

Други домаћи задатак из Генетских алгоритама

1. Задатак: TSPJ

Осмислити и имплементирати решење из области еволуционих алгоритама за проналажење оптималног решења за проблем TSPJ. 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 TSPJ је проширени проблем трговачког путника, где путник пролази кроз n локација, посећује сваку локацију тачно једном да би започео један од n послова и враћа се на прву локацију. По започињању посла, путник одмах прелази на следећу локацију а посао наставља да се обавља независно од њега. Овај проблем је у вези са разним практичним ситуацијама, нпр. са високо-аутоматизованим производним погонима, где оператор посећује машине које могу да ненадгледано обаве неке операције како би доделио/започео посао на тој локацији и враћа се у своју базу када додели све послове. Циљ је да се минимизује време завршетка последњег посла, тј. да се сви послови заједно обаве за што краће време. Проблем се у литератури може наћи под називом TSPJ – Traveling Salesman Problem with Job-times.

У поставци проблема користе се скуп чворова (локација, радних станица) и скуп послова. Скуп чворова припада и почетни чвор (чвор 0) – базна станица, одакле оператор креће и где се враћа када обиђе све локације. Карактеристика базне станице је да у њој није могуће започети (обавити) ниједан посао – број послова који се додељује је за 1 мањи од броја укупног броја чворова.

Сваком чвору (осим базној станици) може се доделити тачно један посао из скупа послова. На свакој радној станици може да се обави било који посао, али је свакој радној станици потребно различито време за исти посао. Радне станице су на различитим местима, тако да је оператору потребно неко време да дође од једне до друге станице да би започео посао. Занемарује се време које је потребно оператору да покрене посао када се нађе на некој радној станици.

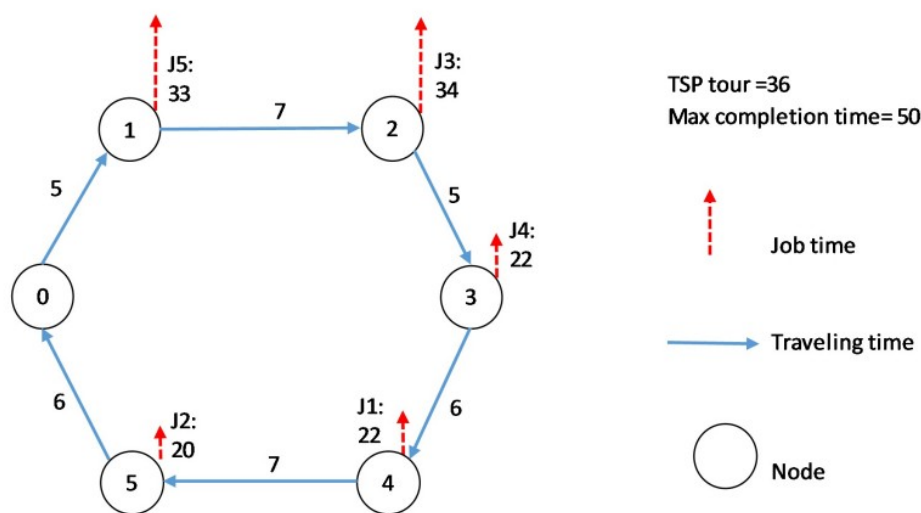
Дати подаци по поставци задатка:

- број послова (n) – послови и радне станице су нумерисани од 1 до n ; подразумева се да постоји и базна станица која је нумерисана са 0
- времена $d(i,j)$ – време које је потребно оператору да дође од станице i до станице j ($0 \leq i, j \leq n$)
- времена $t(i,j)$ – време које је потребно радној станици i да обави посао j ($1 \leq i, j \leq n$)

Пример из литературе, $n=5$ (https://digitalcommons.uri.edu/oa_diss/1205/)

	ЧВОР - $d(i,j)$						ПОСАО - $t(i,j)$				
Чвор	0	1	2	3	4	5	1	2	3	4	5
0	-	5	9	12	6	10	-	-	-	-	-
1	5	-	7	9	12	10	20	22	32	25	33
2	9	7	-	5	10	12	21	20	34	23	32
3	12	9	5	-	6	10	20	22	30	22	34
4	10	12	10	6	-	7	22	24	31	22	32
5	6	10	12	10	7	-	21	20	32	24	34

Чворови по редоследу	0	1	2	3	4	5	0
Додељени посао	0	5	3	4	1	2	0
ТСП тура до чвора	0	5	12	17	23	30	36
Време посла	0	33	34	22	22	20	0
Време завршетка	0	38	46	39	45	<u>50</u>	36



Приложена је поставка проблема и једна инстанца решења (не нужно оптимално решење). Инстанца решења састоји се од редоследа обиласка чворова и одговарајућих додељених послова (прва два реда у другој табели). Поред тога, приказан је начин израчунавања функције трошка коју треба минимизовати - максимално време завршетка посла. Максимално време завршетка за ово решење је подвучено у табели (износи 50). Из табеле се може видети да се време завршетка сваког посла може израчунати као трајање ТСП туре од базне станице до тог чвора + време које је потребно да се изврши тај посао. Нпр. за чвор 3 у овом решењу потребно је обићи део туре 0-1-2-3, што је $d(0,1)+d(1,2)+d(2,3)=17$ и онда стартовати додељени посао (у овом случају посао 4) на том чвору. За посао 4 на чвору 3 потребно је $t(3,4)=22$ јединице времена, тако да је време завршетка посла на том чвору $17+22=39$. Пример је илустрован на слици. Ваш задатак је да направите такву ТСП туру и доделу послова да се максимално време завршетка за дату инстанцу проблема минимизује.

2. Предлог решења

Око предлога решења можете се консултовати са мном, 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 проблем детаљно 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.

2. Имплементација

2.1. Испис резултата

Тестирање се врши на основу улазних фајлова које ћете накнадно добити (биће генерисани на основу неких познатих тсп проблема). Улазни фајл садржи број n (број послова), затим у наредних $n+1$ линија по $n+1$ број $(d(i,j), 0 \leq i,j \leq n)$ и потом у наредних n линија по n бројева $(t(i,j), 1 \leq i,j \leq n)$ који представљају бројеве из описа проблема. У идеалном случају требало би да је могуће покренути алгоритам из командне линије задавањем улазног фајла. Резултат (најбоље решење) је довољно исписати на конзоли, у форми $2n+1$ броја раздвојена размацама, редом: вредност функције трошка, пермутација радних станица (подразумева се да је 0 и на почетку и на крају, дакле исписати само пермутацију чворова од 1 до n) и пермутација послова (по редоследу посеђивања/стартовања). Због приказа резултата је потребно да се за сваки улазни фајл изгенерише по 3 решења (у три извршавања алгоритма) и да се на основу тога израчуна вредност просечног и најбољег решења за сваки фајл. За извршавања која стављате у презентацију, резултате сачувајте у одговарајућим текстуалним датотекама због провере.

3. Презентација

Да бисте освојили поене за презентацију, морате испунити одређене услове

3.1. 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 просечно и најбоље добијено решење (само вредност функције трошка, конкретне пермутације). Закључити која конфигурација се показала као најбоља и да ли решење увек конвергира истим вредностима или варира.

3.2. 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.

4. Предаја рада

Рад треба да садржи **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 алгоритма су: C, C++, Java, Python, Octave.

Меки рок за предају задатка је **16. јун 2023. у 12.00 (подне)** – петак 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 17. јуна у 11.20 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 48 сати од тренутка када сте послали мејл, осим ако је викенд (тада се продужује на први следећи радни дан).