је да упорно избегава супротстављене агенте. Ово је значајно отежана верзија игре у односу на оригиналну и није познато да ли је решива, али ваш задатак је да пробате да обучите свог агента да је реши.

Већи део игре је имплементиран комплетно и на вама је само да искористите генетски алгорита за оптимизацију параметара које ће ваш агент користити да успешно реши проблем, односно да успешно генерише своје следеће потезе када год се то од њега тражи (сваки фрејм). Пожељно је да симулација успешно ради у реалном времену, а за сам процес оптимизације може се користити убрзана верзија симулације која само враћа коначно решење (без рендеровања).

Почетна класа је RunSimulation, цео код је отворен, али није дозвољено мењати енџин игре ни на који начин осим онако како је записано у документацији ове класе.

Саму оптимизацију није неопходно радити у јави, али је пожељно због једноставности покретања (односно због тестирања резултата). Агент мора бити имплементиран у јави, у складу са спецификацијом енџина, односно са описом класе RunSimulation.

2. Предлог решења

Око предлога решења можете се консултовати са мнош, у случају да нисте сигурни да је то што бисте урадили дозвољено. Предлог решења носи 20% поена и ако је добра идеја, али је лоше реализована, можете добити неке поене за дати предлог. Водите рачуна о томе да предлог не би требало да буде генерички, односно да морате имати јасно исказану идеју коју сте желели да имплементирате.

Предлог решења за поене без имплементације се пише у форми текстуалног документа и захтева вашу комплетну анализу проблема, односно које сте могуће идеје размотрили, шта је то од предности и мана проблема што сте хтели да искористите, због чега и како, као и разлоге због којих сте елиминисали неке друге могућности. То значи да треба анализирати начин рада игре и на основу тога закључити какво би потенцијално решење могло да донесе добре резултате. Ако сте већ имплементирали свој предлог и

закључили да је добар, није потребно писати такав документ, јер се предлог брани у оквиру имплементације и одбране. У супротном, потребно је или предложити и имплементирати неку другу стратегију, или у текстуалној форми поред предлога образложити решење, односно разлоге због којих оно није успело. Ако је лоше решење исправно имплементирано, уз добро образложење разлога за неуспех можете добити половичне поене.

3. Имплементација

3.1. Испис резултата

Пратити најбоље решење у току генерација (водити рачуна да се оно не изгуби). Метода коју имплементирате (за генерисање следећег потеза) мора да буде детерминистичка и да зависи само од параметара генерисаних помоћу алгоритма (а никако од других случајних параметара). Може се користити произвољна додатна структура података за прослеђивање параметара, помоћно израчунавање итд. Помоћна израчунавања могу да буду у било којој произвољној форми (нпр. дозвољено је искористити генетски алгоритам за обучавање неуронске мреже која бира следећи потез на основу тренутног стања). Није дозвољено да решење буде хардкодовано (увек исто и базирано на унапред израчунатим познатим шаблонима кретања агената).

По извршавању алгоритма добијене параметре најбољег решења обавезно сачувати у текстуалној форми, за потребе поновног коришћења. Више пута покренути оптимизацију и упоредити добијена решења за истог агента – анализирати да ли су сваки пут слична или постоји више могућих решења.

4. Презентација

Да бисте освојили поене за презентацију, морате испунити одређене услове

4.1. Садржај презентације

На првом слајду обавезно написати: име, презиме и број индекса (**у форми СМЕР-БРОЈ/ГОДИНА**). Надаље описати предлог решења, речима и графички ако је то потребно. Изложити резултате покретања својих експеримената са различитим конфигурацијама (ако сте покушавали више метода) и резултате вишеструког покретања алгоритма за коначно одабрану имплементацију у форми табеле. У табели исписати релевантне параметре најбољег решења и његову прилагођеност (односно освојени број поена) за свако покретање алгоритма. Закључити која конфигурација се показала као најбоља и да ли решење увек конвергира истим вредностима или варира.

4.2. Одбрана

Одбрана је усмена, а састоји се из излагања презентације и образложења добијених резултата. Биће потребно и да се кратко прође кроз код и по потреби нешто детаљније објасни или измени на лицу места. Било какво неразумевање сопственог кода неће бити толерисано (одузимају се сви поени). Образложења типа „радио сам пре две недеље, не сећам се” и слично не долазе у обзир. Пре почетка рада водите рачуна о томе да сте добро разумели тему и захтеве из спецификације и поставите питања уколико нешто од наведеног није јасно. У случају да за одбрану немате презентацију, изгубићете 10% поена (2 поена).

5. Предаја рада

Рад треба да садржи **имплементацију тражене класе која наслеђује Player и имплементацију алгоритма (предлог решења ако није успела имплементација), и презентацију решења**, у једној архиви. Архиву

послати са **свог студентског мејла** на mtomic@raf.rs (у наслову поруке напишите да је други домаћи из Генетских алгоритама). Дозвољени језици за имплементацију алгорита су: C, C++, Java, Python, Octave.

Меки рок за предају задатка је **05. јун 2020. у 12.00 (подне)** – почетак четврте колоквијумске недеље. Предаја до овог рока може донети максимално 100% поена које носи овај задатак, односно 20 поена. Сваки минут закашњења умањује максимални број поена за 0,0125 (задаци предати после 06. јуна у 10.40 вреде 0 поена). **Исправна имплементација са свим наведеним ставкама** носи 90% поена, а **исправна презентација (са одбраном)** 10% поена. **У случају да одбрана не буде успешна, поени са имплементације се не рачунају** (тј. задатак носи 0 поена). У случају да имплементација није успела, ако имате предлог решења у текстуалној форми, он сам носи највише 20% поена.

Сва своја питања у вези са задатком можете слати мејлом. Очекивано време за одговор је 24 сата од тренутка када сте послали мејл, осим ако је викенд (тада се продужује на први следећи радни дан).