Математички факултет

Универзитет у Београду

**СЕМИНАРСКИ РАД**

У оквиру курса Рачунарска интелигенција

Тема:

*Хибридизација Генетског и*

*ВНС Алгоритма*

Професор: Асистенти: Студенти:

Александар Картељ Стефан Капунац Марко Савић,149/2019

Денис Аличић Мирко Кордић,242/2019

**Април, 2023.**

**САДРЖАЈ**

**1.Увод**

**1.1** Предмет истраживања

1.2 Пример проблема

1.3.Потребни програми и ресурси

2.Учитавање и обрада полазних података

3.Генетски алгоритам

3.1.Селекција

3.2.Укрштање

3.3.Мутација

4.ВНС

5.Хибридизација

6.Анализа резултата

7.Закључак

8.Литература

**1.Увод**

* 1. **Предмет истраживања**

У овом семинарском раду, приказан је рад различитих комбинација генетског алгоритма и внс алгоритма.

Идеја је да се на конкретном проблему испробају како појединачни алгоритми тако и њихове различите хибридизације, те да се потом упореде њихови резултати.

* 1. **Пример проблема**

За пример проблема на којем би рад ових алгоритама био тестиран узет је “Проблем трговачког путника” (TSP) , јер у неку руку представља изазов за решавање коришћењем ових алгоритама, због заснованости на пермутацијама о којима посебно треба водити рачуна приликом примене генетског алгоритма.

Проблем трговачког путника (Travelling salesman problem) представља проблем у коме је потребно да за задату листу градова и међусобних растојања међу њима, одредимо редослед обиласка градова тако да рута којом се крећемо буде најкраћа могућа, при чему сваки град треба да буде посећен тачно једанпут.

Конкретан скуп података на коме су тестирани алгоритми представља низ градова Западне Сахаре који се састоји од 29 градова задатих својим координатама.

**1.3. Потребни програми и ресурси**

Комплетан код писан је у програмском језику Python, а извршен је у окружењу Jupyter Notebook. Коришћена је python бибилиотека *networkx* и њене имплементације рада са графовима,библиотека *matplotlib* за цртање графика, и друге основне python библиотеке. Рачунар на ком су тестирани алгоритми садржи 8GB RAM меморије.

**2.Учитавање и обрада полазних података**

Улазни подаци представљају низ од 29 градова Западне Сахаре, задати својим координатама. Зарад компактнијег чувања података и пропратних информација које су потребне током израчунавања, одлучено је да се подаци о градовима чувају у оквиру графа. У те сврхе искоришћена је python библиотека *networkx,* која омогућава ефикасан рад над графовима. Подаци су сачувани у оквиру потпуно повезаног графа, чији су чворови градови, задати својим *x* и *y* координатама, док су гране одређене паром чворова које спајају и додатном иформацијом о њиховом растојању. За рачунање растојања између градова коришћено је еуклидско растојање.

**3. Генетски алгоритам**

Генетски алгоритам је један од еволутивних алгоритама инспирисаних Дарвиновом теоријом еволуције. Припада стохастичним алгоритмима претраге са хеуристиком. Најчешће се користи за проблеме у дискретном домену али може се генерализовати коришћењем одговарајућих пресликавања.

Генетски алгоритми су засновани на посматрању популације јединки и њиховом мењању кроз генерације.

Одлучено је да се јединке представљају њиховим кодом који је представљен пермутацијом бројева од 0 до 28. Друга обавезна компонента сваке јединке јесте фитнес, којим се мери квалитет јединке. Фитнес ће у нашем конкретном примеру представљати дужину пута представљеног кодом јединке. Рачунање се изводи једноставним пролажењем кроз код јединке и сабирањем растојања између суседних градова(чворова).

Како генетски алгоритми спадају у еволутивне алгоритме, који врше оптимизацију коришћењем техника инспирисаних природном еволуцијом као што су селекција,укрштање и мутација, тако ћемо проћи кроз сваки од ових концепата и размотрити њихов начин имплементације у нашем конкретном примеру.

У свим имплементацијама коришћен је елитизам, још једна битна карактеристика еволутивних алгоритама, која спречава губљење добрих јединки. Иако ће се најбоље јединке најверованије налазити у скупу одабраних родитеља, применом укрштања и мутације може доћи до тога да потомци имају лошији квалитет. Приступ подразумева да се неколико најбољих јединки одабере и директно прекопирају у наредну генерацију, при чему број елитистички одабраних једники не сме бити превелики.

**3.1. Селекција**

За процес селекције одабрана је класична турнирска селекција.

Као параметре функције селекција добија популацију јединки, величину турнира и листу забрањених јединки. Листа забрањених јединки се прослеђује како не би дошло укрштања истих родитељских јединки.

На почетку се прави листа дозвољених индекса уклањањем забрањених из листе индекса свих јединки популације.

Потом се из листе дозвољених индекса случајним избором бира tournament\_size јединки које ће учествовати у процесу селекције.

Једноставним проласком кроз одабране јединке вршимо претрагу оне са најмањим фитнесом. Таква јединка ће бити резултат процеса селекције.

**3.2. Укрштање**

За процес укрштања одабрано је једнопозиционо укрштање, где се из дужине кода јединке случајним избором бира *breakpoint* на основу ког вршимо генерисање деце од кодова родитељских јединки.

Проблем код позиционог укрштања може да настане код пермутација, које су основ нашег примера. Ако бисмо позиционон укрстили могли бисмо да добијемо невалидне конфигурације деце (понављање неког броја). Због тога је одабрана модификација позиционог укрштања под називом *укрштање првог реда*.

Оно се састоји из наредних корака:

1. Изаберемо регион у првом родитељу
2. Пресликамо изабрани регион у прво дете на исто место
3. Десно од тог региона(циклично) постављамо гене из другог родитеља. Ако се ген који желимо да ставимо већ налази у детету прескачемо га, и посматрамо наредни ген.
4. Аналогно радимо и за друго дете

**3.3 Мутација**

Како је наш проблем заснован на пермутацијама, тако је и процес мутације морао бити прилагођен томе.

Како би се одржала форма пермутације кода јединке за процес мутације одабрана је ротација два различита гена кода.

Битан параметар функције мутације јесте вероватноћа мутације, о којој ће бити речи касније.

**4.ВНС**

Променљиво околинско претраживање (Variable Neighborhood Search - VNS) је метахеуристички оптимизациони алгоритам који има за циљ да пронађе најбоље решење датог оптимизационог проблема у простору свих могућих решења. Он је дизајниран да се може применити на различите оптимизационе проблеме, укључујући комбинаторне, континуалне и мешовите проблеме.

ВНС алгоритам почиње са почетним решењем и наставља истраживање простора решења итеративним примењивањем различитих околинских структура на тренутно решење. Околинска структура дефинише скуп суседних решења која се могу добити прављењем малих измена на тренутном решењу.

На свакој итерацији, алгоритам оцењује решења у околини тренутног и прихвата најбоље као ново тренутно решење. Ако најбоље решење у околини тренутног није боље од тренутног, алгоритам се премешта на другу околинску структуру и понавља процес.

Алгоритам наставља да истражује простор решења итеративним примењивањем различитих околинских структура, сједињавајући их у један оптимизациони поступак. Он наставља са претрагом свих околинских структура до испуњења услова заустављања, као што је достигнуће максималног броја итерација или одређеног нивоа квалитета решења.

ВНС је показао ефикасност у решавању различитих оптимизационих проблема, укључујући транспорт, логистику и инжењерство, и показао се као моћна и ефикасна техника оптимизације.

За потребе овог рада коришћене су две врсте ВНС алгоритма. Основни нередуковани ВНС који након сваког креирања *new\_solution* позива локалну претрагу како би покушао да унапреди новокреирано решење, и редуковани ВНС који овај корак прескаче.

**5.Хибридизација**

Хибридизација генетског алгоритма (ГА) и Variable Neighborhood Search (VNS) је метод оптимизације која комбинује две различите технике у циљу постизања бољих перформанси у решавању различитих оптимизационих проблема.

Хибридизацијом ГА и ВНС-а могу се уједно користити предности оба приступа. ГА и ВНС имају комплементарне квалитете. ГА је добро прилагођен за истраживање великог простора претраге и налажење разноликог скупа решења, док је ВНС добар у интензивирању претраге и истраживању околина око обећавајућих решења,тј. може се користити за детаљнију локалну претрагу у мањим регионима простора претраге. Ово може омогућити ефикасније и прецизније проналажење глобалних и локалних оптимума у различитим врстама оптимизационих проблема.

Хибридизација ГА и VNS-а може имати значајне предности у поређењу са појединачном применом ГА или VNS-а, посебно за проблеме који имају комплексну структуру претраге и велики број променљивих које треба оптимизовати. Хибридизација ГА и VNS-а може омогућити ефикасније и прецизније проналажење глобалних и локалних оптимума у различитим врстама проблема.

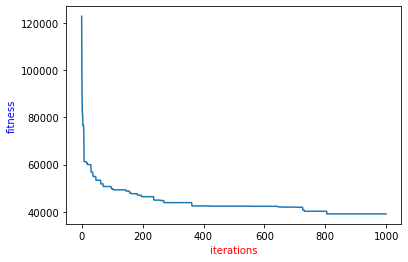
**6.Aнализа решења**

У овом поглављу ћемо приказати резултате примене различитих варијанти наведених техника над описаним проблемом.

Оптимално решење проблема је 27603.

**6.1 ВНС**

Први примењен алгоритам био је ВНС. Креирана је насумична јединка, и над њом је покренут ВНС алгоритам са 10 околина, 1000 итерација и вероватноћом преласка у другу околину 0.5.



Као резултат добија се 39059, што је далеко од оптималног. То можемо приметити и са графика промене фитнеса кроз итерације, где фитнес доста споро конвергира ка решењу јер тек у 800-тој итерацији долази до стагнирања решења.