

Основне академске студије
Информациони инжењеринг

Основи рачунарске интелигенције

Примена генетских алгоритама

(материјали за вежбе)

- 1. Увод**
2. Задатак 1
3. Задатак 2

Генетски алгоритми

алгоритми из класе еволуционих алгоритама

инспирација потиче из биологије и теорије еволуције

класа метода заснованих на популацији

тражење решења се изводи испитивањем већег броја решења
заједно

решења пролазе кроз разне облике трансформација

Садржај

1. Увод
- 2. Задатак 1**
3. Задатак 2

Задатак 1

Задатак

Имплементирати генетски алгоритам намењен налажењу најкраћег пута између два места уз ограничење да пут мора проћи кроз свако од преосталих места тачно једном.

Дати су називи свих места и за сваки пар места дата је у километрима приближна дужина пута који их повезује.

1. Зрењанин (ЗР)
2. Кикинда (КИ)
3. Кањижа (КА)
4. Нови Сад (НС)
5. Суботица(СУ)
6. Сомбор (СО)

Наћи решење за пут СУ–СО преко преосталих места.

	ЗР	КИ	КА	НС	СУ	СО
ЗР	0	55	100	45	130	115
КИ	55	0	55	90	85	115
КА	100	55	0	105	40	95
НС	45	90	105	0	100	95
СУ	130	85	40	100	0	60
СО	115	115	95	95	60	0

Задатак 1

Имплементација

Написати класу **Jedinka**.

Класа описује јединку у популацији током извршења генетског алгоритма.

Јединка има свој генетски материјал (садржај).

Јединка са својим садржајем треба да представља пут између два места.

Пут обухвата четири преостала места: НС, КИ, КА, ЗР

Пут представља секвенцу у којој се свако од четири преостала места помиње тачном једном.

Секвенца пута може бити представљена као секвенца целобројних индекса који одговарају местима кроз која пут пролази.

Путу НС–КИ–КА–ЗР одговара секвенца индекса 4–2–3–1.

Иницијализатор `__init__()` треба да прихвати секвенцу индекса.

Имплементација

Написати класу **Populacija**.

Класа описује популацију која бива формирана током извршења генетског алгоритма.

Популација представља скуп јединки.

У популацији може постојати више јединки које описују исти пут.

На нивоу класе, популација може бити имплементирана као листа јединки.

Имплементација

Написати класу **Prilagođenost**.

Класа описује функцију прилагођености која се током извршења генетског алгорита користи за оцењивање прилагођености јединке, односно квалитета решења које је представљено јединком.

На основу матрице раздаљина, ознака почетног и крајњег места на путу за који се тражи решење, потребно је за дату јединку израчунати прилагођеност.

Потребно је израчунати дужину пута који креће од почетног места, затим иде редом кроз места која одговарају путу описаном преко јединке и завршава се у крајњем месту.

Ако јединка одговара путу НС–КИ–КА–ЗР, који је дужине 245 км, дужина целокупног пута од почетног до крајњег места СУ–НС–КИ–КА–ЗР–СО је 460 км.

За потребе конкретног решења може се искористити дужина целокупног пута као мера прилагођености јединке, при чему мање дужине одговарају већим прилагођеностима.

Иницијализатор `__init__()` треба да прихвати матрицу раздаљина, индекс почетног места и индекс крајњег места.

Класа треба да садржи методу `evaluiranje()` која ће за прослеђену јединку вратити као резултат њену прилагођеност.

Имплементација

Написати класу **GenetskiAlgoritam**.

Класа треба да подржи извршавање генетског алгоритма.

Основни ток алгоритма

Формирати почетну популацију од n јединки.

Индексе у оквиру јединке поставити на (псеудо)случајне вредности тако да јединка описује исправан пут између почетног и крајњег места.

Могу се користити само индекси који не одговарају почетном или крајњем месту и сваки од тих индекса се појављује тачно једном.

Спровести еволуцију крећући од почетне популације као текуће популације.

У сваком кораку еволуције настаје нова популација јединки као нова генерација.

Нова генерација настаје на основу јединки из текуће популације.

Поступак настанка нове генерације је увек исто структуриран.

Изводи се селекција јединки.

Изводи се укрштање јединки.

Изводи се мутација јединки.

Нова генерација представља текућу популацију у наредном кораку.

Еволуција се спроводи кроз укупно g корака.

На основу најприлагођеније јединке из најмлађе генерације, формирати решење.

Имплементација

Написати класу **GenetskiAlgoritam**.

Поступак селекције јединки

Формирати укупно $n/2$ парова јединки.

За сваки пар изабрати две јединке из текуће популације.

Избор имплементирати тако да што је јединка прилагођенија то има веће шансе да буду изабрана у неки пар.

Јединке текуће популације могу бити сортиране по својој прилагођености и на основу њиховог ранга може бити вршен избор јединке с одређеном вероватноћом.

Иста јединка може бити изабрана у више парова.

Не дозволити формирање пара који би исту јединку обухватио два пута.

Могуће је у једном пару имати две различите јединке које имају исти садржај (исте секвенце индекса).

Задатак 1

Имплементација

Написати класу **GenetskiAlgoritam**.

Поступак укрштања јединки

За сваки од $n / 2$ парова јединки који су настали селекцијом извести укрштање.

Од једног пара јединки настају две нове јединке.

Нове јединке постају део нове генерације.

Нове јединке настају тако што долази до размене садржаја између јединки из пара.

Како свака јединка обухвата секвенцу индекса, која је дужине четири, при укрштању могу бити размењена два индекса.

Потребно је водити рачуна о томе да нове јединке одговарају исправним секвенцама индекса.

Пример укрштања

прва јединка из пара: 4-2-3-1

друга јединка из пара: 2-1-3-4

прва нова јединка: 4-2-1-3

последња два индекса су добијена тако што су последња два индекса из прве јединке из пара постављена у оном редоследу у којем се јављају код друге јединке из пара

друга нова јединка: 2-1-4-3

Задатак 1

Имплементација

Написати класу **GenetskiAlgoritam**.

Поступак мутације јединки

За сваку од нових јединки примењује се мутација с вероватноћом m .

Потребно је водити рачуна о томе да нове јединке одговарају исправним секвенцама индекса.

Пример мутације (померање индекса у секвенци према лево)

јединка пре мутације: 4–2–1–3

јединка након мутације: 2–1–3–4

Задатак 1

Имплементација

Написати класу **GenetskiAlgoritam**.

Иницијализатор **__init__()** треба да прихвати следеће аргументе

листу ознака места

матрицу раздаљина између места

ознаку почетног места

ознаку крајњег места

број јединки у популацији (n)

очекује се паран број

број генерација које ће настати кроз еволуцију почетне популације (g)

вероватноћа спровођења мутације (m)

Класа треба да садржи методу **pokretanje()** у оквиру које ће бити спроведено формирање почетне популације и спроведена еволуција кроз g корака.

Класа треба да садржи методу **preuzimanje()** која ће као резултат вратити јединку што одговара решењу.

Користити објекте класа **Jedinka**, **Populacija** и **Prilagodенost**.

У класе укључити додатне методе према потреби.

Задатак 1

Покретање

Поставити вредности параметара и испитати добијено решење.

Поставити број јединки на 10.

Поставити број генерација на 20.

Поставити вероватноћу спровођења мутације на 5%.

Експериментисати с другачијим вредностима параметара и анализирати како се промене у њиховим вредностима могу одразити на поступак тражења решења.

Садржај

1. Увод
2. Задатак 1
- 3. Задатак 2**

Задатак 2

Задатак 2

Проширити решење за Задатак 1 тако да се селекција спроводи као такмичарска селекција.

За први члан пара, бира се с вероватноћом t прилагођенија јединка од неке две јединке које су (псеудо)случајно изабране из текуће популације, док се с вероватноћом $t - 1$ бира она мање прилагођена.

За други члан пара, спроводи се поступак као за први.

Проширити решење за Задатак 1 тако да се користи механизам елитизма.

Из текуће популације пренети две најприлагођеније јединке као један пар у нову генерацију без поступка укрштања и мутације.

За преосталих $n / 2 - 1$ парова јединки спровести поступке селекције, укрштања и мутације на редован начин.